



Universidad de Oviedo

Programa de Doctorado en Informática

Tesis doctoral

**AUTOGESTIÓN ENERGÉTICA
DE UN PARQUE DE PCs**

Ramón Medrano Llamas

2022



Universidad de Oviedo

Programa de Doctorado en Informática

Tesis doctoral

**AUTOGESTIÓN ENERGÉTICA
DE UN PARQUE DE PCs**

presentada por

Ramón Medrano Llamas

para la obtención del grado de

Doctor por la Universidad de Oviedo

Directores: Daniel F. García Martínez, Joaquín Entrialgo Castaño

Gijón, 15 de Julio de 2022



RESUMEN DEL CONTENIDO DE TESIS DOCTORAL

1.- Título de la Tesis	
Español/Otro Idioma: Autogestión Energética de un Parque de PCs	Inglés: Autonomic Power Management of a PC Fleet
2.- Autor	
Nombre: Ramón Medrano Llamas	DNI/Pasaporte/NIE: _ _
Programa de Doctorado: Doctorado en Informática	
Órgano responsable: Centro Internacional de Posgrado	

RESUMEN (en español)

Tanto la transición a energías verdes para reducir a cero las emisiones de CO₂ en 2050 como el incremento de los precios de la energía sugieren que debemos encontrar formas de reducir el consumo de electricidad en todos los sectores, y en particular en el sector TIC. Una gran mayoría de empresas cuentan con flotas de computadores para sus empleados, de tamaño variable, pero creciente. Mientras que, en la operación de estas flotas, uno de los costes más grandes es el consumo energético, muchos de los computadores pasan largos períodos de tiempo encendidos, pero sin uso, malgastando así grandes cantidades de electricidad.

La gestión dinámica de energía (*dynamic power management*, DPM) es un conjunto de técnicas y métodos que se aplican a diferentes niveles para reducir el consumo y la generación de calor de un computador. Incluye técnicas tan variopintas como la gestión de frecuencia de un microprocesador o el apagado de dispositivos que no están en uso. Las diferentes técnicas de DPM están dirigidas por una serie de políticas de gestión energética, que establecen las directrices de operación de los diferentes sistemas. Estas políticas se generan usando diferentes métodos, adaptados al componente que se está gestionando y los objetivos a alcanzar.

Esta tesis presenta una técnica de DPM aplicada a una flota de computadores completa. El objetivo es reducir el consumo de la flota apagando los computadores de forma proactiva, pero manteniendo altos niveles de satisfacción en los usuarios de la flota. La generación de las políticas que dirigen al sistema de gestión energética se producen en base a datos recogidos de la propia flota en estudio y gestión, de forma que se generan unos modelos de utilización que permiten representar y predecir el comportamiento de cada usuario, pudiendo generar así políticas totalmente personalizadas para cada usuario.

Una de las principales aportaciones de esta tesis es la utilización de la satisfacción como métrica central de cara a resolver el problema de optimización que es la generación de políticas energéticas. Se definen nuevas métricas que permiten medir la satisfacción de los usuarios cuando la flota está siendo optimizada por el sistema de gestión energética y, lo más importante, generar políticas energéticas que garantizan un determinado nivel de satisfacción a cada usuario.

Con el objetivo de comprobar y aplicar el método de gestión energética propuesto, se ha implementado una herramienta que permite obtener políticas para una determinada flota, estudiar variaciones y, utilizando un método de simulación, generar registros de flotas sintéticas.



Finalmente, se ha realizado una validación del trabajo presentado, mostrando los resultados que es posible ahorrar hasta el 90% de la energía de otra forma malgastada.

RESUMEN (en Inglés)

Both the transition to green energy to reduce CO2 emissions to net-zero by 2050 and the increase in energy prices suggest that we must find ways to reduce electricity consumption in all sectors, and in particular in the ICT sector. A large majority of companies have fleets of computers for their employees, of variable size, but growing. While in the operation of these fleets one of the biggest costs is energy consumption, many of the computers spend long periods of time turned on, but idling, thus wasting large amounts of electricity.

Dynamic Power Management (DPM) is a set of techniques and methods that are applied at different levels to reduce the consumption and heat dissipation of a computer. It includes techniques as varied as microprocessor dynamic frequency scaling (DFS) or turning off devices that are not in use. The different DPM techniques are directed by a series of energy management policies, which establish the operating guidelines of the different components. These policies are generated using different methods, adapted to the component being managed and the objectives to be achieved.

This thesis presents a DPM technique applied to a complete computer fleet. The goal is to reduce fleet consumption by proactively shutting down computers, while maintaining high levels of user satisfaction. The generation of the policies that direct the energy management system are produced based on data collected from the fleet under study and management. Utilization models are generated from that data and allow representing and predicting the behavior of each user, thus being able to generate fully customized policies for each user.

One of the main contributions of this thesis is the use of satisfaction as a central metric in order to solve the optimization problem that is the generation of energy policies. New metrics are defined that allow user satisfaction to be measured when the fleet is being optimized by the energy management system and, most importantly, to generate energy policies that guarantee a certain level of satisfaction for each user.

In order to verify and apply the proposed energy management method, a tool has been implemented that allows obtaining policies for a given fleet, studying variations and, using a simulation method, generating synthetic fleet records.

Finally, a validation of the presented work has been carried out, showing the results that it is possible to save up to 90 % of the energy otherwise wasted.

SR. PRESIDENTE DE LA COMISIÓN ACADÉMICA DEL PROGRAMA DE DOCTORADO EN INFORMÁTICA

Resumen

Tanto la transición a energías verdes para reducir a cero las emisiones de CO_2 en 2050 como el incremento de los precios de la energía sugieren que debemos encontrar formas de reducir el consumo de electricidad en todos los sectores, y en particular en el sector TIC. Una gran mayoría de empresas cuentan con parques de computadores para sus empleados, de tamaño variable, pero creciente. Mientras que en la operación de estos parques, uno de los costes más grandes es el consumo energético, muchos de los computadores pasan largos períodos de tiempo encendidos, pero sin uso, malgastando así grandes cantidades de electricidad.

La gestión dinámica de energía (*dynamic power management*, DPM) es un conjunto de técnicas y métodos que se aplican a diferentes niveles para reducir el consumo y la generación de calor de un computador. Incluye técnicas tan variopintas como la gestión de frecuencia de un microprocesador o el apagado de dispositivos que no están en uso. Las diferentes técnicas de DPM están dirigidas por una serie de políticas de gestión energética, que establecen las directrices de operación de los diferentes sistemas. Estas políticas se generan usando diferentes métodos, adaptados al componente que se está gestionando y los objetivos a alcanzar.

Esta tesis presenta una técnica de DPM aplicada a un parque de computadores completa. El objetivo es reducir el consumo del parque apagando los computadores de forma proactiva, pero manteniendo altos niveles de satisfacción en los usuarios del parque. La generación de las políticas que dirigen al sistema de gestión energética se produce en base a datos recogidos de la propio parque en estudio y gestión, de forma que se generan unos modelos de utilización que permiten representar y predecir el comportamiento de cada usuario, pudiendo generar así políticas totalmente personalizadas para cada usuario.

Una de las principales aportaciones de esta tesis es la utilización de la satisfacción como métrica central de cara a resolver el problema de optimización que es la generación de políticas energéticas. Se definen nuevas métricas que permiten medir la satisfacción de los usuarios cuando el parque está siendo optimizado por el sistema de gestión energética y, lo más importante, generar políticas energéticas que garantizan un determinado nivel de satisfacción a cada usuario.

Con el objetivo de comprobar y aplicar el método de gestión energética propuesto, se ha implementado una herramienta que permite obtener políticas para un determinado parque, estudiar variaciones y, utilizando un método de simulación, generar registros de parques sintéticos.

Finalmente, se ha realizado una validación del trabajo presentado, mostrando los resultados que es posible ahorrar hasta el 90 % de la energía de otra forma malgastada.

Abstract

Both the transition to green energy to reduce CO_2 emissions to net-zero by 2050 and the increase in energy prices suggest that we must find ways to reduce electricity consumption in all sectors, and in particular in the ICT sector. A large majority of companies have fleets of computers for their employees, of variable size, but growing. While in the operation of these fleets one of the biggest costs is energy consumption, many of the computers spend long periods of time turned on, but idling, thus wasting large amounts of electricity.

Dynamic Power Management (DPM) is a set of techniques and methods that are applied at different levels to reduce the consumption and heat dissipation of a computer. It includes techniques as varied as microprocessor dynamic frequency scaling (DFS) or turning off devices that are not in use. The different DPM techniques are directed by a series of energy management policies, which establish the operating guidelines of the different components. These policies are generated using different methods, adapted to the component being managed and the objectives to be achieved.

This thesis presents a DPM technique applied to a complete computer fleet. The goal is to reduce fleet consumption by proactively shutting down computers, while maintaining high levels of user satisfaction. The generation of the policies that direct the energy management system are produced based on data collected from the fleet under study and management. Utilisation models are generated from that data and allow representing and predicting the behavior of each user, thus being able to generate fully customized policies for each user.

One of the main contributions of this thesis is the use of satisfaction as a central metric in order to solve the optimisation problem that is the generation of energy policies. New metrics are defined that allow user satisfaction to be measured when the fleet is being optimised by the energy management system and, most importantly, to generate energy policies that guarantee a certain level of satisfaction for each user.

In order to verify and apply the proposed energy management method, a tool has been implemented that allows obtaining policies for a given fleet, studying variations and, using a simulation method, generating synthetic fleet records.

Finally, a validation of the presented work has been carried out, showing the results that it is possible to save up to 90% of the energy otherwise wasted.

A mi familia

Agradecimientos

En primer lugar quiero expresar mi agradecimiento a mis directores de tesis, que durante una década han apoyado este desarrollo y apretado cuando había que hacerlo.

También agradecer a todos los revisores y compañeros que han proporcionado comentarios y mejorado este trabajo en el proceso.

No puedo olvidarme de mi familia y amigos, que me han apoyado en todo momento desde que todo esto comenzó hace casi 18 años.

Índice general

1. Introducción	1
1.1. Definición del problema	1
1.1.1. Gestión dinámica de energía	2
1.2. Objetivos y preguntas de investigación	3
1.3. Contribuciones de investigación	4
1.4. Estructura de la tesis	4
2. Estado del arte	7
2.1. Técnicas estocásticas y basadas en control	7
2.2. Técnicas basadas en aprendizaje automático	10
2.3. Comparación de herramientas comerciales	15
2.4. Técnicas que consideran la satisfacción de usuario	16
3. Modelado del comportamiento y satisfacción de los usuarios	21
3.1. Comportamiento de los usuarios	21
3.1.1. Modelado de la actividad e inactividad	26
3.2. Gestión de energía dirigida por la satisfacción de usuario	35
3.2.1. Satisfacción de usuario	37
3.2.2. Inactividad eliminada	42
3.2.3. Cálculo del tiempo de espera	42
4. Herramienta de optimización	45
4.1. Casos de uso	45
4.2. Aplicación de las políticas energéticas	47
4.3. Recolección de registros de utilización	48
4.4. Implementación de los modelos	49
4.4.1. Apagados solicitados por el usuario	49
4.4.2. Consolidación de modelos	50
4.4.3. Extrapolación de modelos	51
4.5. Procesos de simulación y diseño del software	52
4.5.1. Optimización del ajuste de distribuciones	55
4.5.2. Escalabilidad del simulador	58
5. Validación	61
5.1. Partición de los registros de utilización	61
5.2. Validación de la implementación de la distribución empírica	62
5.3. Comparación entre la satisfacción y satisfacción ponderada	63

5.4. Validación del análisis y su impacto energético	65
5.4.1. Impacto de la consolidación de modelos	67
5.5. Generación de parques sintéticos	70
6. Conclusiones	75
6.1. Objetivos alcanzados y aportaciones	75
6.1.1. Preguntas de investigación	76
6.2. Limitaciones y ampliaciones	78
6.3. Publicaciones derivadas de la tesis	78
A. Análisis de ajuste de distribuciones	83
A.1. Máxima verosimilitud	83
A.2. Gráficos Q-Q	84
A.3. Histogramas y ajuste de leyes exponenciales	86
A.4. Kernel Density Estimation	88
A.5. Simulación	89
B. Guía de uso del software de simulación	91
B.1. Descarga e instalación	91
B.2. Ejecución	91
C. Formato de los registros de utilización	95
D. Notación y abreviaturas	97
Bibliografía	99

Índice de figuras

3.1. Ejemplo de parque con los estados sin gestión energética.	22
3.2. Ejemplo de parque con los estados con gestión energética.	22
3.3. Duración media de los intervalos de inactividad por hora del día y semana.	24
3.4. Cantidad de intervalos de inactividad por hora del día y semana.	25
3.5. Distribución de la duración de los intervalos de inactividad en diferentes horas de un martes.	26
3.6. Esquema de monitorización para detección de eventos.	29
3.7. PDF de la inactividad en escala logarítmica.	31
3.8. Comparación del ajuste con distribuciones log-normal, Pareto y exponencial.	32
3.9. Comparación de la longitud de los intervalos por hora del día entre los registros de utilización y los generados con un modelo log-normal y Pareto.	33
3.10. Comparación entre datos medidos y generados mediante el modelo (martes, 12:00).	34
3.11. Elección de diversos valores del tiempo de espera para un conjunto de intervalos de inactividad.	36
3.12. Comparación del ahorro energético y la satisfacción de usuario.	37
3.13. Función escalón para la satisfacción de usuario, $s(I)$.	38
3.14. Función de satisfacción basada en $Apdex$.	39
3.15. Función de satisfacción con progresividad lineal.	40
4.1. Diagrama del ciclo de optimización para gestión energética.	46
4.2. Arquitectura de recolección de registros de utilización.	48
4.3. Diagrama UML de los procesos de simulación principales.	53
4.4. Modelo de colas $G/G/1$ para los usuarios simulados.	54
4.5. Benchmark de la distribución empírica basada en <i>spline</i> .	56
4.6. Uso de memoria del simulador.	57
4.7. Evolución del tiempo de simulación en función del número de usuarios.	58
4.8. Evolución del tiempo de simulación en función de la duración de la simulación.	59
5.1. S , S_w y R en función de T .	64
5.2. Resultados del test 2 con y sin consolidación de modelos.	68
5.3. Resultados del test 6 con y sin consolidación de modelos.	69
5.4. Generación de parques con diferente jornada y distribución de actividad (A).	72

Índice de figuras

A.1. Comparación de distribuciones con diagramas Q-Q (1).	85
A.2. Comparación de distribuciones con diagramas Q-Q (2).	87
A.3. Comparación de distribuciones con diagramas Q-Q (y 3).	88
A.4. Ajuste mediante el método KDE.	89
A.5. Comparación de distribuciones mediante simulación.	90

Índice de tablas

2.1. Resumen de trabajo previo de técnicas estocásticas y de control (1).	11
2.2. Resumen de trabajo previo de técnicas estocásticas y de control (y 2).	12
2.3. Resumen de trabajo previo de técnicas de aprendizaje automático.	14
2.4. Resumen de herramientas comerciales para la gestión energética.	17
3.1. Registros recolectados de un parque real.	23
4.1. Comparación del rendimiento del simulador con diversas implementaciones de la distribución empírica.	57
5.1. Partición de los registros disponibles.	61
5.2. Tamaños de los registros.	62
5.3. Comparación de resultados para la generación de políticas energéticas.	66
5.4. Resultados del test 2 con y sin consolidación de modelos.	67
5.5. Resultados del test 6 con y sin consolidación de modelos.	68
5.6. Valores por defecto de los parámetros globales del generador de parques.	70
5.7. Valores por defecto de los parámetros temporales del generador de parques.	70
A.1. Comparación de valores PPCC para los datos del martes a las 12:00.	84

Capítulo 1.

Introducción

En este capítulo introductorio se establecen las premisas principales de la investigación: se define el problema y se expresa la motivación de la tesis. Asimismo, se resumen las principales preguntas de investigación y las contribuciones de este trabajo de investigación. Estas preguntas se irán respondiendo a lo largo del desarrollo de la tesis, resumiendo en los siguientes capítulos las contribuciones realizadas y los mecanismos utilizados para ello.

1.1. Definición del problema

Para el año 2050, los gobiernos mundiales deberán asegurar una reducción de hasta el 90 % de las emisiones globales de CO_2 para poder evitar un incremento de más de $2^\circ C$ en la temperatura del Planeta [Solomon et al. 2007]. La producción de electricidad es uno de los mayores productores de sustancias con efecto invernadero, tanto CO_2 como otras como el metano, por lo que tanto la generación con nuevas tecnologías renovables como la reducción del consumo son esenciales para abordar el problema.

En el caso de las empresas, muchas cuentan con enormes parques de computadores, que son usadas a diario y suponen un consumo de energía considerable. Además de la producción de sustancias contaminantes para generar la energía consumida, se cuenta con el problema de la volatilidad de los precios de la misma, que en ocasiones puede representar riesgos financieros sustanciales.

El tamaño de los parques de computadores sigue en aumento (hasta un 6,9 % anual hasta 2040 [Belkhir and Elmeligi 2018]), no solo en los centros de datos, sino también en las oficinas corporativas, que suponen una fracción de coste sustancial [Murugesan 2008], permitiendo un ahorro potencial en energía equivalente a hasta 1.500 millones de dólares en costes de energía cada año.

Las estaciones de trabajo están típicamente encendidas durante todo el día, usando menos de la mitad de sus capacidad de cómputo [Webber et al. 2006]. Incluso si son conformes a las últimas regulaciones de gestión de energía [Reinhard et al. 2011], el consumo de una estación de trabajo suele ser uno o dos órdenes de magnitud más alto que el de un dispositivo móvil, como un portátil o una tableta [Procaccianti et al. 2012]. Muchas organizaciones pueden beneficiarse de ahorros derivados de la implementación de políticas de gestión de energía [Chetty et al. 2009; Walker 2009].

1.1.1. Gestión dinámica de energía

La gestión dinámica de energía (*dynamic power management*, DPM) es un conjunto de técnicas y métodos que tienen como objetivo reducir el consumo energético de uno o varios computadores. Pueden aplicarse a diversos niveles de forma simultánea y/o independiente, tales como a microprocesadores (por ejemplo, con técnicas de gestión dinámica de frecuencia), a dispositivos de almacenamiento (por ejemplo reduciendo o parando discos rotacionales), a nodos de red inalámbrica o dispositivos móviles.

Cualquier técnica DPM está dirigida por una política de gestión energética que define las directrices y los objetivos que se buscan obtener. Por ejemplo, una técnica como el escalado dinámico de frecuencia (*dynamic frequency scaling*, DFS) buscará encontrar un balance entre el consumo energético (y el calor disipado) de un microprocesador y el tiempo de respuesta a las instrucciones del usuario. En general, cuanto más ahorro se busca, peor tiempo de respuesta o capacidad prestada al usuario, es decir, satisfacción proporcionada.

La generación de estas políticas es, por tanto, un problema de optimización del consumo energético con restricciones sobre la capacidad de respuesta al usuario, que pueden tener diversas formas y métricas dependiendo del nivel al que se aplique la técnica DPM: serán diferentes si se está optimizando el consumo de un dispositivo, un computador o un parque completo.

El problema abordado en esta tesis consiste en diseñar un método para modelar el comportamiento de los usuarios de un parque de computadores y generar políticas de gestión de energía que permitan materializar un ahorro de energía sustancial. Este modelo se basa en el estudio de los diferentes tiempos de uso de un computador por parte del usuario: hay tiempos de actividad, momentos de inactividad e intervalos de tiempo donde el computador está apagado. Los momentos de inactividad, cuando el computador está encendido pero no siendo utilizado de forma activa presentan una oportunidad de ahorro importante: un gestor energético podría apagar los computadores de forma proactiva, ahorrando la energía que se consumiría durante esos intervalos de tiempo. El gestor energético requiere una serie de políticas y objetivos para controlar el parque, que se generan con el método presentado en esta tesis.

La generación de estas políticas ha de tener en cuenta la satisfacción de los usuarios, con el objetivo de ofrecer el mejor servicio posible dentro de los parámetros establecidos por el administrador del parque. La satisfacción ha de ser definida en función de las prioridades y necesidades del usuario, de forma que refleje de manera inequívoca sus preferencias dentro del modelo del sistema que va a usarse para generar políticas de control energético. En este trabajo se prioriza minimizar el desasosiego que produce tener que encender el computador una y otra vez tras las diferentes pausas que se realizan a lo largo de una jornada laboral. Las políticas se implementan como perfiles de configuración para el gestor energético del sistema operativo.

Otra cuestión relevante del modelo y método para generar políticas energéticas son las relaciones entre los perfiles de uso de diversos usuarios. Personal del mismo

departamento, edificio o país tendrán características similares al tener los mismos días festivos, o momentos para las pausas. En parques muy grandes puede ser relevante encontrar estos grupos de forma que se generen menos políticas o bien su cálculo sea más rápido o más estable.

1.2. **Objetivos y preguntas de investigación**

Habiendo encontrado una cantidad potencial de energía para ahorrar, se establecen dos objetivos de investigación, acompañados de preguntas de investigación que se responderán en esta tesis.

Objetivo A: Diseñar un método para modelar y predecir el comportamiento de los usuarios del parque desde el punto de vista de la actividad.

Para conseguir este objetivo de investigación, se aborda primero el modelo del comportamiento de los usuarios, intentando descubrir cuál es el mejor método para predecirlo; en segundo lugar se investiga cómo modelar y medir la satisfacción con el sistema de gestión de energía que los usuarios reciben del parque:

- P. A.1:** ¿Cuál es el potencial de ahorro energético con un sistema dinámico gestión de energía?
- P. A.2:** ¿Cuáles son los mejores modelos para el comportamiento de los usuarios?
- P. A.3:** ¿Cómo modelar la satisfacción de los usuarios?
- P. A.4:** ¿Cómo se relaciona el comportamiento de unos usuarios con el de otros?
¿Merece la pena buscar modelos individuales o en grupo?

Objetivo B: Establecer una técnica de gestión de energía dinámica que genere políticas energéticas respetando la satisfacción de usuario.

El segundo objetivo, una vez determinados los modelos de satisfacción y comportamiento de los usuarios, consistente en generar políticas energéticas para materializar el ahorro energético e implementarlas en el parque a gestionar:

- P. B.1:** ¿Cómo generar políticas energéticas en base a la satisfacción y el comportamiento de los usuarios?
- P. B.2:** ¿Cómo se relaciona la satisfacción con el ahorro de energía?
- P. B.3:** ¿Cómo se implementan las políticas en el parque?
- P. B.4:** ¿Cómo hacer experimentos con diversas variantes del parque?

1.3. Contribuciones de investigación

Esta tesis tiene tres contribuciones de investigación principales, todas dentro del campo de la gestión dinámica de energía de parques de computadores que se utilizan de forma interactiva, esto es, cualquier parque a excepción de centros de datos. Se gestionan parques que están compuestas tanto por ordenadores de escritorio como por portátiles, sin embargo no se incluyen dispositivos móviles más pequeños, como móviles o tabletas, dado que presentan comportamientos particulares, como se explicará en detalle al desarrollar los modelos.

En primer lugar se desarrolla un método para generar modelos que representen de forma fidedigna el comportamiento de los usuarios del parque. Representar el comportamiento es clave, ya que este modelo servirá de base para la generación de políticas, y lo más importante, para entender cuál es la definición de satisfacción más conveniente para la optimización.

Estos modelos, de tipo estocástico, permiten condensar los datos procedentes de registros de utilización del parque en un conjunto de distribuciones estadísticas. Estos modelos pueden utilizarse para generar políticas energéticas y entender el comportamiento de los usuarios a lo largo de la jornada.

En segundo lugar, se ha determinado una definición de una métrica de satisfacción de usuario. Esta definición, sencilla y con un mínimo de parámetros, permite calcular políticas energéticas en base a un objetivo de satisfacción deseada por el administrador del parque. Asimismo, se ha desarrollado un método para, en base a los modelos de comportamiento, determinar políticas energéticas que aseguran un nivel de satisfacción determinado.

La satisfacción es el argumento central de la optimización energética, dado que es la única restricción que se establece en el problema de optimización que se aborda para la generación de políticas energéticas. Esta métrica, en principio, debe resumir todas las necesidades de los usuarios de forma sencilla, eficiente y explicable.

Finalmente, se ha desarrollado una herramienta utilizando técnicas de simulación por eventos discretos que permite tanto estudiar parques como sus variantes, sin necesidad de realizar estudios *in situ*.

1.4. Estructura de la tesis

El resto de la tesis se organiza como sigue:

- En el capítulo 2 se realiza un análisis exhaustivo del trabajo relacionado con esta tesis.
- En el capítulo 3 se detalla el método de modelado del comportamiento de los usuarios, las diferentes definiciones de satisfacción de usuarios y el método para el cálculo de forma analítica de las políticas energéticas en base a la satisfacción.
- El capítulo 4 se dedica a la herramienta de optimización, que implementa la generación de las políticas. Se detalla tanto la recolección de registros de

utilización como la generación de políticas, la optimización de los modelos y el ajuste de distribuciones y la aplicación de políticas al parque. Asimismo, se detallan otros casos de uso, como la generación de parques sintéticos o el estudio de variantes.

- El capítulo 5 repasa los ejercicios y pruebas de validación que se han realizado tanto a los modelos, como a las políticas generadas. Se cubre también el conjunto de tests a casos de uso de generación de parques y variantes.
- El capítulo 6 resume los objetivos alcanzados, las conclusiones, las limitaciones del trabajo realizado y el trabajo futuro

Capítulo 2.

Estado del arte

En este capítulo se analiza el cuerpo de trabajo precedente dentro del espacio de la gestión dinámica de energía. Los trabajos analizados se han agrupado en tres categorías: técnicas basadas en métodos estocásticos, técnicas basadas en aprendizaje automático y, finalmente herramientas disponibles comercialmente como productos o servicios listos para ser instalados. En la última sección de este capítulo se evalúan las técnicas que consideran la satisfacción de usuario.

2.1. Técnicas estocásticas y basadas en control

La primera clase de técnicas analizadas son las basadas en métodos estocásticos y mecanismos de control. Los procesos estocásticos representan mecanismos naturales aleatorios, que pueden ser estacionarios, cíclicos o depender de muchas variables. En el caso de las técnicas aplicadas a la gestión energética, se trata normalmente de procesos de Markov, esto es, que sólo dependen del estado del sistema en un momento dado y de un conjunto de transformaciones conocido, pero no del histórico de estados por los que el sistema ha transitado. Los mecanismos de control, en cambio, tienen como objetivo mantener una serie de métricas dentro de unos intervalos predefinidos, utilizando actuadores para alterar el estado del sistema de forma que las métricas a controlar varían su valor.

Estas técnicas generalmente modelan a los usuarios utilizando modelos estadísticos que tratan de predecir cuáles van a ser los comportamientos de éstos, en diferentes variables dependiendo de cómo se describa al usuario: desde la duración de periodos de inactividad hasta la demanda futura de interfaces inalámbricas. Los estados de los computadores se modelan en muchos casos usando autómatas que permiten representar los diferentes estados y sus relaciones. Ambos se combinan en modelos dinámicos, como modelos de colas, que permiten representar el estado de un parque a lo largo del tiempo.

Una vez determinadas las variables que se van a usar para modelar a los usuarios y los diferentes estados por los que van a ir pasando los computadores, para generar políticas de gestión energética es necesario resolver los modelos, de forma que se obtengan los parámetros de apagado apropiados para cada momento. Para realizar esta optimización pueden usarse modelos de Markov, que resuelven modelos de colas o técnicas de control, que permiten optimizar el funcionamiento basándose en un objetivo para una métrica.

Las principales ventajas de los modelos estocásticos es su explicabilidad: es relativamente sencillo determinar cómo se producen las políticas al basarse en métodos estadísticos que pueden resolverse de forma analítica en un buen número de ocasiones y por simulación en el resto. Por contra, las desventajas principales se encuentran en la cantidad de parámetros que pueden llegar a tomar y en lo complejo que puede resultar seleccionarlos de forma que representen bien las propiedades del sistema a modelar.

El trabajo de Srivastava [Srivastava et al. 1996] calcula la duración de los intervalos futuros utilizando un método de regresión. Modela el sistema como un autómata de estados energéticos, que determina el consumo en cada estado y las posibles transiciones.

En su trabajo, Lu. [Lu et al. 2000] han definido un gestor energético básico registrando el uso de cada una de las tareas en ejecución en un computador. Cuando una tarea no está utilizando un dispositivo, y el coste del cambio de estado es menor que la energía estimada a ser ahorrada, el controlador apaga dicho dispositivo inmediatamente. Ramanathan [Ramanathan and Gupta 2000] usa un método similar, pero incorporan una comparación por competición de un algoritmo adaptativo con uno no adaptativo.

Hwang [Hwang and Wu 2000] utiliza un método de estimación basado en medias móviles de la inactividad del usuario, sobre ellas modela el sistema como un autómata finito sencillo. Para predecir la duración de la siguiente inactividad y decidir si apagar o no el computador (y programar el correspondiente encendido antes de la vuelta del usuario) hace uso de la media móvil y del estado del computador en ese momento. Consideran que los intervalos de inactividad de corta duración no proporcionan nada más que ruido y recomiendan ignorarlos.

El trabajo de Benini [Benini et al. 1999, 2000] es el trabajo seminal que establece los métodos estocásticos para la gestión energética. En este trabajo, el sistema se modela como un sistema de colas: la tasa de servicio es variable y dependiente del estado energético del sistema en cada momento. Para el establecimiento de políticas proponen resolver un modelo de Markov, que permite obtener la política óptima en cada estado del sistema.

El el primero trabajo de [Qiu et al. 1999] se presenta una aproximación similar, pero incorpora una comparación interesante del consumo para cada tasa de servicio y longitud de la cola. En una actualización de este trabajo, se incorporó la generación de políticas energéticas mediante la resolución de un modelo de Markov parcialmente observable [Qiu et al. 2007]. Con este método, el sistema se puede controlar de forma óptima sin necesidad de conocer todos sus parámetros. Tan [Tan and Qiu 2008] han usado un método similar, pero incorporan la medición de la probabilidad de encontrar un ajuste de política en función del número de estados del modelo de Markov. El trabajo de Jung [Jung and Pedram 2007] presenta un método parecido, y añade una representación en 3D de los registros de utilización que han procesado.

El grupo de investigación de Simunic [Simunic et al. 2001] ha desarrollado dos métodos, uno basado en teoría de renovación y otro modelo de Markov indexado temporalmente. Ambos proporcionan soluciones óptimas, sin embargo el segundo es notablemente más complejo. Otra contribución de este trabajo es el análisis de la

bondad de ajuste de las distribuciones exponencial y Pareto respecto a la inactividad de usuario. Encontraron que la distribución exponencial no presentaba suficiente bondad de ajuste para este tipo de datos. Para el estudio de los modelos recogieron registros de utilización de parques existentes y consideran el coste de cambio de estado para la optimización, al igual que los trabajos previos. Posteriormente, Phillips [Phillips and Singh 2008; Phillips et al. 2008] ha propuesto un modelo similar, contribuyendo una validación detallada del mismo.

El trabajo de [Chung et al. 2002] presenta dos conceptos novedosos: primero, considera una ventana deslizante de acciones del usuario, lo que permite calcular probabilidades más precisas para las transiciones entre estados; segundo, permite adaptar de forma dinámica el modelo del sistema, variando las transiciones y sus pesos conforme la ventana deslizante avanza. Una política se define en este caso como un conjunto de acciones para cada uno de los estados del modelo en un momento dado. El grupo de Luiz [Luiz et al. 2010] extiende este trabajo añadiendo ventanas de longitud variable para la estimación de la carga de trabajo del sistema, lo que permite además de los beneficios anteriores balancear la precisión de las políticas generadas por el optimizador con el tiempo que se tardan en generar, de manera que el sistema se adapta más fácilmente a los cambios de carga y tamaño del parque. Mantienen varios estimadores en paralelo, con ventanas de diferentes longitudes y escogen el mejor en cada momento usando un método de máxima verosimilitud.

La teoría de control es otro mecanismo popular utilizado para optimizar el consumo de parques. Jung [Jung and Vaidya 2002] desarrollaron uno de los primeros controles para el manejo energético, pero aplicado a redes inalámbricas. Más tarde, en [Juang et al. 2005] incorporaron un controlador PID para gobernar la frecuencia de un grupo de núcleos en una CPU multinúcleo, haciendo al planificador de procesos consciente de la demanda y el estado de todo el complejo de la CPU. Minerick [Minerick et al. 2002] agrega un bucle de retroalimentación para adaptar el mecanismo de control con el tiempo. Más recientemente, Bulay [Bulay-og and Gustilo 2018] desarrolla un control para sistemas de aire acondicionado de centros de datos, agregando unos detalles que permiten generalizar el modelo para sistemas de control de energía genéricos.

Lin [Lin et al. 2018] proponen un método para modelar el consumo que, por primera vez, incorpora proporcionalidad en el consumo en relación a la carga a la que se somete el sistema. Hasta entonces los modelos consideraban el consumo solamente dependiente del estado del sistema.

Sólo dos métodos de los analizados comportan métricas de satisfacción de usuario. Generalmente, los métodos estocásticos implementan modelos de computadores basados en autómatas discretos, que permite utilizar un modelo de Markov para obtener las políticas óptimas al resolverlos. Algunos de los problemas que presentan este tipo de métodos es la complejidad del modelo de Markov cuando se desarrollan modelos con muchos estados (por ejemplo, en parques de muchos computadores), además dichos modelos son en muchas ocasiones sólo parcialmente observable, lo que los hace más complejos de resolver.

La tablas 2.1 y 2.2 resumen los diversos trabajos analizados de esta categoría

de forma cronológica. Se categorizan los trabajos en función del entorno al que se aplican (desde computadores individuales, parques u otros elementos, tales como redes Wi-Fi o procesadores) y el tipo de método que se ha utilizado. Asimismo, se muestran características acerca de los modelos escogidos para representar los usuarios y los sistemas ya que se trata de variables importantes de cara a la elección del mecanismo de resolución y la complejidad del método propuesto.

2.2. Técnicas basadas en aprendizaje automático

Las técnicas de aprendizaje automático permiten descubrir los modelos subyacentes en los parques a modelar procesando los registros de utilización usando inducción de conocimiento con variados métodos, tales como redes neuronales o árboles de decisión. Una técnica de este tipo genera modelos energéticos en base a los registros de utilización y la meta información de los mismos, no siendo necesario diseñar una técnica de inducción desde cero o adaptada específicamente a la gestión de energía.

Uno de los primeros trabajos que aplicó métodos de aprendizaje automático a la gestión energética de parques de computadores fue [Chung et al. \[1999\]](#), que permite realizar una predicción basada en un árbol de decisión; estos árboles se actualizan durante los periodos de inactividad. El modelo predice los tiempos de inactividad y prepara un encendido automático, de forma que el computador esté operativo para cuando el usuario lo requiera. Los autores filtran los intervalos de inactividad más cortos, de forma que se evita afectar a la satisfacción del usuario con ciclos muy cortos.

El trabajo de [Dhiman and Rosing \[2006\]](#) presenta un sistema que escoge entre diversos *expertos* predefinidos. Cada experto es una estrategia que permite generar políticas energéticas, cada una creada con un modelo de aprendizaje automático diferente. El trabajo de [Freund and Schapire \[1997\]](#) utiliza aprendizaje de refuerzo, con modelos que evolucionan conforme el sistema está en operación, obteniendo información y realimentación acerca de las operaciones de apagado y encendido que resultan satisfactorias para los usuarios.

En [Kong et al. \[2006\]](#), los investigadores utilizaron un algoritmo genético para predecir la duración futura de la actividad e inactividad basándose en un registro de intervalos pasados. Configuraron los genes como secuencias entrelazadas de duraciones de actividad e inactividad pasada y usaron una función de coste que tiene en cuenta el coste de cambio de estado, pero con buen comportamiento en cargas de trabajo que presentan intervalos de inactividad muy cortos.

Prabha [Prabha and Monie \[2007\]](#) incorpora aspectos básicos de satisfacción de usuarios a un mecanismo de aprendizaje por refuerzo agregando un agente que determina las recompensas y las penalizaciones para los refuerzos. Este agente escoge entre políticas de gestión energética existentes.

El trabajo de Jung [Jung and Pedram \[2010\]](#) aplica aprendizaje supervisado a la adaptación de la frecuencia de un procesador multi-núcleo. Han definido propiedades específicas que representan la ocupación de las colas y el ritmo de

Trabajo	Entorno	Método	Satis.	Métrica rendimiento	Modelo sistema	Modelo usuario
Srivastava et al. 1996	computador	regresión	○	energía	automata estados	regresión
Flinn and Satyanarayanan 1999	computador	n/a	●	duración batería	n/a	n/a
Benini et al. 1999	computador	estocástico	○	tamaño de colas	automata estados	sistema Markov
Lu et al. 2000	computador	tracking	○	energía	listas de tareas	n/a
Ramanathan and Gupta 2000	computador		○	energía, latencia	n/a	número intervalos
Hwang and Wu 2000	computador	estocástico	○	ahorro de energía	automata estados	media móvil
Simunic et al. 2001	computador	estocástico	○	tamaño de colas	automata estados	sistema Markov
Jung and Vaidya 2002	Wi-Fi	control	○	capacidad	ventana de transferencia	n/a
Minerick et al. 2002	procesador	control	○	energía	energía conservada	n/a

Tabla 2.1.: Resumen de trabajo previo de técnicas estocásticas y de control (1).

Trabajo	Entorno	Método	Satis.	Métrica rendimiento	Modelo sistema	Modelo usuario
Chung et al. 2002	computador	estocástico	<input type="radio"/>	tamaño de colas	autómata estados	sistema Markov
Juang et al. 2005	computador	control	<input type="radio"/>	producto energ.-latencia	control PID	n/a
Tan and Qin 2008	computador	estocástico	<input type="radio"/>	latencia, energía	autómata estados	POMDP
Phillips et al. 2008	computador	estocástico	<input type="radio"/>	tamaño de colas	autómata estados	sistema Markov
Lin et al. 2009	procesador	control	<input checked="" type="radio"/>	energía, temperatura	n/a	eventos de usuario
Luiz et al. 2010	computador	estocástico	<input type="radio"/>	energía	ventanas de carga	sistema Markov
Setz et al. 2016	computador	estocástico	<input checked="" type="radio"/>	energía	n/a	probabilidad de actividad
Bulay-og and Gustilo 2018	laboratorio	control	<input type="radio"/>	energía	autómata estados	estático
Turkin and Vdovichenko 2019	computador	estocástico	<input checked="" type="radio"/>	tiempo de espera	sistema Markov O.	sistema Markov O.
<i>Este trabajo</i>	parque	analítico	<input checked="" type="radio"/>	inactividad eliminada	sistema de colas	distribución empírica

Tabla 2.2.: Resumen de trabajo previo de técnicas estocásticas y de control (y 2).

llegada. El método se basa en un clasificador bayesiano, que genera una política energética para gobernar la frecuencia del procesador.

Al mismo tiempo, Liu [Liu et al. 2010] también utiliza aprendizaje por refuerzo implementando el algoritmo Q -learning para seleccionar la mejor de múltiples políticas energéticas. El método propuesto se basa en medir el rendimiento del sistema con una carga de trabajo sintética y comparar los pares de consumo y latencia observados para todas las combinaciones de nivel de carga y política aplicada, escogiendo la mejor para cada situación. En varios trabajos, Candrawati [Candrawati and Hashim 2016a,b,c] también utiliza Q -Learning para gestión energética, en este caso modelando el computador como una serie temporal que informa a un controlador para tomar acciones sobre múltiples componentes.

Skakun [Skakun et al. 2005] y, posteriormente, Csurgai [Csurgai-Horváth and Bitó 2011] utilizaron una red neuronal para predecir la actividad de usuario. Estas redes neuronales utilizan como entradas numerosas métricas de estado del parque, tales como la serie temporal de actividad e inactividad, niveles de utilización de los recursos de los computadores, información horaria, etc. Las redes neuronales se entrenan con registros de utilización que están clasificados previamente con la política energética más apropiada para cada momento, de esta forma para nuevas entradas, la red neuronal puede indicar qué acción tomar para cada usuario. Hay variantes que entrenan una red neuronal por usuario y otras con solamente una para todo el parque, de manera que tanto la entrada como la salida de la red neuronal ha de representar todos los computadores (normalmente mediante vectores y estructuras de datos de más cardinalidad).

La tabla 2.3 resume el conjunto de trabajos analizados que basan su método en técnicas de aprendizaje automático, indicando las mismas métricas que para los trabajos basados en métodos estocásticos y de control.

Una de las razones más fuertes para utilizar técnicas de aprendizaje automático es que permiten al sistema adaptarse de forma dinámica, evolucionando los modelos conforme el parque está en ejecución, en lugar de tener que regenerar los modelos cada cierto tiempo.

Por contra, se necesita una mayor cantidad de datos para tener un modelo fiable, datos que hay de preparar de antemano estableciendo etiquetas, depurando y simplificando las representaciones y entrenando variantes de las redes neuronales para encontrar la adecuada. Dichos modelos son, además, más complejos de explicar y depurar que modelos estocásticos o de control. En el caso de requerir un sistema que se adapte de forma automática conforme evoluciona el parque, el utilizar modelos como los presentados requiere realizar entrenamiento de nuevas redes o árboles de aprendizaje de forma continua.

Sólo uno de los trabajos analizados tenía en cuenta la satisfacción de usuario como métrica fundamental.

Trabajo	Entorno	Método	Satis.	Métrica rendimiento	Modelo sistema	Modelo usuario
[Chung et al. 1999]	computador	árbol de aprendizaje red	<input type="radio"/>	ratio predicción	autómata estados	n/a
[Skakun et al. 2005]	computador	red neural	<input type="radio"/>	energía	n/a	n/a
[Kong et al. 2006]	computador	algoritmo genético	<input type="radio"/>	energía	genes inactividad	n/a
[Dhiman and Rosing 2006]	computador	sistema experto	<input type="radio"/>	energía	autómata estados	sistema Markov
[Theocharous et al. 2006]	computador	$C_{4.5}$, k-nearest	<input checked="" type="radio"/>	energía	autómata estados	insatisfacción
[Prabha and Monie 2007]	computador	aprendizaje refuerzo	<input checked="" type="radio"/>	ahorro energía	autómata estados	agente refuerzo
[Shye et al. 2008]	procesador	red neural	<input checked="" type="radio"/>	ahorro energía	indicadores rendimiento	n/a
[Jung and Pedram 2010]	procesador	aprendizaje supervisado	<input type="radio"/>	energía	indicadores ocupación	n/a
[Lin et al. 2010]	disco duro	Q-Learning	<input type="radio"/>	energía, latencia	sistema de colas	carga sintética
[Csurgai-Horváth and Bitó 2011]	computador	red neural	<input type="radio"/>	energía	n/a	n/a
[Candrawati and Hashim 2016a] [c]	computador	Q-Learning	<input type="radio"/>	energía	componentes sistema de colas	serie temporal distribución empírica
<i>Esta tesis</i>	parque	analítico	<input checked="" type="radio"/>	energía actividad eliminada		

Tabla 2.3.: Resumen de trabajo previo de técnicas de aprendizaje automático.

2.3. Comparación de herramientas comerciales

Las herramientas comerciales, esto es, herramientas que ciertas empresas venden como paquetes de software o servicios, generalmente implementan métodos sencillos, típicamente estáticos: el administrador del parque selecciona un valor de tiempo de espera para apagar o cambiar el computador a un estado de hibernación, teniendo en cuenta información que la herramienta le presenta para guiarle en la decisión (por ejemplo, series temporales de uso o valores recomendados para el valor de tiempo de espera). Algunas herramientas presentan directamente el valor de tiempo de espera a utilizar, requiriendo al administrador confirmar la elección.

Estos valores o recomendaciones no dependen de métricas de satisfacción de usuario determinadas por modelos detallados, sino que sólo lo hacen en función del perfil de consumo energético de cada computador. En la mayoría de los casos, las herramientas son de código cerrado, por lo que no es posible examinar el método utilizado.

Aunque todas las herramientas analizadas se aplican al parque, hay que destacar cuatro grupos principales:

- Aquellas herramientas que generan políticas estáticas, de forma que una vez el administrador del parque acepta una configuración de tiempos de espera, permanecerá inalterada hasta que vuelva a generarse y aplicarse centralmente. Estas herramientas son las más comunes, siendo NightWatchman [IE NightWatchman] siendo la más relevante por cantidad de despliegues. No consideran la satisfacción, a no ser que el administrador la tenga en cuenta de forma manual.
- Aquellas herramientas que realizan agrupación de usuarios, generando políticas diferenciadas para cada grupo, tal y como hace Verdiem Surveyor [Aptean Verdiem Surveyor]. Estas herramientas están especialmente aconsejadas a parques de gran tamaño, de forma que no se debe realizar una optimización por cada usuario, sino por cada grupo. Cómo se realizan estas agrupaciones de usuarios resulta determinante para conocer la efectividad de la herramienta, aunque ningún fabricante lo hace público.
- Aquellas herramientas que ofrecen funcionalidades avanzadas para la gestión del parque, no sólo la distribución de configuraciones a cada computador, sino herramientas de medición y simulación, que permitan predecir cuál es el impacto de ajustes diversos, la relación entre los niveles de satisfacción y ahorro o los cambios previstos en las políticas cuando ciertos parámetros cambian (tales como el tamaño del parque, las horas laborables o las distribuciones de tiempos de inactividad y actividad). Tan sólo una herramienta, Tivoli [IBM Tivoli], proporcionaba parcialmente funcionalidades de este tipo.
- Aquellas herramientas que no sólo realizan apagados, sino que también son capaces de encender computadores en base a predicciones del momento en que el usuario va a retornar a su uso tras un período de inactividad, lo que

ciertamente tiene un impacto muy positivo en la sensación de satisfacción del usuario. En este caso, las herramientas como NightWatchman [1E Night-Watchman], que realizan encendidos, lo hacen de forma estática, es decir, con temporizadores fijos para, por ejemplo, el inicio de la jornada diaria, pero no calculando de forma dinámica la hora del siguiente uso.

Este conjunto de herramientas estudiado es muy bueno en los aspectos de implementación de políticas una vez generadas: desde la distribución de las mismas a los computadores del parque a la aplicación de las mismas en tiempo de ejecución. Ofrecen también buenos mecanismos de recolección, agregación y presentación de informes sobre el estado del parque, así como integración con otros sistemas de las empresas.

La tabla 2.4 resume los principales aspectos de las herramientas comerciales más comúnmente utilizadas. Todas las herramientas revisadas utilizaban métodos para generar políticas estáticas donde el administrador es responsable de seleccionar el valor de tiempo de espera deseado.

Entre las herramientas analizadas, sólo una minoría de ellas [1E NightWatchman; Aptean Verdiem Surveyor; Avob Energy Saver; Data Synergy PowerMAN; New Boundary PwrSmart] permiten realizar agrupaciones de usuarios, sólo generando o bien un modelo global para todo el parque o bien modelos completamente individuales por usuario. Tan solo una herramienta [IBM Tivoli] permite extrapolar políticas para variantes del parque donde se simulan cambios sustanciales (nuevos usuarios, edificios u otras propiedades). Sólo una herramienta [1E NightWatchman] permite ejecutar encendidos programados, de cara a tener el computador preparado para el regreso del usuario.

2.4. Técnicas que consideran la satisfacción de usuario

Finalmente, y considerando la importancia de la satisfacción de usuario en la gestión energética, se comparan a continuación los trabajos que tienen la satisfacción de usuario como base para la optimización que llevan a cabo.

El primer trabajo que consideró la satisfacción de usuario como variable central fue [Flinn and Satyanarayanan 1999]. En este caso la gestión energética está dirigida por las aplicaciones, de esta forma, cada aplicación define los parámetros de calidad de servicio que necesitan, y el sistema determina el comportamiento más adecuado para ellos. Consiguen reducir el consumo hasta un 30 %, pero requiere que todas y cada una de las aplicaciones sean instrumentadas para determinar una clase de servicio (presentan cuatro ejemplos: navegador web, reconocimiento de voz, reproducción de vídeo y sistema de mapas). En esta investigación, los autores se centran en el consumo de batería como medida equivalente a la satisfacción de usuario, entendiendo que una mayor longevidad de una carga es positivo. Aunque no se trate de una métrica que describe directamente la satisfacción, su investigación

2.4. Técnicas que consideran la satisfacción de usuario

Herramienta	Entorno	Satis.	Encendido	Política	Agrup. usuarios	Modelos dinámicos	Extrap. modelos
IE Night Watchman	parque	<input type="radio"/>	<input checked="" type="radio"/>	estático	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Autonomic Software ANSA	parque	<input type="radio"/>	<input type="radio"/>	n/a	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Avob Energy Saver	parque, DC	<input type="radio"/>	<input type="radio"/>	n/a	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Certero PowerStudio	parque	<input type="radio"/>	<input type="radio"/>	estático	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Data Synergy PowerMAN	parque	<input type="radio"/>	<input type="radio"/>	estático	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Dell KACE	parque	<input type="radio"/>	<input type="radio"/>	n/a	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
EventZero Greentrac	parque	<input type="radio"/>	<input type="radio"/>	n/a	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Faronics Power Save	parque	<input type="radio"/>	<input type="radio"/>	estático	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
IBM Tivoli	parque	<input type="radio"/>	<input type="radio"/>	estático	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Ivanti Power Manager	parque	<input type="radio"/>	<input type="radio"/>	estático	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
New Boundary PwrSmart	parque	<input type="radio"/>	<input type="radio"/>	estático	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aptean Verdiem Surveyor	parque, DC	<input type="radio"/>	<input type="radio"/>	estático	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Verismic Power Manager	parque	<input type="radio"/>	<input type="radio"/>	estático	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>Esta tesis</i>	parque	<input checked="" type="radio"/>	<input type="radio"/>	dinámico	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>

Tabla 2.4.: Resumen de herramientas comerciales para la gestión energética.

estudia el concepto de satisfacción y selecciona métricas para optimizar en base a ella.

Ha habido aproximaciones posteriores al problema de definir qué es la satisfacción de usuario, por ejemplo, una medida común para medirla es Application Performance Index (*Apdex*) [Sevcik 2005]. Esta métrica define conceptos novedosos, como la posibilidad de determinar satisfacción parcial; sin embargo no es apropiada para gestión energética ya que se centra exclusivamente en la latencia de servicio de peticiones transaccionales.

Theocharous [Theocharous et al. 2006] definió una métrica de insatisfacción de usuario y utilizaron métodos de aprendizaje automático (regresión logística, C4.5, k-nearest) para aprender políticas que encuentran un balance entre insatisfacción y consumo ahorrado. La definición de la insatisfacción de este trabajo tiene cierta relación con la utilizada en esta tesis; también encontraron que la relación entre la insatisfacción y el consumo ahorrado sigue una relación similar a una función logarítmica. Determinaron que con objetivos de insatisfacción menores al 5%, los ahorros de energía se encontraban entre el 20% y el 50%. Su trabajo se aplica a un único computador y presenta cierta variabilidad en los ahorros y además precisa de un modelo de Markov de bastante complejidad.

Shye, en un trabajo posterior, [Shye et al. 2008] estudió la relación existente entre la satisfacción de usuario con diferentes indicadores de rendimiento del sistema. Demostraron que ambos tienen una relación no lineal y presentaron un modelo basado en una red neuronal que traduce indicadores de rendimiento a políticas energéticas. Pese a que consiguieron ahorrar un 25% de consumo, la satisfacción no es consistente con los ahorros obtenidos.

De nuevo el grupo de Lin [Lin et al. 2009], implementó su método con un controlador que gobierna tanto la frecuencia como el voltaje de un procesador, dirigido por el proceso actual del usuario. El controlador estima la insatisfacción del usuario en un momento dado en diferentes puntos de la curva de frecuencia y voltaje del procesador para determinar la política a aplicar. Mientras que producen ahorros de hasta el 49,9%, éste sólo se produce en cargas muy específicas y requiere de una costosa instrumentación de procesos y sistema operativo.

Setz [Setz et al. 2016] utilizó una métrica de satisfacción para incorporar retroalimentación negativa al proceso de gestión energética. Los usuarios reportarían de forma manual eventos que son insatisfactorios, de forma que un incremento de esta señal negativa haría que los valores de tiempo de espera se fueran incrementando paulatinamente. Este trabajo presenta un modelado de la actividad de usuario similar al de esta tesis, definiendo la probabilidad de tener actividad de usuario en un mecanismo de tiempos discretos (en este caso intervalos de un minuto).

La última propuesta que incorpora satisfacción de usuario es el trabajo de Turkin [Turkin and Vdovitchenko 2019], que utiliza una medida cualitativa para entender el impacto de la gestión energética mediante la valoración directa por parte de los usuarios. Esta medida se toma *a posteriori* en estudios específicos con los usuarios e influencia la generación de políticas nuevas, haciendo que la adaptación del sistema a la satisfacción indicada por los usuarios resulte muy lenta.

Finalmente, es conveniente repasar las herramientas comerciales, que presentan

2.4. Técnicas que consideran la satisfacción de usuario

una gran solidez en lo que a distribución de configuraciones a los equipos del parque se refiere, pero no plantean métricas de satisfacción para el cálculo de las políticas energéticas.

Capítulo 3.

Modelado del comportamiento y satisfacción de los usuarios

En este capítulo se presenta el modelo propuesto para la representación del comportamiento de los usuarios del parque, la caracterización de la satisfacción y el método para el cálculo de políticas energéticas basadas en ésta.

3.1. Comportamiento de los usuarios

Los usuarios de los parques se comportan de diversas maneras, teniendo diferentes hábitos y preferencias de uso de sus computadores. Para modelar su comportamiento, se define un conjunto de acciones que pueden realizar en un momento dado: usar activamente su computador, mantenerlo inactivo pero encendido o apagarlo de forma explícita. Asimismo, cuando existe un gestor de energía en los computadores, este puede apagarlos de forma automática tras un determinado periodo de inactividad. Estos son los cuatro estados en los que un computador puede encontrarse en cualquier momento:

- Encendido, con el usuario haciendo uso del mismo (*activo*).
- Encendido, sin actividad de usuario (*inactivo*). En este estado, el computador consume una cantidad similar de energía que cuando está en uso activo.
- Apagado, determinado por el usuario (*apagado de usuario*). En este estado, el computador consume una cantidad mínima de energía, cercana a cero.
- Apagado, determinado por el gestor de energía (*apagado automático*). Esto solo ocurre cuando el gestor de energía está habilitado, el computador consume una cantidad de energía cercana a cero.

La figura [3.1](#) muestra un ejemplo ilustrativo de un parque sin ninguna gestión energética. Los intervalos en un parque real son típicamente más cortos, especialmente en el caso de los intervalos de actividad.

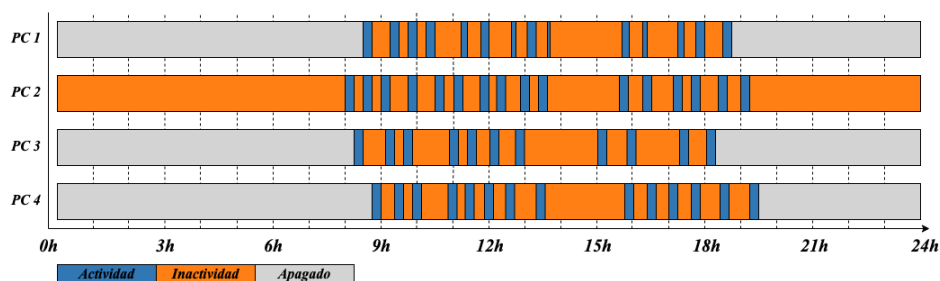


Figura 3.1.: Ejemplo de parque con los estados sin gestión energética.

El ciclo típico de uso de un computador se alinea con la jornada laboral. En el ejemplo, comienza entre las 8:00 y las 9:00 y termina hacia las 18:00. Los intervalos de actividad se turnan con los de inactividad a lo largo del día: los usuarios están escribiendo, moviendo el ratón o realizando otras actividades; pero también paran para hacer descansos, leer o tener reuniones.

Durante la jornada cada usuario presenta un comportamiento diferente, reflejando los diferentes patrones de uso individuales. No obstante, se observa como ciertas inactividades se alinean, por ejemplo los descansos o el parón a mitad de jornada.

Hacia el final de la jornada es cuando la mayoría de los computadores se apagan. Existen ciertos usuarios que no apagan su computador tras la jornada laboral, generando largos intervalos de inactividad durante la noche (como el *PC 2*).

Establecer políticas de gestión de energía permite reducir el tiempo de inactividad, donde los computadores se encuentran encendidos, consumiendo energía, pero sin un uso activo.

La figura 3.2 muestra el mismo parque, pero con el gestor energético habilitado. Éste apaga computadores tras un cierto periodo de inactividad, determinado por las políticas energéticas. De esta forma, una gran parte de la inactividad del parque se elimina, ahorrando la energía correspondiente. Esto es especialmente importante en computadores que de otra forma pasarían noches enteras inactivos, o incluso días completos durante el fin de semana.

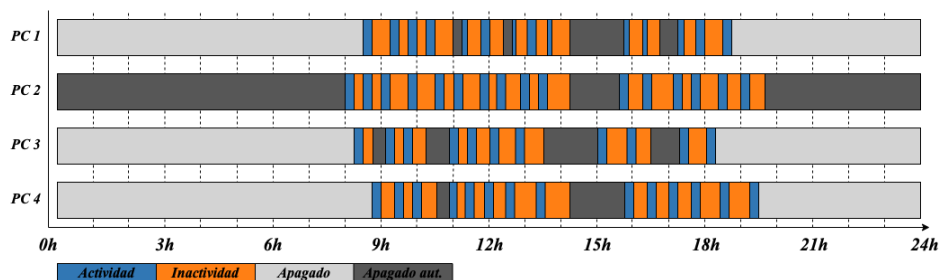


Figura 3.2.: Ejemplo de parque con los estados con gestión energética.

Los intervalos mostrados en las figuras 3.1 y 3.2 se generan capturando conti-

nuamente los eventos de activación y fin de actividad de los computadores que componen el parque en estudio, de forma que se pueden reconstruir y estudiar. Para resolver este problema, se debe contar con un sistema que capture y procese trazas de utilización de dichos computadores, mientras los usuarios los utilizan con normalidad. La sección 4.3 proporciona detalles sobre el sistema de recolección de registros.

Para conocer cómo se comportan los usuarios, se ha estudiado un parque real de una empresa y otro de aulas correspondientes a unos laboratorios de informática de la Universidad de Oviedo, recolectado registros de uso de los computadores mediante un agente instalado en cada uno de ellos que muestreaba el estado del computador con una cierta frecuencia de monitorización. Estos registros se envían a un servidor central que los procesa y almacena, generando una traza de eventos que permite reconstruir todos los intervalos de actividad e inactividad, así como los intervalos durante los cuales los computadores estuvieron apagados.

Es importante notar que la recolección de estos registros de utilización se debe realizar con cualquier mecanismo de control energético desactivado (tales como las funciones de apagado predeterminadas de los sistemas operativos), por dos motivos. En primer lugar, se busca conocer el comportamiento nativo de los usuarios y dichos sistemas pueden distorsionarlo (por ejemplo, durante periodos muy largos como un fin de semana, dónde algunos usuarios dejan sus computadores encendidos). En segundo lugar, para poder estimar con precisión el impacto en la satisfacción que tiene un hipotético control de energía, informando a los usuarios en qué momentos potencialmente su computador se encontraría apagado y contrastando su opinión.

Registro	Fechas	Usuarios	Duración	Notas
1	5/10 - 30/11, 2014	265	7,9 sem.	Sin portátiles
2	1/4 - 29/5, 2015	64	8,1 sem.	Con portátiles, min. 50 eventos

Tabla 3.1.: Registros recolectados de un parque real.

Los datos, resumidos en la tabla 3.1 fueron recolectados durante dos intervalos no consecutivos de dos meses. Con objetivo de establecer cuáles son los patrones de actividad, y si éstos dependen de ciertas variables, tales como la hora del día, se han analizado dos variables inicialmente: la duración y cantidad de los intervalos de inactividad que comienzan en cada hora del día.

Uno de los conjuntos de datos no incluye ordenadores portátiles, mientras que el otro sí lo hace. Esta dicotomía permite entender las diferencias de comportamiento y su impacto en la generación de políticas.

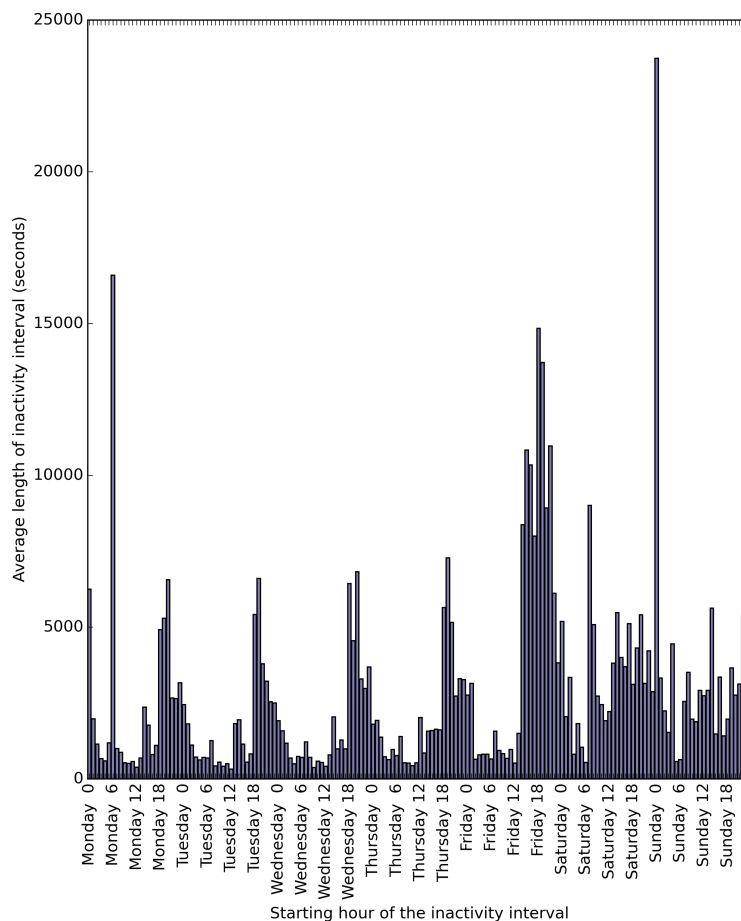


Figura 3.3.: Duración media de los intervalos de inactividad por hora del día y semana.

La figura 3.3 representa la duración media de los intervalos de inactividad que comienzan en cada una de las 168 horas de la semana, independientemente de si sobrepasan la hora de duración. La resolución es de una hora, pero intervalos de inactividad pueden, y frecuentemente así ocurre, superar la hora de duración, en cuyo caso se contaría con varias horas con ausencia de nuevos eventos ya que el computador está el mismo estado durante toda la hora.

Se ha probado con resoluciones temporales más granulares que una hora, pero en ninguna salvo en la horaria se aprecian estos patrones de interés. En particular, es relevante ver el comportamiento cíclico, con intervalos de duración más larga por las tardes (correspondientes a usuarios que dejan sus computadores encendidos toda la noche) y de muy larga duración específicamente los viernes por la tarde (correspondientes al fin de semana). También se observan patrones diferentes para

el fin de semana, pero con una mayor longitud de los intervalos.

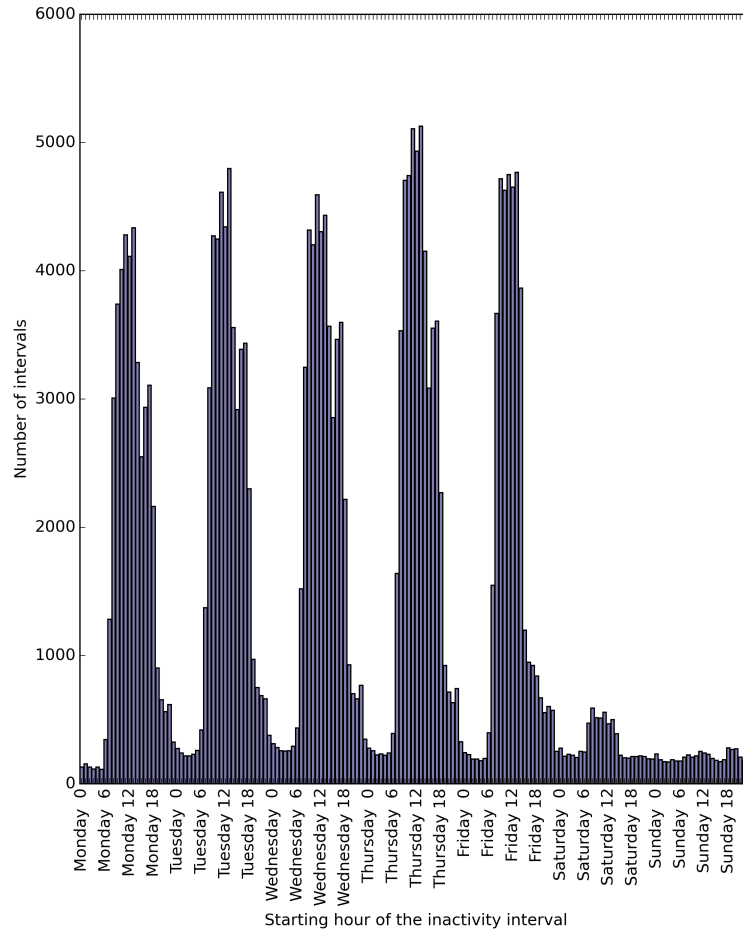


Figura 3.4.: Cantidad de intervalos de inactividad por hora del día y semana.

La figura [3.4](#) proporciona otro punto de vista, indicando la cantidad de intervalos de inactividad que comienzan en cada hora de la semana. Como se observa, se concentran durante las horas laborables, de forma que durante éstas hay numerosos intervalos cortos, mientras que en el resto de la semana se detectan intervalos de muy larga duración, pero en un número mucho menor, al estar los computadores inactivos por tratarse de, por ejemplo, un fin de semana.

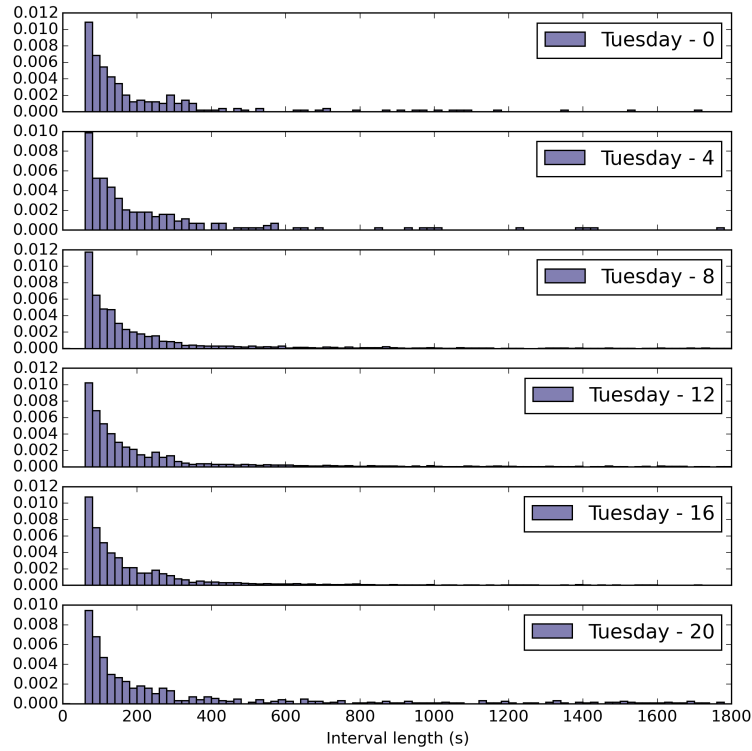


Figura 3.5.: Distribución de la duración de los intervalos de inactividad en diferentes horas de un martes.

Finalmente, es interesante estudiar la distribución de la duración de dichos intervalos dentro de una hora. La figura 3.5 muestra histogramas de intervalos de inactividad en varias horas de un día laborable (martes). Es remarcable la presencia de intervalos de gran duración en la cola de las distribuciones de horas no laborales (0:00, 4:00 y 20:00) en relación al resto de horas.

Hay que notar la ausencia de intervalos de menos de 60 segundos, dado que se eliminan al registrarlos por los agentes en el servicio central. Estos intervalos no se presentan dado que es la resolución máxima de recolección establecida para el agente.

En el apéndice A se proporcionan detalles sobre el estudio de los datos y los diferentes experimentos de ajuste que se han realizado.

3.1.1. Modelado de la actividad e inactividad

Dos variables son fundamentales para modelar el comportamiento de los usuarios de cara a la gestión energética de los computadores que utilizan: la actividad y la inactividad. Aunque una variable parezca la complementaria de la otra a primera

vista, poseen características diferentes.

La actividad consiste en interacciones del usuario con el computador, que típicamente resultan muy breves: una pulsación de teclas, un movimiento del ratón u otro periférico. Estas interacciones han de ser inducidas por el usuario, de forma que otras actividades tales como notificaciones no resultan directamente en actividad a menos que el usuario las active.

Otros momentos interesantes son acciones que requieren tener el computador activo, pero no conllevan una interacción directa durante un tiempo más o menos largo, por ejemplo visualizar un vídeo. El usuario iniciará el vídeo con una interacción como una pulsación de teclas o un movimiento de ratón y a continuación se observará un periodo largo de inactividad, pero que requerirá el computador activo. En estos casos, el software de reproducción multimedia mantendrá bloqueado la suspensión o apagado del computador por lo que no será necesario considerar casos especiales al respecto.

En general, se pueden definir cuatro propiedades concretas de la actividad y la inactividad:

- La duración de los intervalos de actividad es muy pequeña y ocurren con mucha frecuencia, generalmente de unos pocos segundos o minutos: el usuario escribe, o mueve el ratón y para durante unos pocos segundos a leer u observar la interfaz de usuario antes de continuar con otro intervalo de actividad. Si bien se pueden recoger eventos a la resolución temporal que se desee (por ejemplo, cada pulsación de teclado) contar con decenas de miles de intervalos de muy corta longitud no aporta mucho valor y es recomendable fusionarlos en intervalos más largos, que representen una unidad razonable para un ser humano. En esta tesis se ha probado a utilizar los intervalos en bruto y fusiones de 1 minuto de duración mínima, obteniendo mejores resultados con los intervalos fusionados (en términos de facilidad de ajuste, explicabilidad y estabilidad de las políticas generadas).
- Los intervalos de inactividad son infrecuentes, pero mucho más largos que los de actividad: en el mejor de los casos duran varios minutos (por ejemplo cuando el usuario se ausenta momentáneamente) y en el peor varios días (por ejemplo todo un fin de semana). La larga duración de estos intervalos presenta la oportunidad de ahorro energético: convertir tiempo durante el cual el computador está sin uso en tiempo durante el cual está apagado se traduce en eliminación de consumo energético.
- La distribución de la duración de los intervalos, especialmente los de inactividad, presenta patrones cíclicos debido a los hábitos de trabajo del usuario: habrá ciertas pausas en momentos similares y la jornada presentará una duración repetible día tras día. Esta propiedad permite predecir cuál puede ser la duración de un intervalo en un determinado momento del día.
- A la suma del tiempo de actividad e inactividad hay que incorporar el tiempo durante el cual el computador está apagado. Existen dos agentes que pueden

apagar el computador: el propio usuario y el sistema de gestión energética que se aplique al computador o el parque. Sobre el primero no se tiene control y por ese motivo, además de modelarlos de forma separada, es necesario registrarlo en los registros de utilización. Sobre el segundo, su comportamiento es precisamente el objeto de las políticas energéticas generadas.

Sistemas operativos como Microsoft Windows, Linux o macOS proporcionan APIs que permiten monitorizar los tiempos de interacción del usuario, como `GetLastUserInput()`. Usando estas funciones se puede conocer con alta resolución el tiempo que ha transcurrido desde la última entrada procedente de teclado o ratón y así inferir la duración de los intervalos de actividad. Para ello, se obtiene el tiempo de la última interacción de usuario y, mientras no se modifique, se asume que se el computador se encuentra en un intervalo de inactividad. El muestreo continúa hasta que la marca temporal de entrada de usuario cambia, en ese caso, se cuenta con la confirmación de que la inactividad ha concluido y se ha pasado a un período de actividad.

Esta monitorización por parte del agente se puede realizar con la resolución temporal que se desee, ya que el API informa de los intervalos de tiempo resoluciones de hasta 10 ms. No obstante, muestrear con esta frecuencia (hasta 100 veces por segundo) no es práctico por dos motivos: genera demasiada carga en el agente y no proporciona una ganancia de información relevante, ya que cada intervalo, por corto que sea, puede estar en el entorno de 30 a 60 segundos como mínimo. Hacer un muestreo de menos frecuencia es suficiente para reconstruir la casi totalidad de intervalos de actividad. Se ha escogido una frecuencia de 60 segundos.

El agente, en su bucle de muestreo, ha de reconstruir los eventos antes de enviarlos al nodo central que los recolecta. El caso de un intervalo de inactividad es muy claro: si por ejemplo, hay un intervalo de inactividad en progreso, el valor indicado por `GetLastUserInput()` permanecerá invariable desde la última actividad de usuario, de forma que mientras el retorno de la llamada al API no varíe, el intervalo no ha concluido y no debe reportarse. Asimismo, cuando cambia el tiempo de la última entrada de usuario, hay que delimitar la finalización del mismo de forma inversa: se estará en un intervalo de actividad mientras el valor reportado por el API siga cambiando. Es, por tanto, labor de este agente determinar mediante un conjunto de reglas y heurísticos las duraciones de los intervalos de actividad e inactividad de la forma más precisa posible.

La figura 3.6 ilustra este procedimiento. En el ejemplo se cuenta con unos intervalos de inactividad (I) y de actividad (A), las flechas verticales muestran cuándo el agente realiza llamadas a `GetLastUserInput()` de forma regular. Durante todo el primer intervalo de inactividad, se seguirá recibiendo el mismo valor, indicando que la inactividad sigue ocurriendo y el intervalo no ha terminado. En cuanto se presenta una actividad, el valor retornado por el API cambia y el agente detecta el intervalo de actividad, siguiendo el mismo proceso recurrentemente.

Es necesario subrayar cómo los eventos de actividad son los que permiten detectar los finales de los eventos de inactividad. Para reconstruir todos estos eventos, el agente procesa los datos y mantiene una ventana deslizante de tiempos

de `GetLastUserInput()` que permite ir categorizándolos.

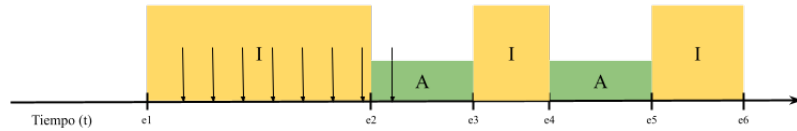


Figura 3.6.: Esquema de monitorización para detección de eventos.

El agente así mismo debe controlar ciertos casos especiales, como son los apagados que el usuario inicia. En este caso, puede recibir una notificación por parte del sistema operativo y utilizarla como señal de finalización del intervalo en curso y para registrar el inicio de un intervalo de apagado, que ha de ser reportado al nodo central para su registro.

Como se puede deducir, el período de muestreo resulta una variable importante con la que configurar los agentes que recogen los datos de utilización: determina la duración mínima de cualquier intervalo detectable y tanto la carga de los agentes como el volumen de datos almacenado. Escoger frecuencias muy altas (inferiores a 1 s) proporcionan rendimientos decrecientes, ya que los intervalos de actividad no son tan cortos, ni enriquecen lo suficiente los modelos. Frecuencias más bajas que una muestra por minuto pueden empezar a ocultar intervalos de actividad de otra forma relevantes.

Con todas estas propiedades, se ha decidido modelar el comportamiento usando cuatro variables: actividad, inactividad, apagados dirigidos por el usuario y apagados dirigidos por el gestor energético. Las tres primeras se recogen en las trazas de utilización en todo caso, mientras que la última sólo lo hace cuando el control energético está activo (es, de hecho, el producto del gestor energético).

Se han estudiado como variables estadísticas de cara a poder establecer un modelo que permita tanto resumirlas para su uso en la generación de políticas energéticas como realizar predicciones sobre su posible valor en un momento dado una vez no se están recogiendo registros, sino gestionando activamente un parque. Los modelos de las cuatro variables deben, además, ser flexibles para permitir tanto una personalización completa para cada usuario como una gestión eficiente del parque a nivel global y generar políticas energéticas de forma rápida.

El primer análisis de los datos recogidos experimentalmente muestra cómo el comportamiento de los usuarios se repite con un ciclo de una semana. Por este motivo, el modelado de cada usuario se hará usando un ciclo de una semana de longitud, partida en franjas de una hora (168 horas en total).

Para cada una de estas 168 franjas, se modela el comportamiento de usuario en base a cuatro variables: la actividad, la inactividad, el tiempo que el usuario ha apagado el computador y la probabilidad de que lo apague. Asimismo, las políticas energéticas se definen para cada una de estas franjas, de forma que se pueden ajustar al comportamiento en cada una de las horas.

Almacenar todos los eventos que ocurren en el parque, esto es, cada vez que se

produce un cambio de estado (por ejemplo, cuando un usuario apaga su computador, o cuando se detecta actividad) requiere procesar tantos eventos como computadores tenga el parque y sea la frecuencia de monitorización. Un parque de 10.000 computadores con una frecuencia de monitorización de 60 s generaría 10.000 entradas por minuto (o 167 entradas/s). Estos datos se acumulan hasta $5,25 \times 10^9$ eventos/año. Para poder analizarlos de forma rápida, se ajusta una distribución estadística a los datos, que posteriormente se usará para generar las políticas energéticas.

La función de densidad (*probability density function*, PDF) de una distribución indica la probabilidad de diferentes resultados de un experimento. En el caso de los intervalos de actividad o inactividad, representa la probabilidad de que ocurra un intervalo de cierta duración.

En este trabajo, se ha determinado que la duración de los intervalos de actividad e inactividad siguen una función exponencial, donde los intervalos cortos tienen mucha más probabilidad de ocurrir. La figura 3.7 muestra las PDF de los intervalos de inactividad de varias franjas horarias de un martes de todos los computadores. Para cada hora, se calcula el histograma de los intervalos registrados y se representan en escalas logarítmica, al tratarse de datos que se ajustan a una ley exponencial. El eje de abscisas se dispone en segundos, ya que mide la duración de los intervalos, mientras que el de ordenadas indica la probabilidad relativa.

Como ambos ejes están en escala logarítmica, una función exponencial perfecta se mostraría como una línea recta, que se marca en la figura para observar las desviaciones. Hay dos puntos a remarcar en la figura: en primer lugar se observan discontinuidades, debidas a la ausencia de datos para intervalos de ciertas duraciones; en segundo lugar existen desviaciones importantes sobre la línea que marca una ley exponencial perfecta, especialmente en las colas de las distribuciones.

La falta de datos que permitan cubrir todo el espectro posible de longitudes de intervalos se pueden mitigar usando distribuciones continuas o consolidando datos de varias horas o usuarios, de forma que se cuenta con una distribución más detallada. En el caso de las desviaciones sobre una ley exponencial, es necesario encontrar un mejor ajuste que represente de forma correcta las colas de las distribuciones, de otra forma se pueden representar incorrectamente al realizar el análisis y la simulación [Medrano Llamas et al. 2015].

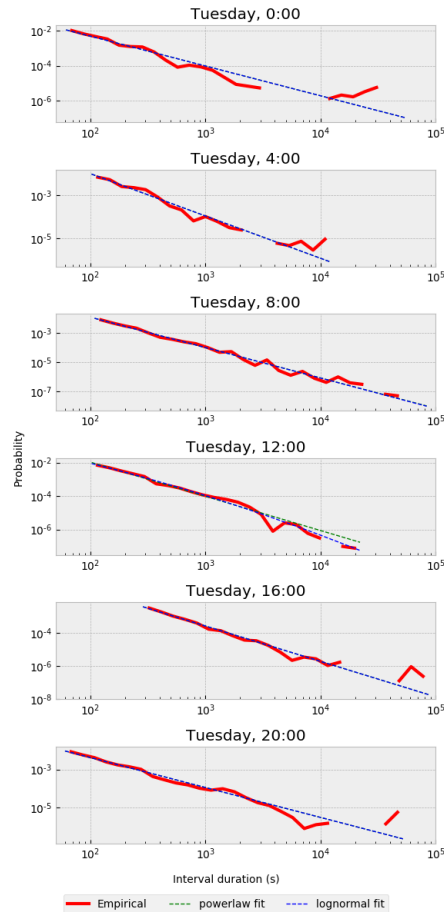


Figura 3.7.: PDF de la inactividad en escala logarítmica.

Se han probado varias distribuciones de probabilidad, como las del paquete `powerlaw` [Alstott et al. 2014](#). Aunque a primera vista las distribuciones de Pareto y log-normal proporcionan un buen ajuste, los resultados son decepcionantes por lo peculiar de la cola de la distribución: tal y como se observa en la figura [3.7](#), existe una masa de probabilidad que se corresponde con, por ejemplo, los usuarios que dejan su computador encendido durante las noches o fines de semana (en la figura, 0:00, 16:00 y 20:00).

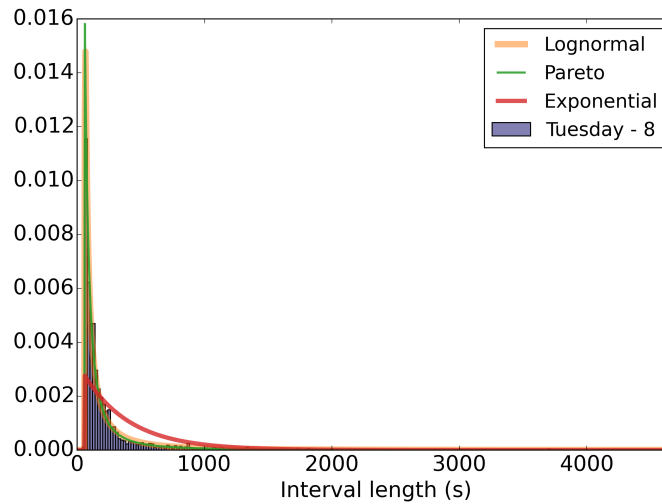


Figura 3.8.: Comparación del ajuste con distribuciones log-normal, Pareto y exponencial.

La figura [3.8](#) muestra de forma clara la gran diferencia en ajuste que se puede producir entre diversas funciones de la familia de leyes exponenciales, especialmente para este caso la distribución exponencial distorsiona mucho el modelo, al no capturar correctamente ni la masa de probabilidad de intervalos cortos ni la cola del histograma.

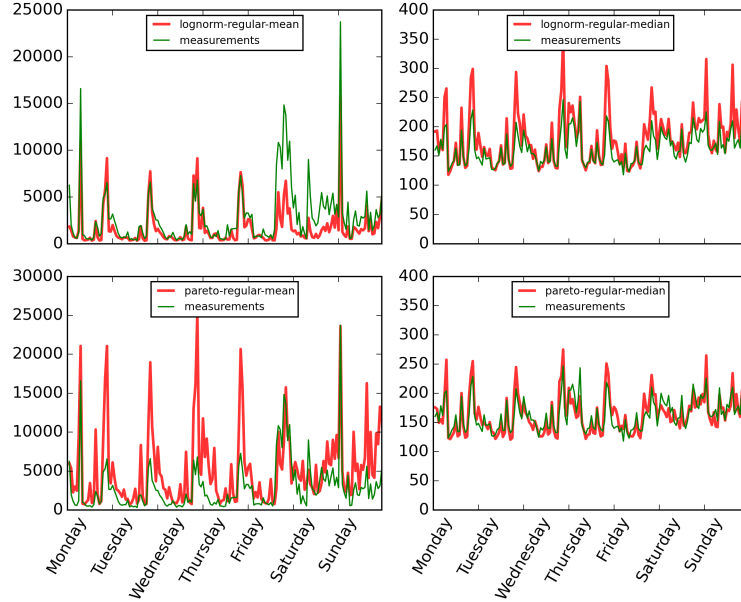


Figura 3.9.: Comparación de la longitud de los intervalos por hora del día entre los registros de utilización y los generados con un modelo log-normal y Pareto.

El efecto de usar un modelo basado en una distribución u otra puede verse en la figura 3.9, que muestra la media y mediana de los intervalos de inactividad procedentes de los datos obtenidos en los registros de utilización y las generadas mediante simulación usando distribuciones Pareto y log-normal. Mientras que las medianas presentan un buen ajuste durante toda la semana, las medias se desvían notablemente en el caso de la distribución Pareto durante días laborables, indicando la generación de intervalos mucho más largos de lo deseado, debido al mayor peso de la cola en el proceso de ajuste. Por contra, el modelo log-normal tiene un mejor ajuste, pero también presenta variaciones durante el fin de semana, generando intervalos más cortos de lo deseable al ponderar más los intervalos cortos durante su ajuste.

Esta cola es difícil de ajustar con una función de la familia de leyes exponenciales estándar, lo que puede generar intervalos con una longitud mucho más larga o más corta que la observada al usar el modelo para predecir el comportamiento de usuario. Al presentar estas grandes variaciones, se puede producir un efecto de acumulación durante una simulación, retrasando todos los intervalos, y distorsionando el resultado considerablemente.

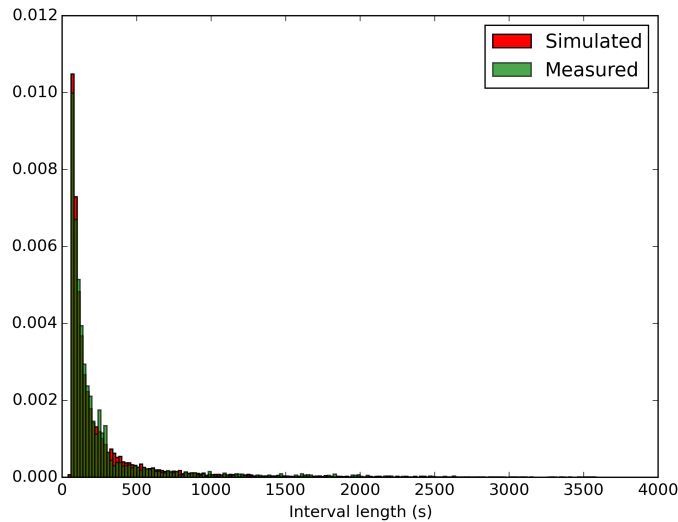


Figura 3.10.: Comparación entre datos medidos y generados mediante el modelo (martes, 12:00).

Se incluye también la figura 3.10, que compara los datos generados mediante un modelo para el martes a las 12:00 basado en una distribución log-normal con los datos de los registros de utilización. Es notable cómo el modelo no recoge la forma del histograma en duraciones de entre 250 y 300 s.

Por estos motivos, se ha usado un ajuste con una distribución empírica, que permite adaptarse mejor a estas particularidades. Se han realizado tests de bondad de ajuste con el método de Kolmogorov-Smirnov [Massey Jr. 1951] tanto para los intervalos de actividad como para los de inactividad, mostrando un ajuste adecuado para niveles de confianza del 99,9%.

Respuesta a la pregunta A.2: ¿Cuáles son los mejores modelos para el comportamiento de los usuarios?

Tal y como expresa este capítulo se han estudiado diversas opciones para modelar el comportamiento de los usuarios, resultado la que ofrece una mayor sencillez y facilidad para calcular políticas energéticas un modelo estocástico donde se representa el comportamiento de los usuarios mediante distribuciones ajustadas en base a datos recopilados en los parques a gestionar antes de establecer políticas energéticas.

La selección de las distribuciones es crucial de cara a representar correctamente las particularidades del comportamiento de los usuarios, tales como los intervalos de larga duración de los fines de semana y la noche.

3.2. Gestión de energía dirigida por la satisfacción de usuario

El método presentado en esta tesis realiza la optimización energética generando políticas para el gestor de energía integrado en el sistema operativo. Estas políticas controlan el valor del tiempo de espera que el gestor de energía utiliza para apagar el computador o ponerlo en un estado intermedio (suspensión o hibernación). Estas políticas minimizan el consumo energético manteniendo restricciones de niveles mínimos de satisfacción de usuario establecidos por el administrador del parque.

Establecer un valor de tiempo de espera (*timeout*) para el gestor de energía del sistema operativo requiere determinar cuánta inactividad debe eliminarse. Si, por ejemplo, el administrador tiene como objetivo eliminar, al menos, el 25 % del tiempo de inactividad, sería suficiente con establecer el valor del tiempo de espera al valor que acumula el 75 % de la duración total de los intervalos de inactividad. Este cálculo puede realizarse en base a la distribución de los intervalos de inactividad.

Es conveniente hacer notar que, especialmente en distribuciones con colas muy largas, donde las observaciones más grandes pueden suponer un porcentaje importante del tiempo total de inactividad, puede que este cálculo conlleve variaciones grandes en los valores obtenidos, dado que pequeños cambios en el objetivo de ahorro pueden traducirse en grandes variaciones en el valor de tiempo de espera y viceversa.

Establecer políticas de gestión energética de esta forma no considera el impacto en la usabilidad del computador y, por tanto, en la satisfacción del usuario. Dependiendo del perfil de uso y del objetivo del administrador, podría resultar en un computador completamente inutilizable, que requiera tener que encenderlo muy frecuentemente, cada vez que se vuelve de un periodo de inactividad.

En la figura [3.11](#) se representan los intervalos de inactividad de un parque ordenados de forma creciente por su duración. Escoger un valor del tiempo de espera permite ahorrar toda la energía correspondiente al consumo del parque para todo el tiempo de inactividad superior al mismo. Para el parque de la figura, la selección 1 permite un ahorro notable, sin embargo, eso comporta una baja satisfacción: en casi todos los intervalos, el usuario encontraría su computador apagado y debería esperar a recuperar su sesión. La selección 2 incrementa la satisfacción de forma que en aproximadamente la mitad de los casos el usuario no encuentra un computador apagado. La selección 3 maximiza la satisfacción, pero hace que el ahorro energético sea mínimo.

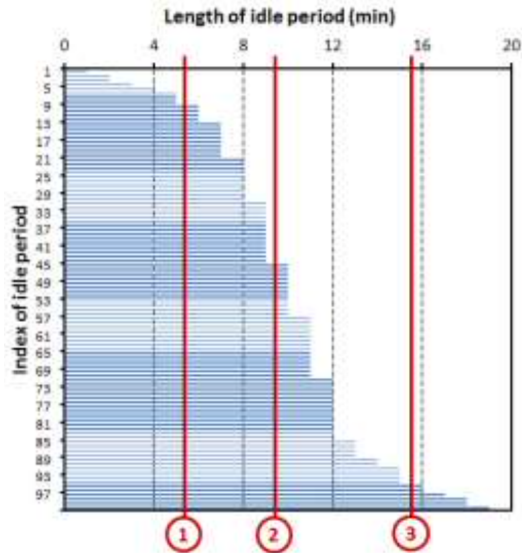


Figura 3.11.: Elección de diversos valores del tiempo de espera para un conjunto de intervalos de inactividad.

La relación entre el ahorro energético y el tiempo de espera no es siempre lineal: incrementar los valores del tiempo de espera incrementa la satisfacción y reduce el ahorro energético, pero no en la misma proporción.

En este trabajo, el ahorro energético se mide como la proporción de tiempo de inactividad que puede eliminarse apagando los computadores, de forma que en lugar de estar encendidos y sin uso, están en un estado de bajo consumo (el cálculo se detalla en la sección 3.2.2).

En la figura 3.12 se presenta un ejemplo ilustrativo de la relación entre la satisfacción y la inactividad eliminada. Se puede observar cómo incrementos del valor de tiempo de espera tienen un rápido impacto hasta obtener valores de satisfacción cercanos al 90%: cambiar del valor de tiempo de espera marcado con 1 al marcado con 2 supone incrementar la satisfacción desde el 10% al 50%. Cambiarlo al marcado con 3 permite llegar a más del 90%.

El aumento es rápido hasta que finalmente se entra en un rango de valores elevados del tiempo de espera (superiores a 12 minutos) para los que no se consiguen grandes incrementos del ahorro energético ni tampoco de la satisfacción del usuario.

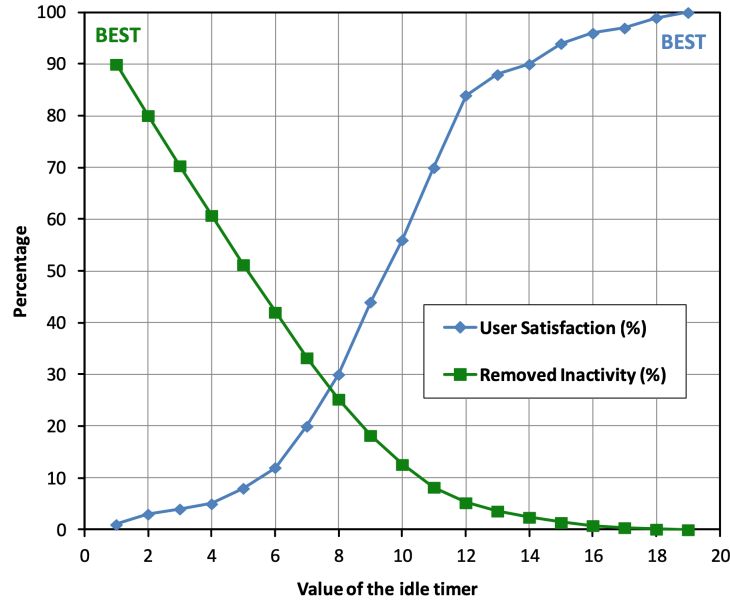


Figura 3.12.: Comparación del ahorro energético y la satisfacción de usuario.

La satisfacción de usuario es una propiedad fundamental para obtener una gestión energética amigable, pues en caso contrario, los usuarios buscarán formas de desactivar el gestor energético.

3.2.1. Satisfacción de usuario

Para calcular la satisfacción, se recogen datos de utilización del parque compuestos por registros de intervalos de actividad e inactividad. Estos registros han sido analizados para determinar cuál es la métrica más relevante para caracterizar la satisfacción, y cómo utilizarla para determinar políticas de gestión energética.

La satisfacción se determina cada vez que el usuario vuelve a usar su computador una vez se produce un evento de apagado durante un periodo de inactividad. Un apagado se considera insatisfactorio si el usuario debe reactivar el computador al volver de la inactividad, y satisfactorio en caso contrario. El ratio de satisfacción se mide como el ratio de intervalos de apagado satisfactorios sobre el total de intervalos de inactividad [García et al. 2014](#).

Para definir de manera formal la satisfacción de usuario para un valor del tiempo de espera de apagado (T), se define I_i como la duración del intervalo de inactividad i -ésimo de una serie de n intervalos de inactividad. I_i será considerado como insatisfactorio si su duración es mayor o igual que T y satisfactorio en caso contrario.

En la ecuación [3.1](#) y la figura [3.13](#) se representa la función de satisfacción para

el intervalo i -ésimo, $s(I_i)$, que toma valores de 1 (satisfecho) si no hay apagado o 0 (insatisfecho), cuando la duración de la inactividad es superior al tiempo de espera y, por tanto, el usuario debe volver a activar su computador. Los valores de $s(I)$ se pueden promediar para obtener la satisfacción agregada, S , mostrada en la ecuación 3.2

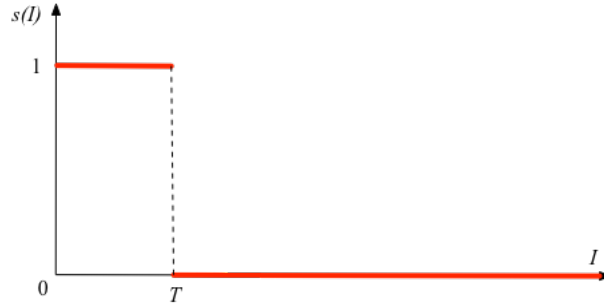


Figura 3.13.: Función escalón para la satisfacción de usuario, $s(I)$.

$$s(I_i) = \begin{cases} 1 & \text{if } I_i < T \\ 0 & \text{if } I_i \geq T \end{cases} \quad (3.1)$$

$$S = \frac{\sum_{i=1}^n s(I_i)}{n} \quad (3.2)$$

Con esta definición, cualquier intervalo de inactividad mayor que el tiempo de espera genera una nula satisfacción, tenga la duración que tenga. Cualquier inactividad se considera igual, dure diez minutos o diez horas. Sin embargo, el impacto en la satisfacción del usuario no es el mismo cuando tiene que volver a activar su computador tras un breve periodo de tiempo o al día siguiente.

Usar la definición de satisfacción de la ecuación 3.2 es además susceptible al ruido que pueden introducir los intervalos muy largos e infrecuentes, tales como los que generan los usuarios que dejan los computadores encendidos durante las noches (representados en las largas colas de las distribuciones de la figura 3.7). Este efecto limita asimismo la efectividad de las políticas energéticas, ya que los valores del tiempo de espera tienden a ser más altos de lo necesario.

Un método que permite ponderar la satisfacción de usuario en función de la duración de la inactividad es *Apdex*. Este método se diseñó para la estimación de la satisfacción en sistemas transaccionales, como servicios web.

Para ello introduce el concepto de la tolerancia, permitiendo tener tres estados de servicio: satisfactorio si se responde en menos de un tiempo T ; otro estado intermedio, *tolerable*, si se responden en menos de un tiempo F ; y un tercer estado, insatisfactorio, cuando el tiempo de respuesta es mayor.

Para aplicar *Apdex* a la gestión energética hay que adaptar el método, haciendo que los intervalos de inactividad que requieren que el usuario encienda su computador, pero que tienen una duración suficientemente larga, se consideren parcialmente satisfactorios o tolerables.

La ecuación 3.3 muestra la definición de *Apdex*

$$Apdex_t = \frac{\sum \text{satisfactorio} + \frac{\sum \text{tolerable}}{2}}{\sum \text{muestras}} \quad (3.3)$$

Apdex ha de ser adaptado para poder ser utilizado para la gestión energética, específicamente hay que dar una definición para los intervalos tolerables, aquellos que tienen una duración muy larga. La figura 3.14 muestra la adaptación realizada, introduciendo el parámetro W_f , que determina el umbral a partir del cual un intervalo se considera tolerable, otorgando la mitad de satisfacción que un intervalo que no requiere reactivar el computador.

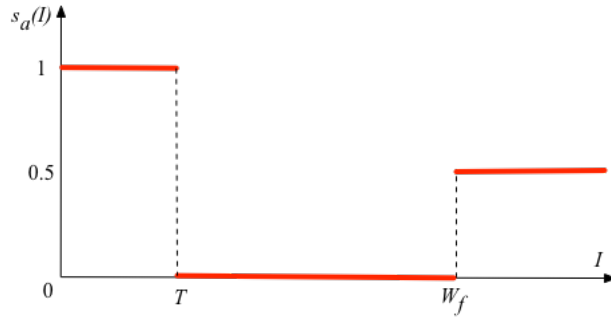


Figura 3.14.: Función de satisfacción basada en *Apdex*.

La definición de la satisfacción basada en *Apdex*, S_a , se detalla en las ecuaciones 3.4 y 3.5, en particular, $s_a(I_i)$ determina los valores de satisfacción función de si resulta satisfactorio (no requiere apagado porque no se alcanza el valor de tiempo de espera), es insatisfactorio o tolerable.

$$s_a(I_i) = \begin{cases} 1 & \text{if } I_i < T \\ 0 & \text{if } T \leq I_i < W_f \\ 0,5 & \text{if } I_i \geq W_f \end{cases} \quad (3.4)$$

$$S_a = \frac{\sum_{i=1}^n s_a(I_i)}{n} \quad (3.5)$$

El método *Apdex* tiene poca progresividad y una memoria muy intensa, especialmente en intervalos de inactividad muy largos. Por ejemplo, ante cualquier inactividad muy larga nunca se llegará a considerar satisfactoria, incluso después de varios días.

Satisfacción ponderada por las preferencias de usuario

Para comprender cómo los usuarios perciben los eventos de inactividad prolongados y cuál es su opinión al respecto de su satisfacción en estos casos, se ha realizado una encuesta en 2020 a los usuarios ($n = 49$) de los parques estudiados, preguntando cuál sería su nivel de satisfacción con diferentes duraciones de intervalos.

La intención es obtener datos que permitan describir de forma objetiva las preferencias de usuarios para definir una función que permita estimar la satisfacción de forma más realista y proporcionar más flexibilidad a la hora de determinar políticas energéticas.

Las preguntas se referían a tres tipos de pausas: cortas (entre 5 y 10 minutos), largas (más de 30 minutos) y hasta la siguiente jornada (toda la noche y/o el fin de semana). Asimismo se preguntaba sobre el impacto en la satisfacción dependiendo del tipo de apagado que se utilizaba: apagado (ACPI nivel S5) o suspendido en RAM (ACPI nivel S3).

Esta encuesta muestra cómo la satisfacción aumenta (o al usuario ya no le importa encender el computador de nuevo) conforme el intervalo de inactividad es más largo. El 7% de los usuarios consideran satisfactorio tener que reactivar el computador tras cinco minutos, mientras que el 67% lo considera satisfactorio pasada la media hora. Asimismo, el tipo de apagado tiene influencia: usando estados ACPI que no requieren que el usuario restaure su sesión (como suspensión o hibernación en lugar de apagado), la satisfacción se incrementa más, al poder recuperar el trabajo intacto tras un apagado.

Para adaptarse mejor a esta realidad, en este trabajo se propone un método para modelar la satisfacción donde ésta se pondera con la duración del intervalo de inactividad, de forma que intervalos cortos se consideran insatisfactorios, pero pasado un tiempo progresivamente se empiezan a considerar satisfactorios, hasta el momento en el que el usuario no tiene preferencia en cuanto a satisfacción.

Definiendo una función de ponderación, w , como la mostrada en la figura 3.15 y las ecuaciones 3.6 y 3.7, la satisfacción ponderada se incrementa de forma progresiva y lineal entre los puntos W_s y W_f . Cualquier intervalo más largo que W_f se considera como satisfactorio.

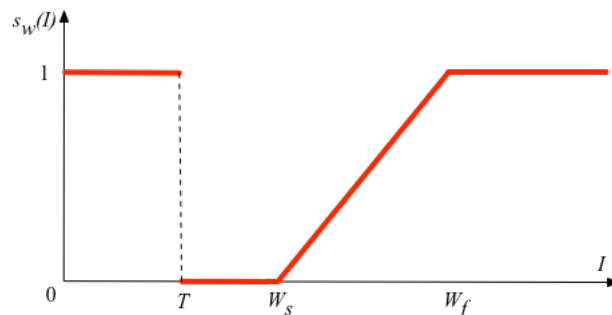


Figura 3.15.: Función de satisfacción con progresividad lineal.

$$s_w(I_i) = \begin{cases} 1 & \text{if } I_i < T \\ 0 & \text{if } T \leq I_i < W_s \\ \frac{I_i - W_s}{W_f - W_s} & \text{if } I_i \in [W_s, W_f] \\ 1 & \text{if } I_i > W_f \end{cases} \quad (3.6)$$

$$S_w = \frac{\sum_{i=1}^n s_w(I_i)}{n} \quad (3.7)$$

El uso de una función de satisfacción ponderada proporciona una métrica más realista de satisfacción. Esta métrica tiende a converger a valores más altos para intervalos más largos, a diferencia de *Apdex*, lo que permite incrementar la energía ahorrada al utilizar tiempos de espera más cortos: al obtener valores de satisfacción más altos para intervalos largos, éstos dejan de tener tanto peso en el cálculo del valor de tiempo de espera, permitiendo al algoritmo de cálculo de políticas centrar su búsqueda en el entorno $[T, W_f]$.

El valor de los parámetros por defecto se ha establecido a $W_s = T + 60$ y $W_f = T + 1800$ en base a los resultados de la encuesta para los parques estudiados.

El valor de W_f viene determinado por la persistencia en las respuestas de los usuarios sobre la tolerancia hacia la media hora (67% lo encuentra satisfactorio) y por los valores establecidos del tiempo de espera por defecto en el parque estudiado.

Se han retirado los intervalos que duran menos de 60 segundos para eliminar un comportamiento ruidoso en relación a intervalos de actividad muy cortos (la frecuencia de muestreo de los agentes que recolectaron los registros de utilización para este parque estaba configurada en 60 segundos también). Asimismo se ha configurado el parámetro W_s en 60 segundos por este motivo.

Cualquier administrador de un parque puede establecerlos a los valores adecuados para su parque con un procedimiento similar.

Respuesta a la pregunta A.3: ¿Cómo modelar la satisfacción de los usuarios?

El modelado de la satisfacción es clave para el cálculo de políticas energéticas que tengan un buen balance entre ahorro e impacto en los hábitos de los usuarios. Esta tesis propone modelar la satisfacción como la fracción de intervalos de inactividad que fuerzan al usuario a encender su computador al volver a usarlo.

Adicionalmente, esta tesis añade el concepto de satisfacción ponderada, que permite adaptar las políticas de forma más satisfactoria para los casos de intervalos de larga duración.

La satisfacción se pondera en base a preferencias de usuario, lo que permite ajustar los valores de satisfacción para representar la realidad de forma fidedigna, pero a la vez dota de mayor flexibilidad a la generación de políticas, especialmente al permitir manejar los intervalos de muy larga duración como si fueran satisfactorios.

3.2.2. Inactividad eliminada

Además de la satisfacción de usuario, otra medida importante es la cantidad de energía ahorrada. En este trabajo, se usa la métrica de inactividad eliminada, R (*removed inactivity*), que es la proporción de inactividad total eliminada (sustituida por tiempo de apagado) tras la aplicación de las políticas de gestión de energía. Esta métrica permite cuantificar el ahorro sin necesidad de medir el consumo en los computadores, lo que requiere instrumentación (R se puede convertir en energía si se conoce la potencia que consumen los computadores) y no depende del tamaño del parque; por ejemplo, para un modelo de computador que consume 100 W al estar siendo utilizado; se utiliza durante unas 8 horas diarias, durante otras 8 horas está inactivo y otras 8 horas apagado, una eliminación de la mitad de la inactividad ($R = 50\%$) supone un ahorro de 400 Wh/día para un único computador. En un parque de 1.000 computadores y un 90% de ahorro significaría alcanzar 720 kWh/día de ahorro.

El valor de R se puede calcular como el ratio del tiempo que los computadores están apagados del total de tiempo de inactividad. Para cada intervalo de inactividad más largo que el tiempo de espera, el computador está inactivo tanto tiempo como el tiempo de espera, T . Una vez que el temporizador expira, el computador se apaga, por lo tanto la inactividad se reduce en un valor igual a $I_i - T$. La ecuación 3.8 expone la definición formal de R .

$$R = \frac{\sum_{i=1}^n (I_i - T) \text{ if } I_i > T}{\sum_{i=1}^n I_i} \quad (3.8)$$

Una propiedad interesante de R es que su valor está limitado por la inactividad total y el valor de T .

3.2.3. Cálculo del tiempo de espera

El cálculo del valor del tiempo de espera depende de la distribución de la inactividad generada por los usuarios, recogida en el conjunto de registros de utilización del parque; pero también del objetivo de satisfacción determinado por el administrador para el parque, S_O .

Por tanto, establecer las políticas de gestión de energía, es necesario ser capaz de calcular el valor del tiempo de espera *a priori*, es decir, antes de conocer cuáles van a ser los intervalos de inactividad que van a ocurrir en el parque, para ello se utiliza el histórico de los registros de utilización como aproximación al comportamiento futuro de los usuarios en el corto plazo.

Durante unas cuantas semanas antes de poner en marcha el sistema de gestión de energía se recolectan registros de utilización del parque a gestionar, teniendo cualquier sistema de gestión energética desactivado, pero permitiendo a los usuarios utilizar y apagar sus computadores con libertad. Para esta tesis se han recolectado datos durante 4 semanas por parque a estudiar.

El valor del tiempo de espera puede obtenerse como la inversa de las ecuaciones de satisfacción S y S_w (ecuaciones 3.2 y 3.7), es decir, para un conjunto determinado de intervalos de inactividad, obtener el valor del tiempo de espera, T , que permite alcanzar un valor en función de la satisfacción.

Debido a que el objetivo de satisfacción es conocido, ya que es un parámetro de configuración del sistema, se puede obtener el valor exacto para el tiempo de espera evaluando la función recién obtenida, inversa de la de satisfacción, para el valor objetivo de satisfacción, S_O .

Basándose en los históricos de uso, se puede utilizar una función de satisfacción sencilla, tal y como se estima S en la ecuación 3.2. Para calcular el valor de T en este caso, por tanto, se requiere utilizar la función inversa a S y evaluarla para obtener el valor tiempo de espera y aplicarlo al sistema de gestión energética, de forma que el computador se apague si la inactividad supera ese valor.

Para una función de satisfacción sencilla, T se puede estimar como el valor que acumula el S_O % de la masa del histograma de tiempos de inactividad, siendo S_O el objetivo de satisfacción.

Para funciones más complejas, como S_w (ecuación 3.7), es posible que no se pueda hallar la inversa de forma analítica (como es el caso), por lo que habrá que recurrir a métodos numéricos o aproximaciones para evaluar el valor de T cuando se utilizan estas funciones.

En este trabajo se ha implementado la búsqueda de la función inversa de S_w mediante métodos numéricos, ya que no cuenta con una inversa que se pueda obtener de forma analítica. De esta forma, se evalúa de forma directa para obtener el valor del tiempo de espera. Hay que notar cómo S_w depende de los intervalos de inactividad de cada fracción horaria y por lo tanto debe ser calculada para cada hora y computador.

En concreto, se ha utilizado para la evaluación de la función inversa el Método de Brent [Brent 1973] para encontrar raíces (se busca la raíz para la función $T_w = S_w^{-1} - S_O$, lo que resulta equivalente a evaluar la función inversa para S_O). Este método tiene un coste computacional que escala de forma lineal con el número de datos que se tienen para procesar, en este caso el número de intervalos de inactividad siendo procesados.

Existen otras soluciones como los métodos genéticos o el descenso de gradiente, pero son más complejos computacionalmente.

Respuesta a la pregunta B.1: *¿Cómo generar políticas energéticas en base a la satisfacción y el comportamiento de los usuarios?*

Tal y como se observa durante todo este capítulo, la generación de políticas tiene como objetivo la selección un valor para el tiempo de espera del gestor energético que contiene el sistema operativo de los computadores del parque. Este valor no puede sino depender del comportamiento del usuario y de cuánta satisfacción se desea garantizar.

El comportamiento del usuario se modela mediante distribuciones de los

tiempos esperados de actividad e inactividad, que van variando a lo largo del día. Estas distribuciones, en particular las de inactividad, nos permiten seleccionar un valor del tiempo de espera utilizando un estimador de satisfacción, que tiene como parámetro una distribución de inactividad (que variará en función del momento del día y del usuario). En el caso de esta tesis se propone utilizar el concepto de satisfacción ponderada (S_w), que ofrece un buen balance entre ahorro y satisfacción, especialmente en parques con intervalos de inactividad muy largos.

Capítulo 4.

Herramienta de optimización

Además del método presentado en el capítulo anterior, se ha desarrollado una herramienta que permite a los administradores de parques generar las políticas de gestión energética de forma analítica y simular diversas configuraciones y variantes del parque en estudio.

El uso de un mecanismo de simulación es particularmente interesante cuando se está experimentando con diversos parámetros o se plantean modificaciones al parque, por ejemplo en previsión del crecimiento futuro de la misma. Sin un simulador, estos experimentos deberían llevarse a cabo en el parque real, con el consiguiente coste para los usuarios de la misma.

Para la generación de políticas energéticas de forma analítica, la herramienta hace uso de los registros de utilización del parque, que contienen información relevante sobre actividad, inactividad y órdenes de apagado de los usuarios. Estos registros se recolectan mediante un sistema de agentes instalados en cada computador, que envían eventos a una base de datos central, que agrega y anonimiza los eventos.

Recolectar eventos no es siempre posible, por ejemplo, cuando se está planificando un nuevo parque. Para este caso de uso, la herramienta permite generar eventos mediante un mecanismo de simulación de eventos discretos, estableciendo un conjunto de parámetros de alto nivel, tales como el número de usuarios o la información de horarios. Esta herramienta ha sido implementada usando Python y SimPy con una simulación implementada en base al método de simulación por agentes.

4.1. Casos de uso

La gestión energética puede implementarse como un ciclo de optimización, que empieza con la recolección de registros de utilización o la generación de los mismos, en el caso de parques sintéticos. Con estos registros, se pueden obtener las políticas de gestión energética de forma analítica, estudiar opciones de agrupación de usuarios o experimentar con variaciones y ajustes sobre el parque usando el componente de simulación.

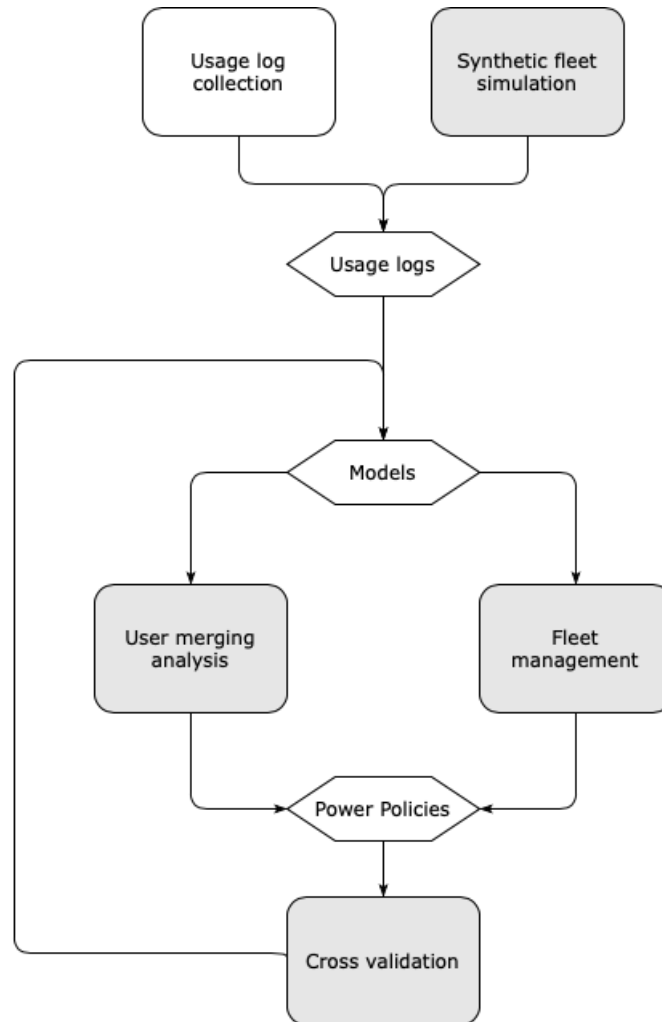


Figura 4.1.: Diagrama del ciclo de optimización para gestión energética.

La herramienta implementa cuatro casos de uso, representados en gris en la figura 4.1 (la recolección de registros de utilización se realiza con una herramienta aparte):

Simulación de parques sintéticos. Con unos pocos parámetros de alto nivel, es posible simular un parque para el que no se cuenta con registros de utilización, generando los eventos correspondientes en base a distribuciones estadísticas proporcionadas por el administrador.

Análisis de grupos de usuarios. La herramienta permite comparar el impacto en el ahorro energético al generar políticas energéticas personalizadas para cada

usuario, o bien para grupos de diversa clase de usuarios. También se permite realizar agrupaciones temporales, esto es, usando datos que corresponden a otras horas para completar o ajustar momentos del día con menos información.

Generación de políticas energéticas. Las políticas energéticas se generan de forma analítica en base a los modelos ajustados con los registros de utilización del parque o producto de la simulación de un parque sintético.

Validación cruzada. La herramienta permite utilizar diversos registros de utilización para validar las políticas generadas, mediante simulación o de forma analítica, y cómo se adaptan a diferentes escenarios proporcionados por el administrador.

Se trata de una herramienta que proporciona múltiples opciones al administrador de la parque para planificar, gestionar y hacer crecer dicho parque. Las soluciones basadas en mecanismos de simulación son especialmente interesantes para casos de parques grandes, donde probar cambios puede ser especialmente costoso.

Se proporciona una guía de uso de esta herramienta en el apéndice [B](#).

4.2. Aplicación de las políticas energéticas

La aplicación de las políticas energéticas se realiza por medio del gestor energético del sistema operativo: se le proporciona las políticas en forma de valores de tiempo de espera para distintos niveles de apagado, y éste ajusta el nivel de apagado pasado el tiempo pertinente.

El estándar ACPI [ACPI Specification, Version 6.3](#) especifica diferentes niveles de apagado, desde S1 a S5. Mientras que S5 representa un estado de apagado casi total, donde el usuario debe recuperar su sesión al regresar al computador, niveles intermedios como S3 (suspensión) ó S4 (hibernación) pueden ser utilizados para incrementar la satisfacción ya que proporcionan una recuperación más rápida al salvaguardar el estado del computador.

Respuesta a la pregunta B.3: ¿Cómo se implementan las políticas en el parque?

El gestor de parques genera de forma analítica una política para cada usuario y hora del día. Esta política consiste en un valor del tiempo de espera, de forma que superado ese tiempo de inactividad se apaga el computador hasta la vuelta del usuario. Un agente recibe desde el nodo de análisis políticas actualizadas cada hora del día.

Las políticas energéticas se aplican en el parque mediante los gestores de energía del sistema operativo de cada computador. El agente actualiza de forma automática la configuración del sistema operativo y es éste el que cambia el estado ACPI correspondiente.

4.3. Recolección de registros de utilización

Para generar las políticas energéticas es necesario contar con unos modelos de utilización que resuman métricas como la actividad, inactividad y el apagado que los usuarios hacen con su computador. Estos modelos se ajustan sobre registros de utilización que se recolectan mediante agentes instalados en cada computador del parque.

Se utiliza una arquitectura sencilla cliente/servidor, donde un pequeño agente instalando en cada computador observa ciertos eventos que permiten inferir los cambios de actividad a inactividad y vice versa y un servidor que permite recolectar y tratar los datos.

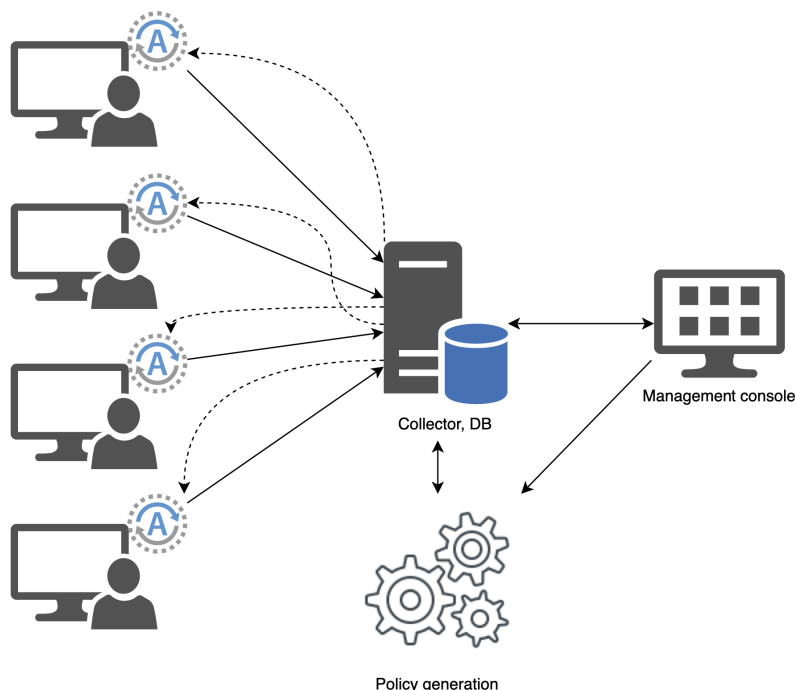


Figura 4.2.: Arquitectura de recolección de registros de utilización.

La figura 4.2 representa la arquitectura del sistema. Cada computador cuenta con un agente que envía información de forma periódica a un colector central, que anonimiza los registros y los almacena en una base de datos central. Estos registros se utilizan como entrada para la generación de políticas energéticas descrita en esta tesis.

El servidor central cuenta con una serie de servicios añadidos, además de la recolección de los datos. En primer lugar, anonimiza los registros, utilizando un identificador no personal para cada computador; en segundo lugar, realiza

una primera pasada a los datos al recibirlos para eliminar posibles duplicados, reconstruir eventos perdidos cuando es posible y compactar registros de eventos muy frecuentes, como intervalos de actividad muy cortos; en tercer lugar ofrece una interfaz web sencilla donde se pueden ver gráficos con medidas resumen y comparativas entre usuarios; finalmente implementa el generador de políticas, que permite extraer de forma analítica las políticas energéticas óptimas para el parque en un momento dado.

Los agentes utilizan de forma periódica el API `GetLastuserinput()`^[1] en equipos con Windows (en Linux se utiliza `XScreenSaverQueryInfo()`^[2] y en macOS las funciones de gestión del salvapantallas de `IOKit`^[3]), que indica el momento de la última interacción del usuario con el computador, pudiendo así inferir cuál es el estado actual (actividad o inactividad) y su duración. Estos agentes determinarán el estado del computador cada 60 s y enviarán una actualización al colector, que elimina duplicados y calcula la duración de los eventos para almacenarlos en la base de datos.

El agente es un proceso muy ligero, que consume una cantidad despreciable de CPU y entre 5 y 10 MiB de memoria: fundamentalmente estará dormido, excepto un ciclo corto cada 60 s para hacer una llamada al sistema y a un API remoto. Para proteger al colector, el timer inicial del agente se aleatoriza, de forma que las peticiones llegan con una distribución uniforme.

Cabe destacar el efecto de cuantización que ocurre en los registros: al enviar actualizaciones cada 60 s al colector, este valor es la duración mínima de cada intervalo. En algunos parques podría ser interesante incrementar la frecuencia de recolección a costa de una generación de registros más voluminosa.

Los registros se estructuran como una lista de intervalos, siendo cada intervalo una tupla de la forma (*computador, evento, timestamp, duracion*) y se almacenan en una base de datos relacional. Sobre esta base de datos se pueden consultar conjuntos de registros para obtener gráficos y comparativas o generar políticas de forma analítica.

El apéndice C muestra más detalles sobre el formato de las trazas y las herramientas de procesamiento proporcionadas.

4.4. Implementación de los modelos

4.4.1. Apagados solicitados por el usuario

Cuando un usuario decide apagar su computador, se graba un evento en el registro de utilización por los agentes, de la misma forma que ocurre con los intervalos de inactividad. La única diferencia para modelar el apagado dirigido por el usuario es la necesidad de contar no solo con la duración de los intervalos de

¹<https://msdn.microsoft.com/en-us/library/windows/desktop/ms646302.aspx>

²<https://linux.die.net/man/3/xscreensaverqueryinfo>

³<https://developer.apple.com/documentation/iokit>

apagado, sino también de la probabilidad de que ocurra un apagado en una hora determinada.

Mientras que los intervalos de actividad e inactividad se modelan como una sucesión ininterrumpida de ellos, los intervalos de apagado pueden no ocurrir. La probabilidad de que ocurra un apagado dirigido por el usuario en una hora determinada se modela estimando una función de probabilidad en base al número de apagados que han sucedido en dicha hora en los registros de utilización.

Existen diversas opciones para estimar esta función de probabilidad, por ejemplo, se podría ajustar una función binomial para indicar si en un determinado momento procede un apagado dirigido por un usuario, intentando generar un número similar de apagados que los registrados. Cuando se realizan múltiples simulaciones para obtener resultados estadísticamente significativos, poder estimar esta distribución binomial requiere conocer el número de replica de simulación, lo que no es siempre posible, como en el caso de la implementación proporcionada, que realizará un número variable de repeticiones de simulación hasta tener significación.

La opción escogida para este trabajo es simular el mismo número de eventos de apagado que los registrados y aleatorizar su ocurrencia durante la hora de forma uniforme. En el caso de estar generando un parque sintético, el operador introduce los parámetros de una distribución estadística para generar de forma aleatoria tanto la duración como la cantidad de eventos de apagado solicitados por el usuario.

4.4.2. Consolidación de modelos

Aunque generar un modelo por cada usuario proporciona en principio el mejor ajuste y adaptación a las necesidades de cada usuario (y por tanto de satisfacción), existen casos donde puede producirse sobre-ajuste: por ejemplo en casos de estudio de parques donde se cuentan con registros de utilización compuestos por muy pocos datos al producirse pocos eventos, como durante las noches.

Este efecto de sobre-ajuste puede mitigarse usando datos de múltiples usuarios o de múltiples horas para ajustar un modelo consolidado, el cual al contar con una mayor cantidad de datos, produce unos ajustes con menor ruido.

Por omisión, el modo de operación tanto del análisis como de la simulación de parques es sin consolidar, es decir, generando los modelos más personalizados posibles para cada usuario. En este caso, se produce un modelo por cada usuario y hora de la semana (esto es, 168 modelos para cada usuario). Cada uno de estos modelos contiene una distribución para cada una de las cuatro variables principales a modelar, ajustada desde los datos recogidos en los registros de utilización.

La consolidación de modelos puede realizarse de dos formas: en la dimensión temporal o por usuario. En el primer caso, todos los datos obtenidos para cada uno de los usuarios se consolidan en un único modelo para cada uno de ellos, de forma que cada usuario cuenta con un único modelo, pero individualizado. En el caso de la consolidación por usuario, se realiza una consolidación para cada una de las horas, obteniendo un modelo para cada hora, compartido por todos los usuarios, en este caso, cada hora presenta un comportamiento diferente, pero no hay personalización para cada usuario. Finalmente, al consolidar por ambas

dimensiones al mismo tiempo, se obtiene un único modelo, que se usa de forma singular para todos los usuarios y horas de la semana.

Al consolidar modelos se reduce el número de los mismos, teniendo tantos como usuarios en el caso de consolidar por tiempo, exactamente 168 al consolidar por usuario y sólo 1 al hacerlo por ambas dimensiones. Sin embargo, la cantidad de datos que se utilizan para ajustar las distribuciones que describen las variables de los modelos es mayor, y el tiempo para extraer números aleatorios en el caso de usarlas para simulación puede ser sustancialmente mayor (Toraichi et al. 1987). Por contra, al contar con distribuciones ajustadas con muestras más grandes, la posibilidad de tener variables ruidosas disminuye.

Es necesario indicar que la consolidación de modelos aquí propuesta no es un mecanismo de *clustering* al uso. En este caso, se consolidan los modelos de todos los usuarios presentes en la simulación o ninguno. Aplicando un método de búsqueda de clústeres permitiría determinar grupos de usuarios con similar comportamiento donde consolidar los modelos. No se ha implementado búsqueda de clústeres en este trabajo dada la pequeña diferencia en inactividad eliminada en los parques estudiados usando consolidación o no, como se verá en la sección 5.4.1, lo que indica un margen de mejora discreto con un coste importante de complejidad.

4.4.3. Extrapolación de modelos

Otra transformación que se puede hacer a los modelos es extrapolar parques con diferente configuración basadas en un parque conocido, por ejemplo, con una cantidad de usuarios diferente o simular durante períodos de diferente duración.

Cuando se simula un parque más grande que el parque usado para el estudio y obtención de los registros de utilización, se tienen que extrapolar nuevos usuarios de los datos existentes. Un método sencillo puede ser utilizar un modelo consolidado y simplemente replicar tantos usuarios como sea necesario en base a él. Todos los usuarios compartirán el mismo comportamiento en este caso. Otra opción es replicar usuarios individuales, de forma aleatoria.

Cuando se simula un parque más pequeño que la utilizada para el estudio, al necesitar reducir el número de usuarios a simular es necesario retirar usuarios de los que se disponían en los registros de utilización. Por ejemplo, bastaría con realizar un muestreo aleatorio de los usuarios originales. Al retirar usuarios, se elimina parte del comportamiento global del parque, no obstante. Para evitarlo, puede usarse un método similar al anterior: usar un modelo consolidado y replicar tantos usuarios como sea necesario, esta vez menos que con los que se contaba originalmente.

Cuando se desea adaptar la duración de la simulación no es necesario realizar cambios en los modelos: como todos los modelos se usan en base al ciclo de una semana, sólo es necesario simular tantas semanas como sea necesario.

<p>Respuesta a la pregunta B.4: <i>¿Cómo hacer experimentos con diversas variantes del parque?</i></p>
--

Para realizar experimentos con variantes del parque a estudiar, se necesita emplear un proceso de simulación, ya que el análisis solamente no es suficiente.

En esta tesis se propone utilizar un método de simulación por eventos discretos que permite utilizar los mismos modelos que se utilizan para el análisis y generación de políticas. Estos modelos se pueden generar mediante tanto registros de utilización como generando modelos sintéticos basados en una serie de parámetros definidos por el usuario.

Para la generación de variantes, se modifican los modelos conforme a las necesidades del administrador del parque: se pueden añadir más computadores en base a los conocidos, reducir el número, alterar los tiempos de simulación o utilizar otros modelos diferentes durante la simulación para comprobar el buen comportamiento de los modelos utilizados para la generación de políticas.

El simulador utiliza exactamente el mismo método de análisis para la generación de políticas, de forma que representa bien lo que ocurriría en la realidad. Este estudio de variantes es más ágil y barato que plantear cambios reales en el parque estudiado (en otras ocasiones incluso imposibles, por ejemplo cuando se está estudiando un parque de nueva creación).

4.5. Procesos de simulación y diseño del software

El software de simulación se ha diseñado siguiendo el método de procesos de simulación. Cada uno de los elementos a simular del parque (usuarios y computadores) es un proceso, dirigido por el modelo correspondiente. Los modelos, y otras herramientas de soporte tales como de recolección y cálculo de estadísticas, se encapsulan en objetos llamados distribuciones de actividad.

Una simulación basada en procesos es un patrón que permite estructurar un software de simulación de eventos discretos en base a los diferentes procesos estocásticos que constituyen el modelo compuesto. Cada uno de los procesos representa a uno de los agentes que toman parte dentro del modelo a simular (tales como los usuarios o el computador con su gestor de energía en este caso) y tendrá un comportamiento encapsulado en forma de una lógica de generación de eventos. Los eventos que genera cada proceso se intercambian con otros procesos, permitiendo así un comportamiento tan complejo como sea necesario y dependencias entre los diversos procesos.

La generación de eventos por parte de cada proceso representa el comportamiento de dicho agente como respuesta a un estímulo en forma de evento recibido y podrá ser determinista (una respuesta fija), estocástica (una respuesta aleatorizada) o ninguna (el final del procesamiento del evento).

Los procesos pueden modelarse como clases en un lenguaje orientado a objetos, de esta forma pueden replicarse muchos procesos con el mismo o similar comportamiento de forma sencilla, para por ejemplo crear todos los usuarios del parque.

Cada proceso de simulación se ha implementado como una clase en Python usando el framework SimPy⁴, lo que permite instanciar de forma sencilla simulaciones con cientos o miles de procesos. SimPy ofrece un API basado en generadores que permite implementar los procesos como un sencillo bucle infinito, donde la duración de cada evento se simula con un *timeout* (cualquier intervalo de tiempo en la simulación es un sencillo avance del reloj de simulación, lo que hace muy eficiente el proceso).

El diagrama UML de la figura 4.3 resume el diseño de los procesos de simulación y sus relaciones. Los procesos principales, `User` y `Computer`, se instancian tantas veces como usuarios tiene el parque a simular emparejados: cada usuario requiere de una instancia de cada clase, una para simular el comportamiento de la persona ante el computador y la otra para simular el sistema operativo y su correspondiente gestor de energía.

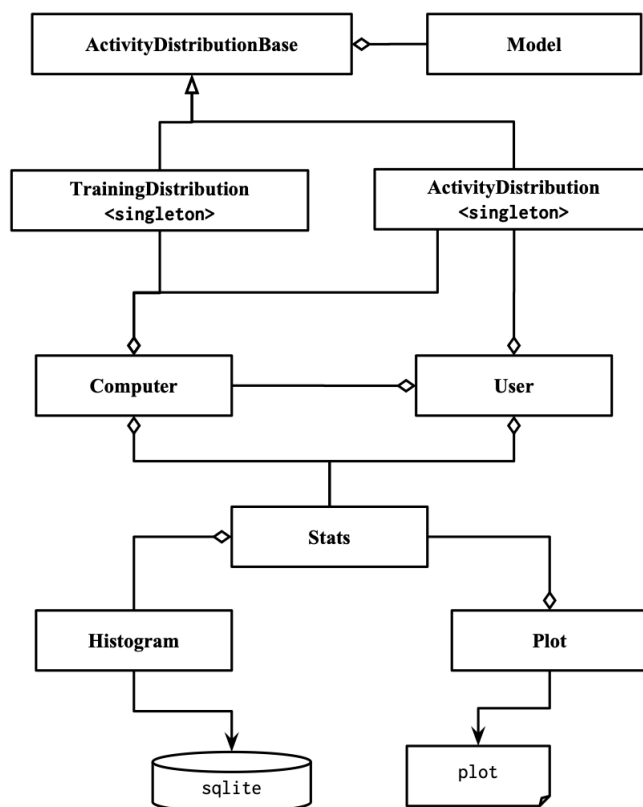


Figura 4.3.: Diagrama UML de los procesos de simulación principales.

⁴<http://simpy.readthedocs.org>

Cada computador cuenta con un identificador único, que puede ser su dirección MAC u otro elemento definido por el administrador. Todos los usuarios, modelados por la pareja de instancias `User` y `Computer`, se comportan siguiendo un modelo generalizado de colas ($G/G/1$), tal y como muestra la figura 4.4.

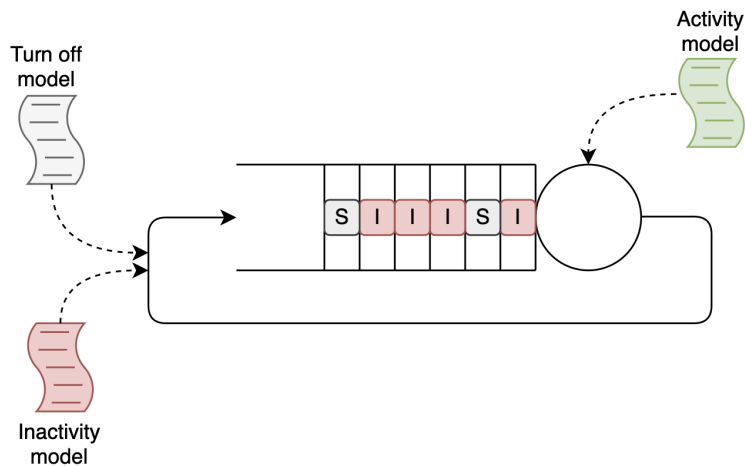


Figura 4.4.: Modelo de colas $G/G/1$ para los usuarios simulados.

Cada petición en ese modelo de colas representa una solicitud de realización de actividad del usuario, de forma que el tiempo entre llegadas representa la duración de la inactividad y el tiempo de procesamiento la duración de la actividad. Estos parámetros del modelo de colas se representan mediante distribuciones aprendidas desde los registros de utilización, con o sin consolidación.

Las peticiones que se realizan al modelo de colas pueden ser de dos tipos: de inactividad (I) o de apagado (S). Cada evento determina su duración en base a la distribución correspondiente. En el caso del apagado, el procesamiento incluye un cambio de estado del computador, lo que permite trazar cuándo el usuario ha tenido que encenderlo para procesar un evento de actividad.

Adicionalmente, cada uno de los computadores contiene un proceso que simula el gestor energético del sistema operativo y que, en base a las políticas energéticas determinadas, puede apagar computadores sin indicación del usuario. La generación de políticas energéticas se realiza de forma analítica para cada hora de la semana.

La clase `ActivityDistribution` (singleton) encapsula todo el comportamiento aprendido de los usuarios y computadores mediante los registros de actividad, resumiéndolos en distribuciones empíricas para cada variable. En tiempo de ejecución, estas distribuciones generarán eventos de forma aleatoria para los procesos de simulación.

Existe otro objeto con el mismo comportamiento que permite separar los modelos ajustados para determinar políticas energéticas y para realizar simulaciones de validación: `TrainingDistribution` se utiliza para encapsular los datos para la

generación de políticas y `ActivityDistribution` para validar. Estos objetos son completamente independientes de la manera de obtención y partición de los registros de utilización.

Finalmente, otra clase de utilidad es `Stats` (singleton): permite almacenar estadísticas de diversas variables desde cualquier proceso de simulación, organizado en variables. Cada variable cuenta con una instancia de `Histogram`, que almacena en una base de datos `sqlite3`⁵ una serie temporal de pares (*timestamp*, *valor*). Estas estadísticas se usan para validar la salud de la simulación, además de obtener los resultados para analizar y resumir en gráficos.

La salud de la simulación es una comprobación que se realiza tanto durante el procesamiento, como al finalizarlo. Es necesario que todos los eventos tengan un total de tiempo procesado igual al producto del número de computadores y el tiempo simulado en segundos, de esta forma se asegura que no ha habido errores en el procesamiento de eventos. Asimismo, se pueden establecer otras invariantes, como que el tiempo total que los computadores están apagados por el gestor de energía siempre ha de ser menor al total de tiempo de inactividad registrado.

Utilizar un backend de almacenamiento en disco como `sqlite3` permite tener persistencia permanente y, además, reducir el consumo de memoria en simulaciones muy largas, donde pueden generarse millones de eventos.

4.5.1. Optimización del ajuste de distribuciones

Existen dos secciones de código donde el simulador pasa la mayoría del tiempo de ejecución: ajustando las distribuciones y extrayendo números aleatorios a partir de las mismas. Optimizar el código que implementa la distribución empírica permite acelerar la simulación notablemente.

En [Law 2015, p. 462], Law ofrece una implementación que permite extraer aleatorios en tiempo $O(1)$. Además, cuenta con una optimización para precalcular las diferencias de los valores de la distribución acumulada (CDF) para reducir las operaciones para cada extracción.

Mientras que la implementación de Law es rápida, utiliza una gran cantidad de memoria al almacenar todos y cada uno de los valores de la muestra usada para el ajuste. Para solucionar este problema se ha implementado una versión basada en una curva *spline* [Ivezić et al. 2014], que reduce el consumo de memoria, manteniendo una extracción de aleatorios rápida. Tal y como muestra la figura 4.5, tanto el ajuste como la extracción tienen una complejidad lineal con el tamaño de la muestra.

⁵<https://www.sqlite.org>

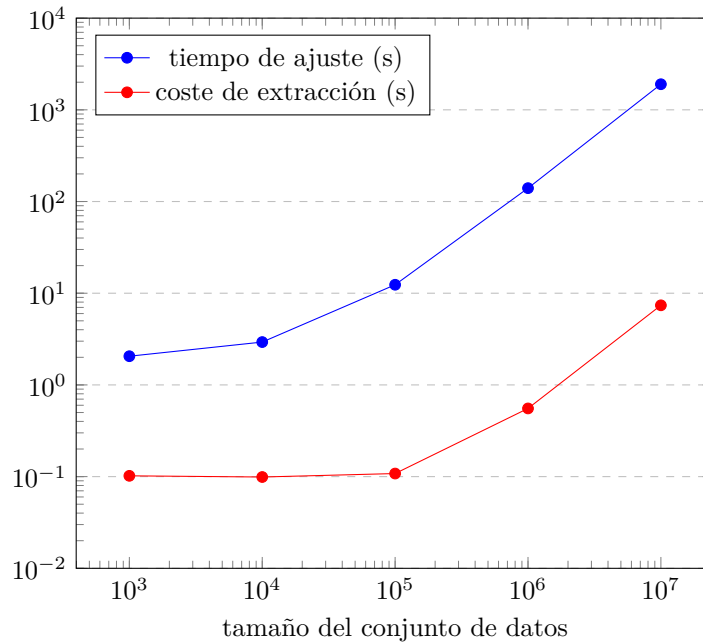


Figura 4.5.: Benchmark de la distribución empírica basada en *spline*.

Adicionalmente, la figura [4.6](#) muestra un perfil de uso de memoria durante una ejecución de una serie de simulaciones de un parque con 265 usuarios. Esta serie requirió de tres simulaciones para que los resultados convergieran dentro de un intervalo de confianza de $\pm 0,5$ puntos porcentuales con un nivel de confianza del 95%. El proceso tiene tres fases: el análisis, casi inmediato, donde se procesan los registros y se ajustan las distribuciones de los modelos; la simulación, que requerirá un número de iteraciones hasta que los resultados sean estables; y por último el resumen de los datos y la generación de gráficos y tablas finales. Esta última fase es la que más tiempo consume al tener que iterar sobre todos los datos y eventos recogidos en la base de datos de estadísticas.

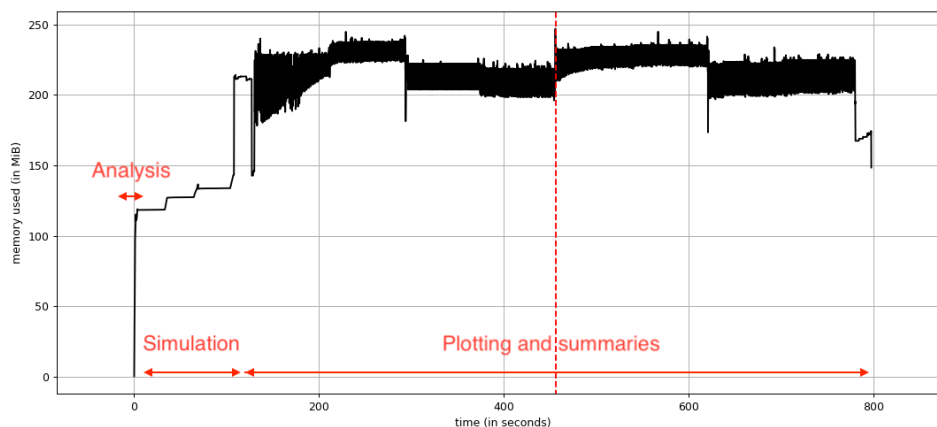


Figura 4.6.: Uso de memoria del simulador.

Como se anotaba anteriormente, el componente crítico para optimizar la simulación es la distribución empírica. La tabla [4.1](#) compara el impacto en tiempo de ejecución y uso de memoria de dos implementaciones diferentes, consolidando modelos (*merged*) o no (*unmerged*), con un parque de 265 usuarios.

La versión *Law* es la implementación básica, que con un conjunto de datos grande no es capaz de terminar el ajuste tras 20 minutos. La versión con *spline* mantiene el uso de memoria bajo y muestra una velocidad aceptable.

Tabla 4.1.: Comparación del rendimiento del simulador con diversas implementaciones de la distribución empírica

Variante	Tiempo	Memoria
<i>Law</i> , no consolidado	2m44	281 MiB
<i>Law</i> , consolidado	>20m	—
<i>spline</i> , no consolidado	2m10	287 MiB
<i>spline</i> , consolidado	3m37	237 MiB

Todos los modelos generados se distribuyen en una estructura de árbol multigrado implementado con diccionarios para tener un acceso rápido a los mismos ($O(1)$), existiendo una instancia de dicho árbol por cada variable modelada. Cuando se realiza una consolidación, se eliminan los duplicados de los modelos de este árbol, manteniendo una única copia de las muestras de datos de los registros de utilización en memoria.

4.5.2. Escalabilidad del simulador

Tanto el simulador como el análisis deben escalar para soportar grandes parques, tales como las de corporaciones con desde miles a decenas de miles de usuarios. El tiempo necesario para simular un parque depende tanto del número de usuarios a simular como de la duración de la simulación en semanas, mientras que el análisis para generar los modelos desde los registros de utilización también escala de forma lineal, pero con el tamaño de dichos registros.

La figura 4.7 resume un conjunto de pruebas donde se han realizado simulaciones con los mismos datos de entrada, pero un número creciente de usuarios y demuestra cómo el tiempo para completar la simulación escala de forma lineal con el tamaño del parque [Medrano Llamas et al. 2015], mientras que la figura 4.8 replica el experimento, pero incrementando el número de semanas a simular.

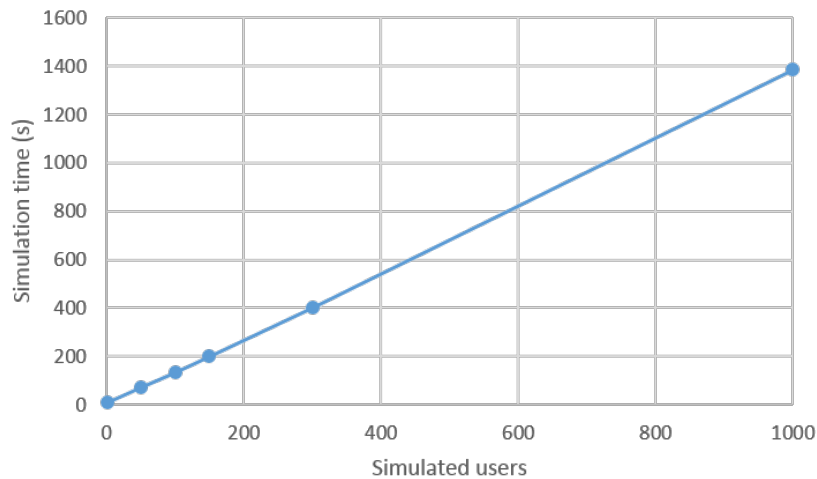


Figura 4.7.: Evolución del tiempo de simulación en función del número de usuarios.

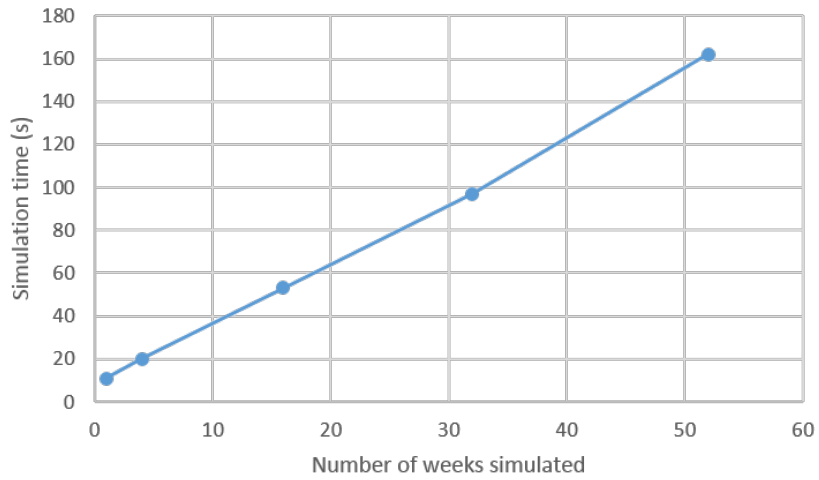


Figura 4.8.: Evolución del tiempo de simulación en función de la duración de la simulación.

Es conveniente recordar que el tiempo necesario para completar simulaciones depende de ambas variables a la vez, de forma que simulaciones donde se incrementa mucho el número de usuarios y durante muchas semanas pueden escalar de forma supralineal, tanto en tiempo de ejecución como en uso de memoria.

Capítulo 5.

Validación

En este capítulo se presenta cómo se ha validado de forma empírica tanto el método de generación de políticas energéticas como la herramienta que permite el análisis y simulación de parques. Asimismo, se ha estudiado el impacto que en la efectividad de las políticas puede tener la consolidación de modelos de usuario.

La validación se ha realizado de forma incremental, estudiando primero los registros de utilización, acto seguido la bondad de ajuste de la distribución empírica y finalmente la generación de políticas de gestión energética. El capítulo termina con una valoración de la precisión de la funcionalidad de generación de parques sintéticos.

5.1. Partición de los registros de utilización

Los registros de utilización recogidos de forma empírica de un parque real deben ser particionados para poder realizar una mejor validación de las políticas energéticas generadas por la herramienta, dividiendo en dos mitades los datos registrados, una mitad para generar las políticas y la otra para validar su efectividad antes de aplicarlas a un parque real.

Existen muchos métodos de partición de datos, en este trabajo se han partido los registros en dos mitades, basándose en la duración temporal del registro. De esta forma, y tal y como detalla la tabla 5.1 los dos registros de unas 8 semanas de duración obtenidos en dos parques de una empresa local, se han partido en sendas mitades de cerca de un mes de duración.

Registro	Fechas	Usuarios	Duración	Notas
1.1	5/10 - 1/11, 2014	265	3,9 sem.	Sin portátiles
1.2	2/11 - 30/11, 2014	265	4 sem.	
2.1	1/4 - 30/4, 2015	64	4,1 sem.	Con portátiles, min. 50 eventos
2.2	1/5 - 29/5, 2015	75	4 sem.	
2.3	1/4 - 17/9, 2015	65	20,1 sem.	

Tabla 5.1.: Partición de los registros disponibles.

Se ha añadido una partición de registro adicional (2.3) que es más extensa temporalmente, llegando hasta las 20 semanas. Este registro se utiliza para pruebas

en simulaciones largas y en tests unitarios y de integración.

La tabla 5.2 resume los tamaños de los registros y sus particiones, contando el registro 1 con muchos más eventos al tratarse de un parque más grande. Los registros se almacenan en formato JSON comprimido tras ser extraídos de la base de datos del agente central con los filtros temporales que se desee. Es destacable la diferencia entre el número de eventos de apagado registrados en la partición 1.1 con respecto a la partición 1.2, lo que es debido a la presencia de dos jornadas de fiesta en el primero, efecto similar ocurre en la partición 2.1, aunque es menos intenso al sólo contar con medio día de fiesta.

Registro	Tamaño	Ev. inactividad	Apagados	Notas
1.1	16 MB	242.484	7.729	Fiesta 12/10 y 1/11.
1.2	13 MB	125.517	4.290	
2.1	3 MB	27.300	1.409	Fiesta 2/4.
2.2	3,5 MB	31.418	1.574	
2.3	5,8 MB	113.844	4.913	

Tabla 5.2.: Tamaños de los registros.

Los dos registros tienen propiedades diferentes, mientras que el segundo incluye dispositivos portátiles, el primero no lo hace, ya que se suelen mantener apagados de forma más consistente y su comportamiento depende de la fuente de energía que utilicen (batería o corriente alterna), de esta forma se puede comparar el impacto de ambas clases de dispositivos.

El segundo registro excluye computadores que no presentaban 50 o más eventos durante los dos meses aproximadamente que duró la recogida de datos. Esta exclusión permite comparar los resultados de un registro con datos más ruidosos (computadores que presentaban muy pocos eventos) con un registro con datos más consistentes de cara a validar la consolidación de modelos y su impacto en las políticas energéticas generadas.

5.2. Validación de la implementación de la distribución empírica

Para validar la implementación del software que genera la distribución empírica, basada en *splines*, es necesario asegurar que el software ajusta correctamente los parámetros de la distribución y es capaz de generar aleatorios en base a dicha distribución sin sesgos o errores. Es decir, es necesario dado un conjunto de datos, ajustar una distribución, generar otro conjunto de datos aleatorios con la distribución ajustada y validar la calidad de ajuste comparando propiedades del conjunto de datos original y el generado aleatoriamente.

Esta validación se ha realizado mediante tests de Kolmogorov-Smirnov ($K-S$) (Massey Jr. 1951) que se ejecutan de forma automática con cada cambio al

código fuente, utilizando el sistema de integración continua GitHub Actions¹. Se realizan dos pruebas: una con registros de utilización y otra con datos generados de forma aleatoria, de forma que se pueden maximizar las características de los conjuntos de pruebas. Asimismo, se prueban otras funciones añadidas, como el agrupamiento de conjuntos de datos en sus distintas variantes.

Los tests aleatorios generan una muestra de 50.000 números aleatorios procedentes de diversas distribuciones conocidas (dos variantes de la normal, exponencial y Pareto). A continuación ajustan una distribución empírica sobre dichas muestras y extraen otros 50.000 aleatorios de la distribución ajustada. Es con las dos muestras cuando se ejecuta el test $K-S$ con un nivel de confianza del 99,9% para comprobar que ambas muestras provienen de la misma distribución.

Estas comprobaciones permiten asegurar que el código que implementa la distribución empírica es correcto, ya que la precisión y corrección del análisis y simulación depende de esto. En todos los casos, la significación de las pruebas viene determinada por el valor del estadístico $D_{n,m}$ y el $p-value$, rechazando la hipótesis si las distribuciones fueran diferentes (el $p-value$ en ese caso sería menor a la significación de la prueba, 0,01 %).

5.3. Comparación entre la satisfacción y satisfacción ponderada

Se ha comparado la bondad de los estimadores S y S_w , utilizando el parámetro $W_f = 1800$ y los registros 1.1 y 2.1 (cuyas particiones están descritas en la tabla 5.1), para comparar la satisfacción de usuario en función del tiempo de espera, T , y de la inactividad eliminada, R . En la figura 5.1 pueden observarse los tres valores (S , S_w y R) para diferentes valores de T . Como es esperable, a mayores tiempos de espera, la satisfacción se incrementa, mientras que menores valores de T permiten incrementar el ahorro de energía, eliminando más tiempo de inactividad del parque, con valores de R cercanos a 95 %.

¹<https://github.com/asi-uniovi/power-simulation/actions>

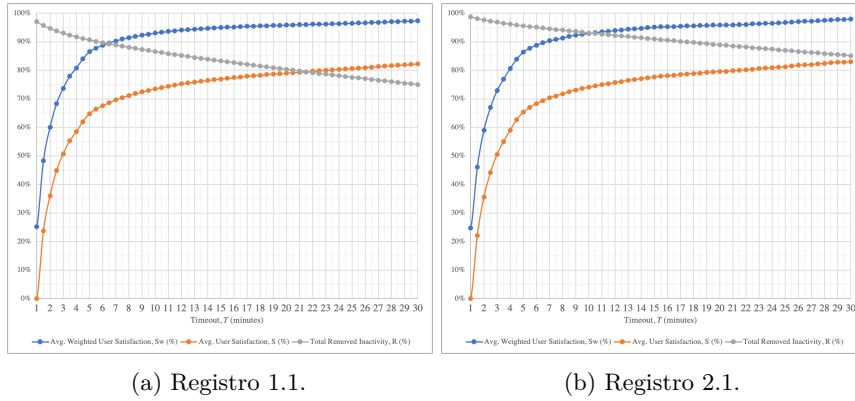


Figura 5.1.: S , S_w y R en función de T .

El coste de incrementar la satisfacción es reducir el ahorro energético mediante tiempos de espera más largos. Sin embargo, esta relación no es lineal: en ambos casos, a partir de tiempos de espera de unos 10 minutos, se entra en un régimen de retornos decrecientes, en los que ligeros incrementos de satisfacción conllevan reducir notablemente el ahorro energético. P. ej., en la figura 5.1a, al pasar de una satisfacción ponderada, S_w , del 95 % al 99 %, reduce la inactividad eliminada, R , del 85 % al 69 %.

En la figura 5.1 también se observa una clara diferencia entre los comportamientos de los estimadores de la satisfacción S y S_w . S penaliza de forma severa cualquier intervalo de inactividad de más duración que el tiempo de espera (T), ya que dure lo que dure va a ser insatisfactorio. Este problema se mitiga en S_w , ya que intervalos de mucha duración se considerarán satisfactorios.

Es interesante la relación entre la satisfacción, S , y la ponderada, S_w ; en ambos casos existe una diferencia de entre 20 y 25 puntos porcentuales, estrechándose conforme se alarga el valor de tiempo de espera (para un valor suficientemente alto, o infinito, la satisfacción siempre alcanza el 100 %).

Respuesta a la pregunta B.2: *¿Cómo se relaciona la satisfacción con el ahorro de energía?*

La relación entre el nivel de satisfacción objetivo y el ahorro energético es de proporcionalidad inversa, pero no lineal.

Hay que hacer notar que la relación entre ambas variables depende de varios factores. En primer lugar, de las variables del entorno tales como el parque y el perfil de uso de los computadores por parte de sus usuarios. En segundo lugar, depende de la función que se utilice para estimar la satisfacción, por ejemplo si se usa la función de satisfacción ponderada presentada en esta tesis, se mejora notablemente el ahorro para el mismo nivel de satisfacción. Finalmente, el nivel de satisfacción objetivo escogido para el parque determinará no solo el

potencial de ahorro total, sino también la elasticidad del ahorro energético.

En regímenes de satisfacción bajos (p. ej. 50 %-70 %), incrementos altos de satisfacción conllevan bajadas moderadas de ahorro energético (elástico), mientras que en regímenes de alta satisfacción (p. ej. >90 %), es necesario reducir mucho el consumo para seguir incrementando la satisfacción (inelástico).

5.4. Validación del análisis y su impacto energético

Para validar la corrección del método de generación de políticas energéticas y su impacto en el ahorro energético, se han partido los registros en dos mitades y realizado una validación con todas las parejas a un nivel de satisfacción ponderada objetivo del 90 %. Las políticas se generan de forma analítica con datos de un registro de utilización y se validan con datos de otro registro usando el simulador.

Al determinar las políticas y validarlas con el mismo registro, se espera que la satisfacción esté lo más cerca posible del objetivo, a excepción de ligeras desviaciones por efectos de aleatoriedad de los modelos.

La tabla 5.3 resume los resultados obtenidos, indicando la combinación de registros y los valores medios del tiempo de espera y la satisfacción, indicados en la tabla como \bar{T} , \bar{S}_w (ponderada) y \bar{S} (no ponderada), asimismo se indica el valor mediana del tiempo de espera, T_m y la inactividad eliminada total (R). Todos los estudios se han realizado sin consolidación de modelos, para obtener el ajuste más personalizado posible para cada usuario.

Se han utilizado combinaciones de ambas particiones de los registros de utilización para comprobar que el método es suficientemente robusto, indicado que los resultados de satisfacción ponderada (S_w) se alinean con los objetivos. Cuando se utiliza el mismo registro tanto para ajustar modelos como para simular un parque, se espera que el valor de S_w obtenido coincida exactamente con el objetivo, salvo pequeñas variaciones por redondeo de algún intervalo simulado. Sin embargo, al utilizar registros diferentes, se espera una variabilidad mayor, debido al cambio de comportamiento de los usuarios.

Los tests 2 y 4 son interesantes porque el registro 1.1 sólo contiene un día de fiesta (1 de noviembre). Cuando se usa el registro 1.2 para obtener los modelos y el 1.1 para simular el parque (test 4), se observa cómo la satisfacción obtenida baja ligeramente por efecto de tener un modelo que no considera inactividades largas del día de fiesta. Una situación similar ocurre con los tests 6 y 8, pero con las vacaciones de semana santa. La tabla muestra el valor mediana del tiempo de espera (T_m), indicando el mismo efecto.

La inactividad eliminada (R) se mantiene entre el 86 % y el 95 % en todos los casos, indicando cómo una gran parte del tiempo que los computadores pasan encendidos se debe a periodos como los fines de semana o las noches, donde pueden obtenerse buenos ahorros sin perjudicar la satisfacción de los usuarios.

Test	Registro ajuste	Registro validación	\bar{T}	\bar{S}_w	\bar{S}	R
1	1.1	1.1	385,30 s	90,56% ± 3,26	74,87% ± 28,74	86,76%
		1.2	$T_m = 223,54s$	90,79% ± 11,15	75,92% ± 28,92	87,48%
3	1.2	1.2	388,56 s	90,75% ± 4,23	77,30% ± 28,10	87,11%
		1.1	$T_m = 220,23s$	86,37% ± 12,96	68,52% ± 33,21	86,99%
5	2.1	2.1	350,34 s	90,49% ± 3,54	75,65% ± 28,70	94,16%
		2.2	$T_m = 195,36s$	89,75% ± 10,40	66,12% ± 37,97	95,07%
7	2.2	2.2	307,73 s	90,84% ± 3,57	75,94% ± 29,05	94,42%
		2.1	$T_m = 169,38s$	78,83% ± 12,69	59,46% ± 37,60	95,22%

Tabla 5.3.: Comparación de resultados para la generación de políticas energéticas.

Respuesta a la pregunta A.1: *¿Cuál es el potencial de ahorro energético con un sistema dinámico gestión de energía?*

Como se puede observar tanto de forma analítica como empírica, el ahorro de energía potencial al aplicar un sistema de gestión energética a un parque de computadores es sustancial, llegando a superar el 90 % de la energía malgastada aún cuando la satisfacción ponderada se sitúa en un objetivo del 90 %. Si bien la cantidad ahorrada dependerá del parque, en los estudios realizados en esta tesis, siempre se ahorra la energía correspondiente a más del 85 % del tiempo de inactividad. Asimismo, y tal como se indica en la respuesta a la pregunta B.2, el ahorro dependerá también del nivel de satisfacción seleccionado.

5.4.1. Impacto de la consolidación de modelos

El generador de modelos puede consolidar los modelos por tiempo, por usuario, o por ambos, tal y como se detalla en la sección 4.4.2. Las tablas 5.4 y 5.5 resumen el impacto de todas las opciones de consolidación disponibles para los tests 2 y 6 (se han escogido estas dos configuraciones como las más representativas al tener unos registros de ajuste y validación diferentes), indicando el valor medio (\bar{T}) y mediana (T_m) del tiempo de espera y otras métricas.

	Consolidado por usuario	No consolidado por usuario
Consolidado por tiempo	$\bar{T} = 407,99s$ $T_m = 407,99s$ $\bar{S}_w = 91,83 \% \pm 14,75$ $\bar{S} = 76,87 \% \pm 32,82$ $R = 84,87 \%$	$T = 463,36s \pm 142,46$ $T_m = 430,83s$ $\bar{S}_w = 92,75 \% \pm 14,78$ $\bar{S} = 77,44 \% \pm 32,59$ $R = 82,89 \%$
No consolidado por tiempo	$\bar{T} = 429,43s \pm 195,04$ $T_m = 374,56s$ $\bar{S}_w = 91,04 \% \pm 14,11$ $\bar{S} = 76,55 \% \pm 32,14$ $R = 86,35 \%$	$T = 385,30s \pm 454,93$ $T_m = 223,54s$ $\bar{S}_w = 90,79 \% \pm 11,15$ $\bar{S} = 75,92 \% \pm 28,92$ $R = 87,48 \%$

Tabla 5.4.: Resultados del test 2 con y sin consolidación de modelos.

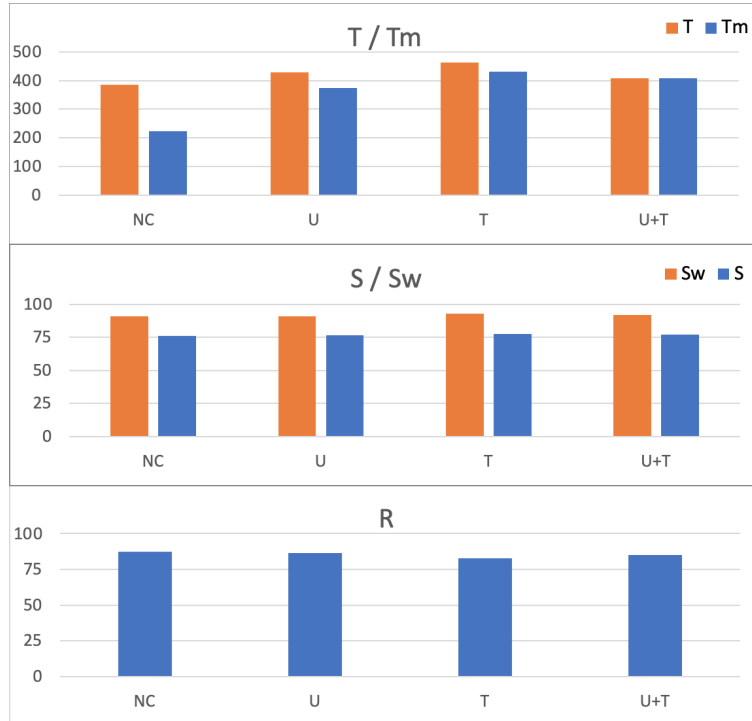


Figura 5.2.: Resultados del test 2 con y sin consolidación de modelos.

	Consolidado por usuario	No consolidado por usuario
Consolidado por tiempo	$\bar{T} = 428,00s$ $T_m = 428,00s$ $\bar{S}_w = 91,43\% \pm 12,88$ $\bar{S} = 69,24\% \pm 32,16$ $R = 93,46\%$	$\bar{T} = 494,93s \pm 452,30$ $T_m = 366,17s$ $\bar{S}_w = 90,37\% \pm 15,86$ $\bar{S} = 67,28\% \pm 33,12$ $R = 93,87\%$
No consolidado por tiempo	$\bar{T} = 517,61s \pm 297,46$ $T_m = 483,63s$ $\bar{S}_w = 91,93\% \pm 13,67$ $\bar{S} = 71,12\% \pm 32,24$ $R = 92,80\%$	$\bar{T} = 350,34s \pm 417,90$ $T_m = 195,36s$ $\bar{S}_w = 89,75\% \pm 10,40$ $\bar{S} = 61,12\% \pm 37,97$ $R = 95,07\%$

Tabla 5.5.: Resultados del test 6 con y sin consolidación de modelos.

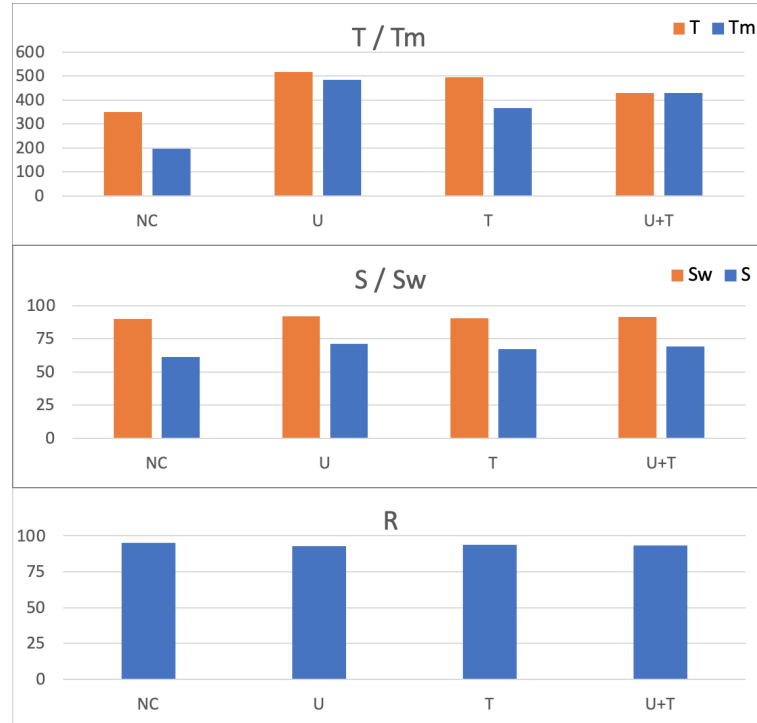


Figura 5.3.: Resultados del test 6 con y sin consolidación de modelos.

Se incorporan las figuras [5.2](#) y [5.3](#) para facilitar observar las diferencias de los resultados con los diferentes métodos de consolidación (en las figuras, NC indica no consolidado, U consolidado por usuario y T por tiempo).

Aunque existen pequeñas diferencias entre los diversos modos de consolidación, los resultados muestran que el impacto en la satisfacción es marginal (unos 2 puntos porcentuales en el caso del test 2 y alrededor de 1 punto para el test 6). Sí es más notable la reducción de entre 2 y 3 puntos porcentuales en la inactividad eliminada cuando se consolidan modelos, debido al incremento del valor de tiempo de espera fruto de tener modelos menos ajustados al comportamiento individual.

Es importante notar cómo la desviación típica de T decrece cuando se consolidan los modelos: todos los intervalos se usan para modelar los usuarios, de forma que las políticas se extraen de modelos más densos y compartidos por varios usuarios. Estos modelos más densos se ven influenciados en menor medida por intervalos que existan en la cola de la distribución. Por contra, la desviación típica de S_w se incrementa ligeramente debido a que las políticas dejan de ajustar de forma tan personalizada a casa usuario.

Este efecto se explica también al mirar al tiempo de espera mediana, T_m , en ambos casos, el uso de consolidación incrementa su valor, indicando políticas más laxas para mantener la satisfacción. Esto, junto con la disminución de su desviación,

confirma el efecto reductor de la influencia de intervalos de inactividad muy largos en las políticas energéticas, a expensas de menor personalización para cada usuario.

Respuesta a la pregunta A.4: ¿Cómo se relaciona el comportamiento de unos usuarios con el de otros? ¿Merece la pena buscar modelos individuales o en grupo?

Tanto en la sección 4.4.2 como en la 5.4.1 se abunda en los mecanismos de consolidación y el impacto que tienen en la satisfacción y el ahorro energético. Lo cierto es que en los parques sometidos a estudio, si bien se han encontrado relaciones entre unos usuarios y otros de forma cualitativa, el generar modelos conjuntos para todos los usuarios en las diversas dimensiones consideradas no tiene un gran impacto en el ahorro final.

Es conveniente señalar cómo la consolidación de modelos reduce la personalización de las políticas energéticas, efecto no deseable salvo que sea necesario por motivos de ausencia de datos en determinados momentos del día o por estar gestionando un parque de un tamaño muy considerable.

5.5. Generación de parques sintéticos

La herramienta permite generar parques de computadores sintéticos, sin necesidad de recolectar registros de utilización, proporcionando como entrada parámetros sencillos que describen el parque que se está estudiando (tales como el número de usuarios, la duración de la jornada laboral o el tipo de distribuciones para la actividad y la inactividad). Existen dos tipos de parámetros: globales, que describen atributos generales del parque a generar; y temporales, que describen atributos que varían durante el curso de una jornada laboral.

Duración	4 semanas
Número de usuarios (N)	1000
Inicio de la jornada	9:00
Fin de la jornada	17:00

Tabla 5.6.: Valores por defecto de los parámetros globales del generador de parques.

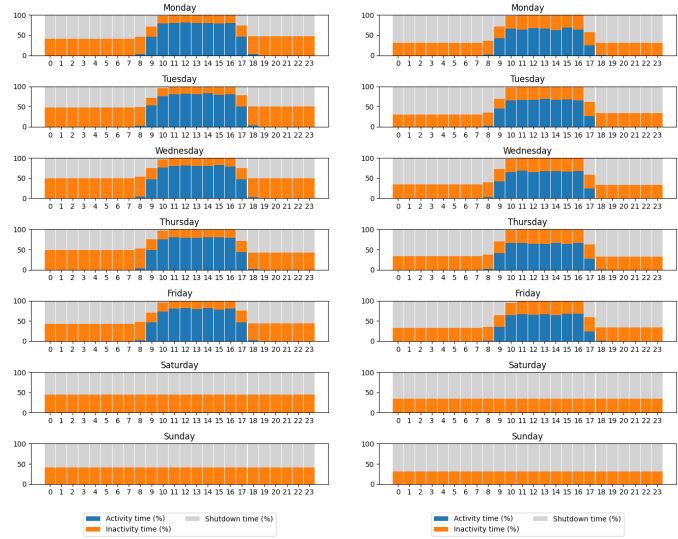
	Horas laborables	Fuera de horario laboral
Actividad	$\mathcal{LN}(\mu = 3600, \sigma = 60)$	$\mathcal{LN}(\mu = 360, \sigma = 30)$
Inactividad	$\mathcal{LN}(\mu = 1800, \sigma = 60)$	$\mathcal{LN}(\mu = 1800, \sigma = 60)$
Fracción apagado	1 %	70 %
Duración apagado	$\mathcal{LN}(\mu = 1800, \sigma = 900)$	“hasta el próximo día”

Tabla 5.7.: Valores por defecto de los parámetros temporales del generador de parques.

Las tablas 5.6 y 5.7 listan los parámetros globales y temporales que la herramienta utiliza por defecto. En el caso de los parámetros temporales, el más destacado es el que determina la duración de los apagados. Además de depender del momento en el que se encuentra la simulación, la duración de un evento de apagado también depende de cuánto tiempo queda para el comienzo de la siguiente jornada laboral. Por ejemplo, cuando un usuario decide apagar su computador, permanecerá así hasta el siguiente día.

Los valores por defecto se han escogido de forma cualitativa, con valores similares a los observados en los registros de utilización de los parques estudiadas. Un administrador de parques deberá hacer una estimación de los valores que desea usar y comparar variaciones de los mismos. Para las distribuciones se ha escogido la distribución log-normal (\mathcal{LN}), que tiene un comportamiento similar a lo observado experimentalmente en la figura 3.9.

Se han realizado validaciones cualitativas de esta funcionalidad, observando diferentes experimentos y el impacto de las variaciones de los parámetros en el resultado final.



(a) Jornada 9:00 - 17:00,
 $A \sim \mathcal{LN}(\mu = 3600, \sigma = 600)$

(b) Jornada 9:00 - 17:00,
 $A \sim \mathcal{LN}(\mu = 1800, \sigma = 600)$



(c) Jornada 7:00 - 14:00,
 $A \sim \mathcal{LN}(\mu = 3600, \sigma = 600)$

(d) Jornada 7:00 - 14:00,
 $A \sim \mathcal{LN}(\mu = 1800, \sigma = 600)$

Figura 5.4.: Generación de parques con diferente jornada y distribución de actividad (A)

Los gráficos de la figura 5.4 comparan la generación de parques con dos variaciones de la duración de la jornada y la distribución de la duración de la actividad.

Los intervalos de actividad de las figuras 5.4a y 5.4c son mas largos que los de las figuras 5.4b y 5.4d y por ello se puede observar una proporción mas elevada de actividad en las horas correspondientes a la jornada laboral (algo más del 75 % en lugar del 60 %).

Finalmente, las figuras 5.4a y 5.4b en relación con las figuras 5.4c y 5.4d muestran el efecto del desplazamiento de las horas de trabajo especificadas en la simulación.

Capítulo 6.

Conclusiones

En este capítulo se resumen los objetivos alcanzados durante el desarrollo de la tesis y se discuten las limitaciones de la propuesta y las posibilidades de trabajo futuro. Adicionalmente, se repasan las publicaciones derivadas de esta investigación desde el inicio de la investigación.

6.1. Objetivos alcanzados y aportaciones

Esta tesis presenta una técnica novedosa para modelar la utilización de grandes parques de computadores y optimizar su consumo energético, aplicando políticas energéticas que apagan computadores tras periodos de inactividad calculados para cada usuario. Los modelos se generan usando registros de utilización recolectados de forma sencilla usando un agente instalado en cada computador, que permite agregarlos, limpiarlos y enviarlos a un servicio para tratarlos.

En comparación con todos los trabajos previos analizados, la propuesta presentada en esta tesis incorpora la satisfacción de usuario como la medida central (y única) que determina la agresividad hacia el ahorro con la que se generan las políticas de gestión energética. En general, el método presentado en esta tesis es más sencillo y adaptativo y proporciona un único parámetro para encontrar el balance entre satisfacción y ahorro energético.

La determinación de las políticas energéticas se hace en base a la satisfacción de usuario, un concepto central del método presentado en este trabajo. De forma analítica se pueden generar las políticas energéticas para el parque maximizando el ahorro para un nivel de satisfacción determinado.

Se ofrece una herramienta que determina las políticas de forma analítica, pero además permite realizar simulaciones para validar las políticas con variantes de parques y generar resultados con parques sintéticos (en lugar de obtener registros de utilización en un parque real).

Todos los modelos implementados son sencillos de explicar y depurar, ya que se resumen en tan sólo cuatro distribuciones por usuario, permitiendo además retener una gran personalización, al tener cada usuario su propio modelado independiente. Los administradores del parque pueden controlar la generación de modelos mediante el objetivo de satisfacción y la utilización de las opciones de consolidación de modelos, que permiten ajustar modelos con más datos reduciendo la personalización de los mismos.

En este sentido, los objetivos de investigación propuestos se han alcanzado, por un lado la definición de un mecanismo de modelado que permite representar el comportamiento de los usuarios y por el otro un método para determinar políticas de gestión energética en base a una métrica de satisfacción. Se proporciona una herramienta que implementa múltiples casos de uso tales como la generación de políticas de forma analítica, la simulación con diversas opciones de parques para validar los resultados o la generación de registros de utilización de parques genéricos.

6.1.1. Preguntas de investigación

Todas las preguntas de investigación planteadas al inicio de los trabajos y explicadas en la introducción han sido respondidas, en particular:

- P. A.1:** *¿Cuál es el potencial de ahorro energético con un sistema dinámico gestión de energía?* El ahorro de energía potencial al aplicar un sistema de gestión energética a un parque de computadores es sustancial, llegando a superar en los escenarios estudiados el 90 % de la energía malgastada aún cuando la satisfacción se sitúa en un objetivo del 90 %.
- P. A.2:** *¿Cuáles son los mejores modelos para el comportamiento de los usuarios?* Se han estudiado diversas opciones para modelar el comportamiento de los usuarios, resultado la que ofrece una mayor sencillez y facilidad para calcular políticas energéticas un modelo estocástico donde se representa el comportamiento de los usuarios mediante distribuciones ajustadas en base a datos recopilados en los parques a gestionar antes de establecer políticas energéticas.
- P. A.3:** *¿Cómo modelar la satisfacción de los usuarios?* Esta tesis propone modelar la satisfacción como la fracción de intervalos de inactividad que fuerzan al usuario a encender su computador al volver a usarlo. También se añade el concepto de satisfacción ponderada, que permite adaptar las políticas de forma más satisfactoria para los casos de intervalos de larga duración.
- P. A.4:** *¿Cómo se relaciona el comportamiento de unos usuarios con el de otros? ¿Merece la pena buscar modelos individuales o en grupo?* Tanto en la sección [4.4.2](#) como en la [5.4.1](#) se abunda en los mecanismos de consolidación y el impacto que tienen en la satisfacción y el ahorro energético. Lo cierto es que en los parques sometidos a estudio, si bien se han encontrado relaciones entre unos usuarios y otros de forma cualitativa, el generar modelos conjuntos para todos los usuarios en las diversas dimensiones consideradas no tiene un gran impacto en el ahorro final.
- P. B.1:** *¿Cómo generar políticas energéticas en base a la satisfacción y el comportamiento de los usuarios?* La generación de políticas tiene como objetivo la selección un valor para el tiempo de espera del gestor energético que contiene el sistema operativo de los computadores del parque. Este valor no puede

sino depender del comportamiento del usuario y de cuánta satisfacción se desea garantizar.

El comportamiento del usuario se modela mediante distribuciones de los tiempos esperados de actividad e inactividad, que van variando a lo largo del día. Estas distribuciones, en particular las de inactividad, nos permiten seleccionar un valor del tiempo de espera utilizando un estimador de satisfacción, que tiene como parámetro una distribución de inactividad (que variará en función del momento del día y del usuario).

- P. B.2:** *¿Cómo se relaciona la satisfacción con el ahorro de energía?* La relación entre el nivel de satisfacción objetivo y el ahorro energético es de proporcionalidad inversa, pero no lineal.

La relación entre ambas variables depende de varios factores. En primer lugar, de variables del entorno tales como el parque y el perfil de uso de los computadores por parte de sus usuarios. En segundo lugar, depende de la función que se utilice para estimar la satisfacción, por ejemplo si se usa la función de satisfacción ponderada presentada en esta tesis, se mejora notablemente el ahorro para el mismo nivel de satisfacción. Finalmente, el nivel de satisfacción objetivo escogido para el parque determinará no solo el potencial de ahorro total, sino también la elasticidad del ahorro energético.

- P. B.3:** *¿Cómo se implementan las políticas en el parque?* El gestor de parques genera de forma analítica una política para cada usuario y hora del día. Esta política consiste en un valor del tiempo de espera, de forma que superado ese tiempo de inactividad se apaga el computador hasta la vuelta del usuario. Un agente recibe desde el nodo de análisis políticas actualizadas cada hora del día.

Las políticas energéticas se aplican en el parque mediante los gestores de energía del sistema operativo de cada computador. El agente actualiza de forma automática la configuración del sistema operativo y es éste el que cambia el estado ACPI correspondiente.

- P. B.4:** *¿Cómo hacer experimentos con diversas variantes del parque?* Para realizar experimentos con variantes del parque a estudiar, se necesita emplear un proceso de simulación, ya que el análisis solamente no es suficiente.

En esta tesis se propone utilizar un método de simulación por eventos discretos que permite utilizar los mismos modelos que se utilizan para el análisis y generación de políticas. Estos modelos se pueden generar mediante tanto registros de utilización como generando modelos sintéticos basados en una serie de parámetros definidos por el usuario.

Para la generación de variantes, se modifican los modelos conforme a las necesidades del administrador del parque: se pueden añadir más computadores en base a los conocidos, reducir el número, alterar los tiempos de simulación o utilizar otros modelos diferentes durante la simulación para comprobar

el buen comportamiento de los modelos utilizados para la generación de políticas.

El simulador utiliza exactamente el mismo método de análisis para la generación de políticas, de forma que representa bien lo que ocurriría en la realidad. Este estudio de variantes es más ágil y barato que plantear cambios reales en el parque estudiado (en otras ocasiones incluso imposibles, por ejemplo cuando se está estudiando un parque de nueva creación).

6.2. Limitaciones y ampliaciones

El trabajo presentado en esta tesis se aplica únicamente a parques de computadores de escritorio, en lugar de un espectro más amplio, tales como dispositivos móviles o centros de datos. La aplicación de las políticas se realiza mediante un sencillo agente, en lugar de un mecanismo de configuración de parques más sofisticado.

La implementación de las políticas se realiza mediante el gestor energético del sistema operativo, que generalmente permite determinar tiempos de espera no sólo para el apagado (estado ACPI S5) sino además de entrada en estados intermedios como suspensión (S3 ó S4) o para dispositivos concretos como los monitores o los discos. La generación de políticas propuesta en esta tesis sólo permite determinar un valor de tiempo de espera en cada momento, ya sea para suspensión o apagado.

Como trabajo futuro se plantea la utilización de múltiples estados ACPI de forma progresiva, en lugar de solamente apagar los computadores directamente. Además, una posible ampliación sería la adaptación de las políticas en tiempo real, por ejemplo utilizando alguna de las técnicas de ventana deslizante descritas en el análisis del estado del arte, lo que permitiría tener políticas que se adaptan de forma continua, en lugar de cada hora de la semana.

Asimismo, pueden plantearse optimizaciones más detalladas en cuanto al mecanismo de ponderación de la función de satisfacción S_w , incorporando opciones como rampas dinámicas, inactividades mínimas u objetivos variables a lo largo del día.

6.3. Publicaciones derivadas de la tesis

A medida que se progresaba en el trabajo de investigación, se han publicado resultados intermedios en varios congresos y revistas con el objetivo de incorporar revisiones por parte de la comunidad investigadora y publicitar el trabajo realizado.

La primera publicación, presentada en 2014 en el congreso Advances In Computer and Electronics Technology (ACET) [García et al. 2014] contiene la primera propuesta para el sistema de gestión energética, así como la arquitectura del sistema de recolección de registros de utilización. Esta publicación contiene las primeras definiciones de conceptos fundamentales, tales como satisfacción, inactividad eliminada o un estudio entre las variables y su relación.

En 2015, en el congreso International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS) se publica la segunda publicación [Medrano Llamas et al. 2015]. En este caso se analiza el comportamiento de los usuarios en base a registros de utilización recolectados en una empresa local, y se determinan propiedades como el ajuste a leyes exponenciales o el comportamiento cíclico a lo largo de la semana. Asimismo se incluye la primera versión de la herramienta, en este caso centrada en su caso de uso de simulación, y un extenso análisis de los datos recopilados desde el parque a estudio.

Finalmente, en 2020 se publica un artículo en la revista Simulation Modelling Practice and Theory (SIMPAT), con factor de impacto JCR 3,272 a cierre de 2020 y del primer cuartil en la categoría de ingeniería del software [Medrano Llamas et al. 2021], que ya contiene todos los conceptos desarrollados para el mecanismo generación de políticas energéticas, en particular se incluye el concepto de satisfacción ponderada, el ciclo de vida de la gestión energética y una extensiva validación de los modelos y el método completo. Otra novedad de esta publicación es la inclusión de ajustes de modelos basados en la distribución empírica en lugar de leyes exponenciales.

Apéndices

Apéndice A.

Análisis de ajuste de distribuciones

Durante el estudio de los datos de los registros de utilización, se han comprobado numerosas propiedades de los datos obtenidos de los parques bajo estudio, tratando de obtener un ajuste lo más detallado posible, pero a la vez suficientemente genérico como para poder utilizarlo en múltiples casos y perfiles de uso. Este apéndice repasa las técnicas utilizadas para el estudio de los datos.

A.1. Máxima verosimilitud

Como existen decenas de familias de distribuciones, tales como la exponencial y, de hecho, en la familia de las leyes exponenciales hay varias opciones, se necesita una guía para comenzar la exploración de los datos, que califique qué familia y qué distribución son las más apropiadas. Además, deben permitir calcular los parámetros de las mismas.

El proceso de búsqueda se puede automatizar utilizando el coeficiente de correlación PPCC [Filliben \[1975\]](#). Este coeficiente permite calcular el parámetro de forma de la distribución, pero además saber cuál es la más apropiada.

La tabla [A.1](#) contiene los valores del estadístico PPCC para diversas distribuciones de la familia de las leyes exponenciales, proporcionando una guía de qué distribución utilizar en cada caso.

Dist	ppcc
invweibull	0.99418934
lomax	0.99415232
pareto	0.99415232
fisk	0.99414088
loglaplace	0.9929184
genpareto	0.99181516
lognorm	0.98636602
alpha	0.96235088
exponpow	0.95707549
invgauss	0.92826428
pearson3	0.91729804
chi2	0.9099308
gamma	0.87418736

Tabla A.1.: Comparación de valores PPCC para los datos del martes a las 12:00.

Los valores del estadístico PPCC pueden utilizarse para ajustar los modelos con distribuciones dinámicas, primero determinando cuál es la más apropiada y de esta forma usando una diferente según convenga. Por contra, incluso la mejor distribución en relación a PPCC puede presentar un ajuste malo para la cola de la distribución, como se ha visto anteriormente.

A.2. Gráficos Q-Q

Los gráficos cuantil-cuantil (Q-Q, por su nombre en inglés) permiten comparar la distribución de los datos con los que se cuenta con una distribución teórica, representando pares de cuantiles para los valores recogidos. Si el ajuste es bueno, la mayoría de los datos se encontrarán superpuestos a una función identidad ($y = x$), mientras que cuando el ajuste no es bueno, no existirá correlación. Los gráficos P-P (percentil-percentil) son equivalentes, pero presentan una versión normalizada.

Se trata de un método sencillo y efectivo para un primer estudio, ya que permite estudiar de la misma forma numerosas distribuciones teóricas. La figura [A.1](#) muestra una comparativa de los datos del martes a las 12:00 mediante gráficos Q-Q y P-P con las distribuciones teóricas Pareto y Exponencial.

Los gráficos Q-Q son interesantes porque permiten entender cómo se diferencia una distribución teórica de los datos de forma reproducible, además de explicar de forma muy evidente el comportamiento en las colas, mecanismo clave para representar correctamente la actividad e inactividad en el método propuesto en esta tesis.

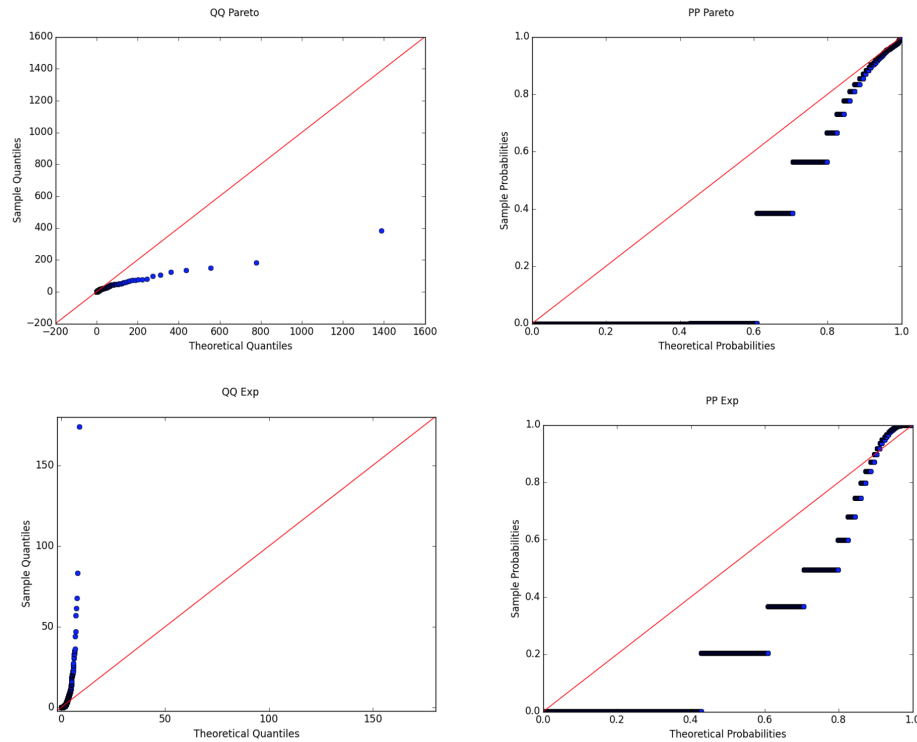


Figura A.1.: Comparación de distribuciones con diagramas Q-Q (1).

Los gráficos Q-Q presentados en la figura [A.1](#) presentan con elocuencia por tanto cómo los datos procedentes del martes a las 12:00 no se ajustan bien a las leyes exponenciales propuestas. La representación Q-Q indica cómo el avance los cuantiles para la función teórica es mucho más rápida que en la muestra correspondiente, eso se debe a que la distribución teórica no ajusta bien masa de probabilidad que existe en la cola de la distribución correspondiente a la muestra.

Este comportamiento se observa en todas distribuciones teóricas, a excepción de la exponencial, que ofrece precisamente el comportamiento opuesto. Tal y como se representa en la figura [A.5](#), la forma de la distribución exponencial es mucho más progresiva, ya que su ajuste se ve afectado por la inusual cola que presentan las muestras.

Cabe destacar que todas las distribuciones de la familia de las exponenciales cuentan con parámetros, tales como los de forma o posición, para los que hay que determinar valores al momento del ajuste. Estos valores se determinan mediante un método de máxima verosimilitud, implementado en el paquete `powerlaw`, utilizado a lo largo de esta tesis. De esta forma, puede no sólo determinarse los parámetros que proporcionan el mejor ajuste para una distribución teórica concreta, sino comparar la calidad de ajuste entre múltiples distribuciones de la misma familia.

A.3. Histogramas y ajuste de leyes exponenciales

Tras el primer estudio de los datos mediante gráficos Q-Q, quedando claro que una ley exponencial era apropiada dada la forma de las muestras, pero no proporcionando directamente un ajuste de máxima calidad, fue necesario encontrar mejores ajustes y estudiar otras familias de distribuciones.

Lo que indicaban los gráficos Q-Q recomendaba realizar estudios detallados sobre la cola de la distribución de los registros de utilización, ya que las distribuciones exponenciales más comunes como la propia exponencial o la Pareto no llegaban a representarla correctamente.

Además de gráficos Q-Q, realizar un ajuste tradicional y observar las diferencias resulta útil. La figura [A.2](#) muestra esa comparativa para los datos del martes 12:00. Se puede observar cómo de las cuatro distribuciones exponenciales (pareto, lomax, Weibull y Frisk) estudiadas presentan un buen ajuste en el centro de masa de la distribución según los gráficos de densidad de probabilidad (siendo la lomax la de mejor ajuste, según los estudios Q-Q). En cualquier caso, la cola presenta grandes diferencias, atestiguadas en los gráficos Q-Q con fuertes divergencias respecto a la función identidad.

En este ejemplo, la función lomax sería la que mejor ajuste en total proporcionaría, ya que los valores del gráfico Q-Q están sobre la función identidad (efecto difícil de observar en el ajuste con el histograma).

Es necesario hacer hincapié en la forma de la cola, esta vez tomando como referencia la escala temporal de los gráficos. Puede verse en las comparaciones de las funciones de densidad con los estudios Q-Q cómo la mayor parte de la masa de la distribución de las muestras se encuentra en intervalos de tiempo menores a unos 400 s, sin embargo existen intervalos de inactividad tan largos como 60.000 ú 100.000 s que sólo pueden apreciarse con visualizaciones como los gráficos Q-Q. Esta característica hace que los ajustes, que tienden a minimizar los errores cuadráticos medios, ponderen demasiado o bien la masa de intervalos cortos o bien los intervalos largos de las colas, pero no sean capaces de representar ambos.

A.3. Histogramas y ajuste de leyes exponenciales

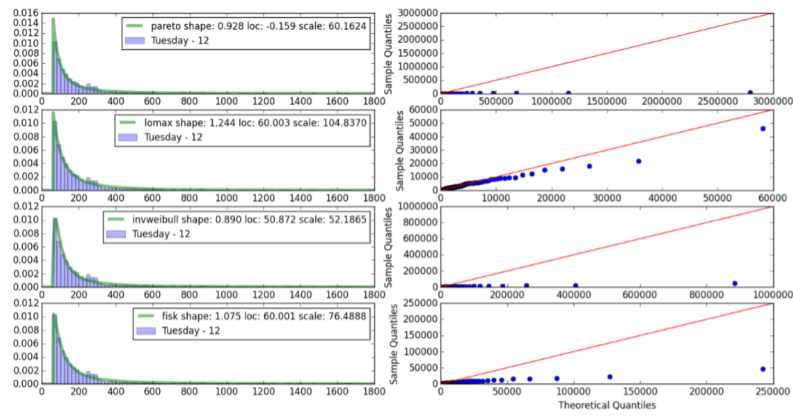


Figura A.2.: Comparación de distribuciones con diagramas Q-Q (2).

La figura [A.3](#) muestra el mismo análisis para los datos del viernes a las 14:00, mostrando otro comportamiento completamente diferente, incluso en distribuciones que funcionan bien para otras horas como la lomax. Claramente, otros métodos u otras distribuciones, adaptadas a cada hora, son necesarias para modelar de forma representativa los datos obtenidos de los parques en estudio.

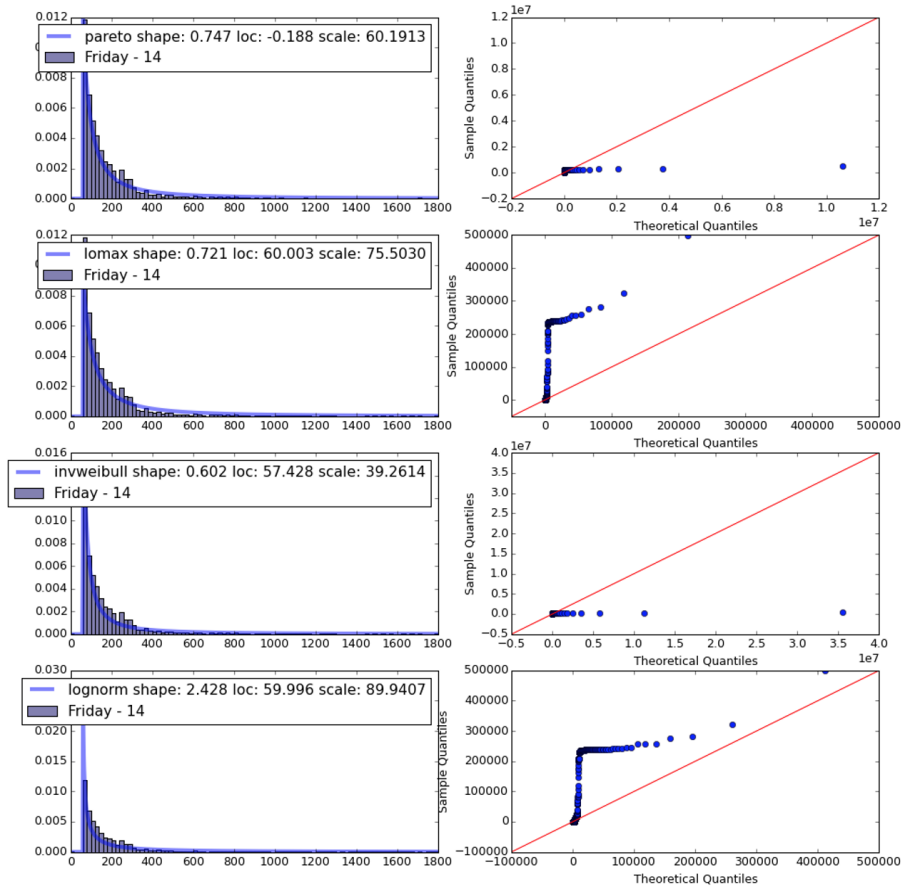


Figura A.3.: Comparación de distribuciones con diagramas Q-Q (y 3).

A.4. Kernel Density Estimation

Kernel Density Estimation (KDE) [Turlach \[1993\]](#) es un método de ajuste que permite estimar la función de densidad de una distribución de forma no paramétrica, esto es, sin tener que escoger qué función utilizar o qué valores como los parámetros de forma utilizar.

Efectivamente, en KDE sí hay que escoger un parámetro, el método de ancho de banda que, de hecho, determina de forma profunda el funcionamiento del algoritmo.

La figura [A.4](#) muestra cómo KDE permite generar ajustes precisos (incluso llegando al overfit).

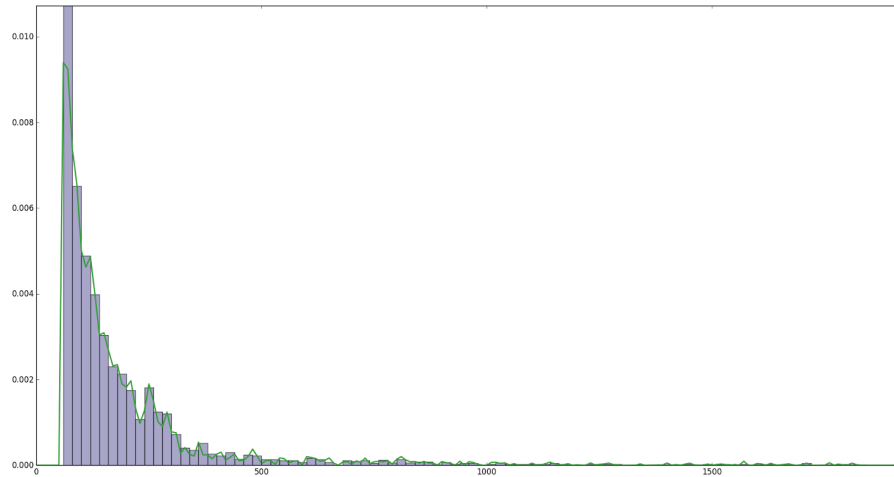


Figura A.4.: Ajuste mediante el método KDE.

Uno de los principales problemas de KDE, además del potencial overfit y la necesidad de escoger cómo determinar el ancho de banda, es lo costoso que es extraer muestras aleatorias del modelo una vez ajustado.

A.5. Simulación

El último método que se ha utilizado para validar el ajuste y lo apropiado del uso de una distribución concreta ha sido la simulación, en concreto por eventos discretos.

Generando eventos acorde a una distribución ajustada sobre los datos de los registros de utilización es sencillo comparar las muestras con datos procedentes de distribuciones ajustadas a los datos, bien de forma gráfica, como en la figura [A.5](#) (para datos del martes a las 11:00), o mediante pruebas de ajuste. Puede observarse, en este caso cómo la simulación mediante una distribución exponencial no se corresponde a lo observado, en este caso por el comportamiento de la cola, que arruina la representatividad en el centro de masa de la distribución.

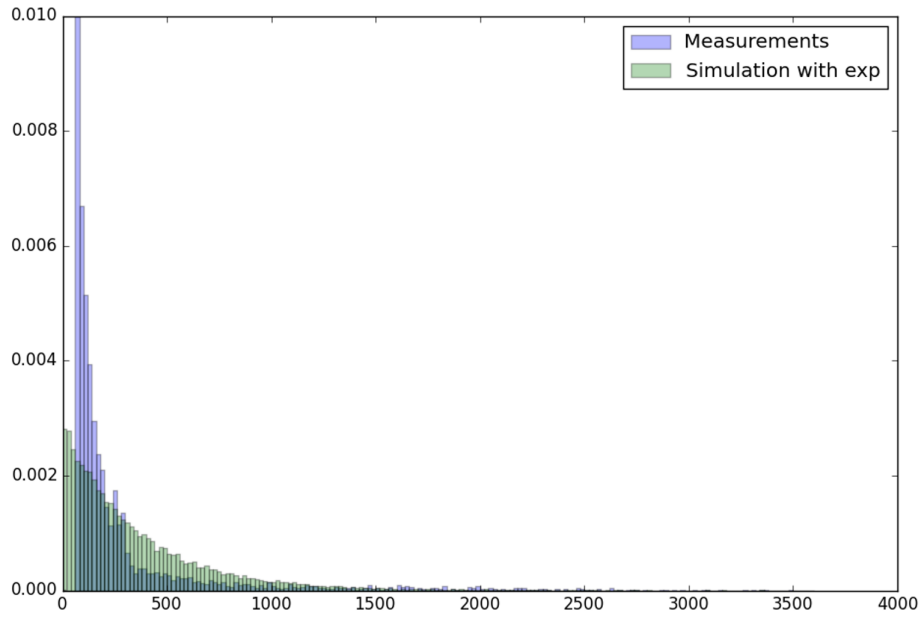


Figura A.5.: Comparación de distribuciones mediante simulación.

Apéndice B.

Guía de uso del software de simulación

La herramienta que se adjunta permite el estudio de registros de utilización y simulación de diferentes parques. Se proporciona como software libre disponible en GitHub [\[1\]](#)

B.1. Descarga e instalación

La instalación se realiza directamente desde la copia local del código fuente, mediante el comando siguiente:

```
$ git clone https://github.com/asi-uniovi/power-simulation.git
$ cd power-simulation
$ make init
```

La inicialización crea un entorno virtual de Python e instala los paquetes y dependencias correspondientes. Pipenv es la herramienta que gestiona el entorno virtual y los comandos para ejecutar la simulación y los análisis.

Además de versiones recientes de Python (3.7 y siguientes), es necesario contar con el paquete `sqlite3` instalado y los *bindings* para Python.

B.2. Ejecución

En cualquier momento, se puede obtener ayuda con el siguiente comando:

```
$ pipenv run ./main.py --help
```

Una vez inicializado el entorno, la ejecución de la herramienta se hace con el comando `run`, por ejemplo:

```
$ pipenv run ./main.py --config=config/2015.ini
```

Las configuraciones de cada parque se proporcionan mediante un fichero de configuración, que incluye los registros descritos en el apéndice [\[C\]](#) y que además incorpora todas las opciones necesarias para el análisis o la simulación a realizar. Este fichero se dispone en formato *.ini*, un sencillo fichero textual con claves y

¹<https://github.com/asi-uniovi/power-simulation>

valores. Entre otras opciones, determina la ubicación de los registros de utilización para ajustar los modelos y validarlos, opciones sobre las distribuciones empíricas y otras opciones de filtrado de datos.

```
$ pipenv run ./main.py --config config/dev.ini --noplot --
  merge_by_hour --merge_by_pc
INFO: 3 servers have been filtered out.
INFO: training_distribution: Merging histogram both per hour and PC.
INFO: 0 servers have been filtered out.
INFO: Parsing done (0.65 s)
INFO: Simulating 13 users during 1123200 s (1.9 week(s)).
INFO: User Satisfaction (US) target is 90%.
INFO: Average global timeout will be 285.50 s (median = 285.50 s, std
      = 0.00 s)
INFO: A priori WUS = 88.10% (median = 100.00%, std = 18.88 p.p.), US
      = 68.08% (median = 81.82%, std = 36.65 p.p.), RI = 90.95%.
INFO: A priori analysis done (0.03 s)
INFO: Run 1: US = 77.44%, RI = 73.48%, timeout = 611.39
INFO: Run 2: US = 77.19% (d = 3.147), RI = 75.93% (d = 31.100),
      timeout = 648.85
INFO: Run 3: US = 77.37% (d = 0.965), RI = 74.46% (d = 8.768),
      timeout = 648.85
INFO: Run 4: US = 77.31% (d = 0.538), RI = 72.39% (d = 8.022),
      timeout = 342.48
INFO: Run 5: US = 77.22% (d = 0.434), RI = 73.19% (d = 5.863),
      timeout = 429.87
INFO: Run 6: US = 77.27% (d = 0.354), RI = 73.08% (d = 4.441),
      timeout = 475.62
INFO: Run 7: US = 77.29% (d = 0.288), RI = 73.68% (d = 3.864),
      timeout = 558.50
INFO: Run 8: US = 77.32% (d = 0.252), RI = 74.28% (d = 3.530),
      timeout = 650.18
INFO: Run 9: US = 77.39% (d = 0.270), RI = 73.70% (d = 3.322),
      timeout = 465.00
INFO: Run 10: US = 77.37% (d = 0.241), RI = 73.43% (d = 2.974),
      timeout = 382.22
INFO: Run 11: US = 77.38% (d = 0.215), RI = 73.67% (d = 2.702),
      timeout = 724.34
INFO: Run 12: US = 77.40% (d = 0.200), RI = 73.17% (d = 2.674),
      timeout = 487.78
INFO: Run 13: US = 77.45% (d = 0.216), RI = 73.30% (d = 2.452),
      timeout = 586.40
INFO: Run 14: US = 77.50% (d = 0.219), RI = 73.85% (d = 2.536),
      timeout = 685.16
INFO: Run 15: US = 77.47% (d = 0.207), RI = 73.75% (d = 2.353),
      timeout = 506.24
INFO: Run 16: US = 77.48% (d = 0.193), RI = 73.62% (d = 2.206),
      timeout = 531.52
INFO: Run 17: US = 77.48% (d = 0.181), RI = 73.30% (d = 2.163),
      timeout = 494.67
INFO: Run 18: US = 77.49% (d = 0.170), RI = 72.98% (d = 2.140),
      timeout = 342.48
INFO: Run 19: US = 77.48% (d = 0.161), RI = 72.94% (d = 2.018),
      timeout = 648.85
INFO: Run 20: US = 77.48% (d = 0.153), RI = 73.23% (d = 2.002),
      timeout = 409.70
```

```
INFO: Run 21: US = 77.48% (d = 0.145), RI = 73.34% (d = 1.911),
      timeout = 605.59
INFO: All runs done (21).
INFO: Simulation runs done (536.31 s)
INFO: All done (total 537.01 s)
```

Durante una ejecución de una simulación, se proporcionan datos sobre el avance de la misma. Ocurre en tres fases: en primer lugar se realiza el procesamiento de los datos y se ajustan los modelos tanto de los registros que se van a utilizar para obtener las políticas como de los que se vayan a utilizar para validarlas, en su caso. A continuación, se realiza el análisis *a priori*, esto es, la generación de políticas en base a los registros de forma analítica y se presentan los resultados. Finalmente, se procede a una serie de simulaciones para su validación o estudio de flotas variantes.

Cuando se solicita una simulación, siempre se realizan una serie de las mismas, dado que los resultados están sometidos a la propia variabilidad del proceso aleatorio. Con cada nueva ejecución se calcula un intervalo de confianza para los valores de satisfacción e inactividad eliminada y se se sigue realizando nuevas ejecuciones hasta que el intervalo de confianza tiene una amplitud menor que un parámetro especificado por el usuario, típicamente entre 0,5 y 2 puntos porcentuales (cuanto más pequeño, mayor precisión de los datos, pero más número de ejecuciones son necesarias).

Los ficheros de configuración tienen el siguiente contenido:

```
[training_distribution]
file = data/Company1-2014_10_05-2014_11_01-no_laptops.json
xmin = 60
xmax = 2332800
duration = 2332800

[activity_distribution]
file = data/Company1-2014_11_02-2014_11_30-no_laptops.json
xmin = 60
xmax = 2419200
duration = 2419200

[stats]
cache_size = 60000
database_name = 2014.db

[simulation]
users = 265
target_satisfaction = 90
satisfaction_threshold = 1800
```

Incorporan la localización de los ficheros que contienen las trazas de utilización, junto con ciertos parámetros para procesarlos, como su duración temporal o indicaciones para filtrar los datos y evitar ruido (*xmin*, *xmax*). En la segunda mitad del fichero se configuran valores globales para la simulación, como las bases de datos de las estadísticas, el objetivo de satisfacción y el número de usuarios a simular.

Apéndice C.

Formato de los registros de utilización

Los registros de utilización se proporcionan en formato JSON, de forma que son independientes de la herramienta y pueden ser utilizados para otros propósitos.

Se exportan desde una base de datos relacional con la que cuenta el recolector de actividad, convertidos en el esquema JSON. Se proporciona un módulo de Python que permite interpretarlos, sin ninguna dependencia, para máxima reutilizabilidad (`tools/parse_trace.py`).

Los registros están estructurados como una lista de diccionarios que contienen las siguientes claves:

PC (string) Identificador único del PC, con uno adicional (“_Total”) que resume todos los datos.

Type (string) Tipo de la variable a la que corresponden los datos del diccionario. Existen hasta cuatro tipos: actividad, inactividad, frecuencia de apagados y duración de apagados (los apagados son siempre dirigidos por el usuario en este caso).

data (list) Corresponde a una lista de intervalos, agrupados por cada hora de la semana. Para realizar la agrupación, la lista contiene hasta 168 diccionarios con tres claves: “Hour” (int) y “Day” (int), que indican la hora de la semana e “Intervals” (list(float)), que se corresponde con una lista de valores en coma flotante. La presencia de horas sin intervalos es opcional y podría añadirse una lista vacía si así se desea.

Apéndice D.

Notación y abreviaturas

ACPI *Advanced Configuration and Power Interface.*

DPM *Dynamic Power Management.*

PDF *Probability Distribution/Density Function*, función de distribución o densidad de probabilidad.

T Valor del tiempo de espera.

I Inactividad.

A Actividad.

$s(I_i)$ Satisfacción de un intervalo de inactividad I_i .

S Función de satisfacción.

S_a Función de satisfacción Apdex.

S_w Función de satisfacción ponderada.

W_s y W_f Parámetros de inicio (W_s) y final (W_f) de la rampa de ponderación de la satisfacción.

R Inactividad eliminada.

S_O Objetivo de satisfacción.

Bibliografía

- 1E NightWatchman. <https://www.1e.com/products/nightwatchman-pc-power-management/>, 2020.
- ACPI Specification, Version 6.3. https://uefi.org/sites/default/files/resources/ACPI_6_3_final_Jan30.pdf, 2020.
- Jeff Alstott, Ed Bullmore, and Dietmar Plenz. powerlaw: a python package for analysis of heavy-tailed distributions. *PLoS ONE*, 9(1), 2014. doi: 10.1371/journal.pone.0085777.
- Aptean Verdiem Surveyor. <https://www.aptean.com/solutions/vertical-business-applications/aptean-verdiem/>, 2020.
- Autonomic Software ANSA. <https://autonomic-software.com/products/power-manager.html>, 2020.
- Avob Energy Saver. <https://www.avob.com/gestion-energetique/>, 2020.
- Lotfi Belkhir and Ahmed Elmeligi. Assessing ict global emissions footprint: Trends to 2040 & recommendations. *Journal of Cleaner Production*, 177:448 – 463, 2018. ISSN 0959-6526. doi: 10.1016/j.jclepro.2017.12.239.
- L. Benini, A. Bogliolo, G. A. Paleologo, and G. De Micheli. Policy optimization for dynamic power management. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(6):813–833, June 1999. ISSN 0278-0070. doi: 10.1109/43.766730.
- L. Benini, A. Bogliolo, and G. De Micheli. A survey of design techniques for system-level dynamic power management. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 8(3):299–316, June 2000. ISSN 1063-8210. doi: 10.1109/92.845896.
- Richard P Brent. *Algorithms for Minimization without Derivatives*. Prentice-Hall, 1973.
- Lory Liza D Bulay-og and Reggie C Gustilo. Power management control system for computer laboratories. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 10(1-8):61–66, 2018.
- Ria Candrawati and Nor Laily Hashim. Analysis of control, learn, and knowledge model for computer power management using unified modelling language. *Advanced Science Letters*, 22(5-6):1327–1330, 2016a. ISSN 1936-6612. doi: 10.1166/asl.2016.6759.

- Ria Candrawati and Nor Laily Hashim. Adaptive approach in handling human inactivity in computer power management. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 8(8):65–69, 2016b.
- Ria Candrawati and Nor Laily Binti Hashim. A learning approach for computer power management. *Advanced Science Letters*, 22(10):2738–2741, 2016c.
- Certero PowerStudio. <https://www.certero.com/products/powerstudio/>, 2020.
- Marshini Chetty, AJ Brush, Brian R Meyers, and Paul Johns. It's not easy being green: understanding home computer power management. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1033–1042. ACM, 2009.
- Eui-Young Chung, Luca Benini, and Giovanni De Micheli. Dynamic power management using adaptive learning tree. In *Proceedings of the 1999 IEEE/ACM International Conference on Computer-aided Design, ICCAD '99*, pages 274–279, Piscataway, NJ, USA, 1999. IEEE Press. ISBN 0-7803-5832-5.
- Eui-Young Chung, L. Benini, A. Bogliolo, Yung-Hsiang Lu, and G. De Micheli. Dynamic power management for nonstationary service requests. *IEEE Transactions on Computers*, 51(11):1345–1361, Nov 2002. ISSN 0018-9340. doi: 10.1109/TC.2002.1047758.
- László Csurgai-Horváth and János Bitó. Primary and secondary user activity models for cognitive wireless network. In *Telecommunications (ConTEL), Proceedings of the 2011 11th International Conference on*, pages 301–306. IEEE, 2011.
- Data Synergy PowerMAN. <http://www.datasynergy.co.uk/products/powerman/>, 2020.
- Dell KACE. <https://www.quest.com/products/kace-systems-management-appliance/>, 2020.
- Gaurav Dhiman and Tajana Simunic Rosing. Dynamic power management using machine learning. In *Proceedings of the 2006 IEEE/ACM International Conference on Computer-aided Design, ICCAD '06*, pages 747–754, New York, NY, USA, 2006. ACM. ISBN 1-59593-389-1. doi: 10.1145/1233501.1233656.
- EventZero Greentrac. <https://enoten.com/Greentrac/>, 2020.
- Faronics Power Save. <https://www.faronics.com/en-uk/products/power-save>, 2020.
- James J Filliben. The probability plot correlation coefficient test for normality. *Technometrics*, 17(1):111–117, 1975.

- Jason Flinn and M. Satyanarayanan. Energy-aware adaptation for mobile applications. *SIGOPS Oper. Syst. Rev.*, 33(5):48–63, December 1999. ISSN 0163-5980. doi: 10.1145/319344.319155.
- Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119 – 139, 1997. ISSN 0022-0000. doi: 10.1006/jcss.1997.1504.
- D.F. García, J. Entrialgo, R. Medrano Llamas, and J. García. A technique to minimize the power consumption of computer fleets. In *Proc. of the Intl. Conf. on Advances In Computer and Electronics Technology- ACET 2014*, pages 100–105, 2014. ISBN 978-1-63248-024-8.
- Chi-Hong Hwang and Allen C.-H. Wu. A predictive system shutdown method for energy saving of event-driven computation. *ACM Trans. Des. Autom. Electron. Syst.*, 5(2):226–241, April 2000. ISSN 1084-4309. doi: 10.1145/335043.335046.
- IBM Tivoli. <https://www-01.ibm.com/software/tivoli>, 2020.
- Ivanti Power Manager. https://help.landesk.com/ld/help/en_US/LDMS/10.0/ISV/tv-Powman.htm, 2020.
- Željko Ivezić, Andrew J Connolly, Jacob T VanderPlas, and Alexander Gray. *Statistics, Data Mining, and Machine Learning in Astronomy: A Practical Python Guide for the Analysis of Survey Data*. Princeton University Press, 2014.
- P. Juang, Qiang Wu, Li-Shiuan Peh, M. Martonosi, and D. W. Clark. Coordinated, distributed, formal energy management of chip multiprocessors. In *ISLPED '05. Proceedings of the 2005 International Symposium on Low Power Electronics and Design, 2005.*, pages 127–130, Aug 2005. doi: 10.1145/1077603.1077637.
- Eun-Sun Jung and Nitin H Vaidya. An energy efficient mac protocol for wireless lans. In *IEEE infocom*, volume 3, pages 1756–1764. Institute of Electrical Engineers Inc. (IEEE), 2002.
- H. Jung and M. Pedram. Dynamic power management under uncertain information. In *2007 Design, Automation Test in Europe Conference Exhibition*, pages 1–6, April 2007. doi: 10.1109/DATE.2007.364434.
- H. Jung and M. Pedram. Supervised learning based power management for multicore processors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(9):1395–1408, Sept 2010. ISSN 0278-0070. doi: 10.1109/TCAD.2010.2059270.
- F. Kong, P. Tao, S. Yang, and X. Zhao. Genetic algorithm based idle length prediction scheme for dynamic power management. In *The Proceedings of the Multiconference on Computational Engineering in Systems Applications*, volume 2, pages 1437–1443, Oct 2006. doi: 10.1109/CESA.2006.4281864.

- Averill M. Law. *Simulation Modeling and Analysis*. McGraw-Hill, 5th edition, 2015. ISBN 978-0-07-340132-4.
- B. Lin, A. Mallik, P. Dinda, G. Memik, and R. Dick. User- and process-driven dynamic voltage and frequency scaling. In *2009 IEEE International Symposium on Performance Analysis of Systems and Software*, pages 11–22, April 2009. doi: 10.1109/ISPASS.2009.4919634.
- Weiwei Lin, Haoyu Wang, Yufeng Zhang, Deyu Qi, James Z. Wang, and Victor Chang. A cloud server energy consumption measurement system for heterogeneous cloud environments. *Information Sciences*, 468:47 – 62, 2018. ISSN 0020-0255. doi: 10.1016/j.ins.2018.08.032.
- Wei Liu, Ying Tan, and Qinru Qiu. Enhanced q-learning algorithm for dynamic power management with performance constraint. In *Proceedings of the Conference on Design, Automation and Test in Europe, DATE '10*, pages 602–605, 3001 Leuven, Belgium, Belgium, 2010. European Design and Automation Association. ISBN 978-3-9810801-6-2.
- Yung-Hsiang Lu, L. Benini, and G. De Micheli. Operating-system directed power reduction. In *ISLPED'00: Proceedings of the 2000 International Symposium on Low Power Electronics and Design (Cat. No.00TH8514)*, pages 37–42, July 2000. doi: 10.1109/LPE.2000.155250.
- S. O. Luiz, A. Perkusich, and A. Lima. Multisize sliding window in workload estimation for dynamic power management. *IEEE Transactions on Computers*, 59:1625–1639, 04 2010. ISSN 0018-9340. doi: 10.1109/TC.2010.90.
- Frank J. Massey Jr. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951. doi: 10.1080/01621459.1951.10500769.
- Ramón Medrano Llamas, Daniel F. García, Joaquín Entrialgo, and Javier Garía. User inactivity modeling and simulation for power management of PC fleets. In *Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems, Spectrs '15*, pages 1–6, San Diego, CA, USA, 2015. Society for Computer Simulation International. ISBN 978-1-5108-1060-0. doi: 10.1109/SPECTS.2015.7285276.
- Ramón Medrano Llamas, Joaquín Entrialgo, and Daniel F. García. Modelling user satisfaction for power-usage optimisation of computer fleets. *Simulation Modelling Practice and Theory*, 108:102263, 2021. ISSN 1569-190X. doi: 10.1016/j.simpat.2020.102263. URL <https://www.sciencedirect.com/science/article/pii/S1569190X20301908>.
- Robert J Minerick, Vincent W Freeh, and Peter M Kogge. Dynamic power management using feedback. In *Workshop on Compilers and Operating Systems for Low Power*, pages 6–1. Citeseer, Citeseer, 2002.

- San Murugesan. Harnessing green IT: Principles and practices. *IT professional*, 10(1):24–33, 2008.
- New Boundary PwrSmart. <https://www.newboundary.com/products/pwrsmart-service>, 2020.
- C. Phillips and S. Singh. An empirical activity model for wlan users. In *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, pages 2065–2073, April 2008. doi: 10.1109/INFOCOM.2008.272.
- Caleb Phillips, Suresh Singh, Douglas Sicker, and Dirk Grunwald. Applying models of user activity for dynamic power management in wireless devices. In *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services, MobileHCI '08*, pages 315–318, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-952-4. doi: 10.1145/1409240.1409277.
- Viswanathan Lakshmi Prabha and Elwin Chandra Monie. Hardware architecture of reinforcement learning scheme for dynamic power management in embedded systems. *EURASIP Journal on Embedded Systems*, 2007(1):065478, Jul 2007. ISSN 1687-3963. doi: 10.1155/2007/65478.
- Giuseppe Procaccianti, Luca Ardito, Antonio Vetro, and Maurizio Morisio. *Energy efficiency in the ICT-profiling power consumption in desktop computer systems*. Prof. Moustafa Eissa, 2012.
- Qinru Qiu, Massoud Pedram, Massoud Pedram, Massoud Pedram, and Massoud Pedram. Dynamic power management based on continuous-time markov decision processes. In *Proceedings of the 36th Annual ACM/IEEE Design Automation Conference, DAC '99*, pages 555–561, New York, NY, USA, 1999. ACM. ISBN 1-58113-109-7. doi: 10.1145/309847.309997.
- Qinru Qiu, Ying Tan, and Qing Wu. Stochastic modeling and optimization for robust power management in a partially observable system. In *Proceedings of the Conference on Design, Automation and Test in Europe, DATE '07*, pages 779–784, San Jose, CA, USA, 2007. EDA Consortium. ISBN 978-3-9810801-2-4.
- D. Ramanathan and R. Gupta. System level online power management algorithms. In *Proceedings Design, Automation and Test in Europe Conference and Exhibition 2000 (Cat. No. PR00537)*, pages 606–611, March 2000. doi: 10.1109/DATE.2000.840847.
- E. Reinhard, B. Champion, N. N. Schulz, R. Gould, E. Perez, and N. Brown. Computer power consumption and management: “earth, wind, and fire: Sustainable energy for the 21st century” research experience for undergraduates. In *2011 IEEE/PES Power Systems Conference and Exposition*, pages 1–8, March 2011. doi: 10.1109/PSCE.2011.5772621.

- Brian Setz, Faris Nizamic, Alexander Lazovik, and Marco Aiello. Power management of personal computers based on user behaviour. In *Proceedings of the 5th International Conference on Smart Cities and Green ICT Systems - Volume 1: SMARTGREENS*, pages 409–416. INSTICC, SciTePress, 2016. ISBN 978-989-758-184-7. doi: 10.5220/0005762604090416.
- Peter Sevcik. Defining the application performance index. *Business Communications Review*, 20, 2005.
- A. Shye, B. Ozisikyilmaz, A. Mallik, G. Memik, P. A. Dinda, R. P. Dick, and A. N. Choudhary. Learning and leveraging the relationship between architecture-level measurements and individual user satisfaction. In *2008 International Symposium on Computer Architecture*, pages 427–438, 2008.
- T. Simunic, L. Benini, P. Glynn, and G. De Micheli. Event-driven power management. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(7):840–857, July 2001. ISSN 0278-0070. doi: 10.1109/43.931003.
- Sergey V Skakun, Nataliya N Kussul, and Alexander G Lobunets. Implementation of the neural network model of users of computer systems on the basis of agent technology. *Journal of Automation and Information Sciences*, 37(4), 2005.
- Susan Solomon, Dahe Qin, Martin Manning, Z Chen, M Marquis, KB Averyt, MMHL Tignor, HL Miller, et al. The physical science basis. *Contribution of working group I to the fourth assessment report of the intergovernmental panel on climate change*, 2007:235–337, 2007.
- M. B. Srivastava, A. P. Chandrakasan, and R. W. Brodersen. Predictive system shutdown and other architectural techniques for energy efficient programmable computation. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 4(1):42–55, March 1996. ISSN 1063-8210. doi: 10.1109/92.486080.
- Y. Tan and Q. Qiu. A framework of stochastic power management using hidden markov model. In *2008 Design, Automation and Test in Europe*, pages 92–97, March 2008. doi: 10.1109/DATE.2008.4484668.
- Georgios Theocharous, Shie Mannor, Nilesh Shah, Prashant Gandhi, Branislav Kveton, Sajid Siddiqi, and Chih-Han Yu. Machine learning for adaptive power management. *Intel Technology Journal*, 10(4), 2006.
- Kazuo Toraichi, Kazuki Katagishi, Iwao Sekita, and Ryoichi Mori. Computational complexity of spline interpolation. *International Journal of Systems Science*, 18(5):945–954, 1987. doi: 10.1080/00207728708964021.
- Igor Turkin and Oleksandr Vdovitchenko. *Model and Methods of Human-Centered Personal Computers Adaptive Power Control*, pages 323–348. Springer International Publishing, Cham, 2019. ISBN 978-3-030-00253-4. doi: 10.1007/978-3-030-00253-4_14.

- Berwin A Turlach. Bandwidth selection in kernel density estimation: A review. In *CORE and Institut de Statistique*. Citeseer, 1993.
- Verismic Power Manager. <https://www.cloudmanagementsuite.com/>, 2020.
- J. M. Walker. Power management for networked computers: A review of incentive programs. In *2009 IEEE International Symposium on Sustainable Systems and Technology*, pages 1–6, May 2009. doi: 10.1109/ISSST.2009.5156753.
- Carrie A. Webber, Judy A. Roberson, Marla C. McWhinney, Richard E. Brown, Margaret J. Pinckard, and John F. Busch. After-hours power status of office equipment in the usa. *Energy*, 31(14):2823 – 2838, 2006. ISSN 0360-5442. doi: 10.1016/j.energy.2005.11.007.