XVI Jornadas de Ingeniería Telemática

JITEL 2023

La Salle – Universitat Ramon Llull

Actas de las XVI Jornadas de Ingeniería Telemática (JITEL 2023), Barcelona (España), 8-10 de noviembre de 2023.

Estrategias de offloading en arquitecturas Fog-Cloud: Un esquema basado en Lyapunov

Neco Villegas*, Luis Diez*, Idoia de la Iglesia†, Marco González-Hierro†, Ramón Agüero*
*Departamento Ingeniería de Comunicaciones, Universidad de Cantabria
e-mail: villegasn@unican.es, {ldiez, ramon}@tlmat.unican.es
†IoT and Digital Platforms Department, Ikerlan Technology Research Centre
e-mail: {idelaiglesia, marco.gonzalez}@ikerlan.es

En este trabajo se introducen políticas de offloading para arquitecturas Fog-Cloud que tienen en cuenta diferentes parámetros de rendimiento. Se afronta el diseño y desarrollo de una plataforma de tres niveles, utilizando técnicas de virtualización, que se puede utilizar para desplegar escenarios con nodos que tienen diferentes características, imitando aquellas de elementos Fog y Cloud. A continuación, se emplea la Teoría de control de Lyapunov para introducir políticas de offloading que equilibren el consumo de energía en los nodos de Fog y el coste monetario de utilizar el Cloud. El esquema propuesto es capaz de encontrar un equilibrio entre estos dos parámetros, garantizando al mismo tiempo la estabilidad del sistema y los requisitos de retardo. A continuación, se compara el algoritmo propuesto con soluciones de referencia (round-robin) y los resultados demuestran que es capaz de ofrecer un mejor rendimiento, incluso en situaciones de elevada demanda y requisitos de energía estrictos. Ajustando los parámetros operativos del algoritmo, los resultados obtenidos demuestran que es capaz de adaptar su comportamiento a diferentes objetivos, evaluando su rendimiento en configuraciones realistas.

***Palabras Clave*—fog, cloud, offloading, Lyapunov, energía, modelado**

I. INTRODUCCIÓN

Hoy en día, el número de servicios en la nube está aumentando continuamente, especialmente aquellos ofrecidos por los principales proveedores, como Microsoft Azure, Amazon Web Services (AWS) y Google Cloud. Este incremento en la demanda viene motivado, entre otras razones, por el aumento de servicios de Internet of Things (IoT) e Industrial Internet of Things (IIoT). Al mismo tiempo, existe una creciente implantación de redes 5G, cuyas tecnologías subyacentes ofrecen varias ventajas, por ejemplo, en términos de latencia, disponibilidad o confiabilidad. Por todo ello, son muchos los sectores que ven ahora una oportunidad para implementar diferentes servicios IoT e IIoT. De hecho, junto con la masiva implementación de redes celulares 5G, el número de conexiones IoT ya ha alcanzado los $14.6 \cdot 10^3$ millones y se espera que supere los $30 \cdot 10^3$ millones en 2027[1]. La gran cantidad de datos generados por estos dispositivos requiere encontrar una arquitectura de sistema adecuada, que sea capaz de acometer su procesado y almacenamiento.

Como resultado, existe un interés creciente en una integración de servicios IoT con Cloud Computing, puesto que la combinación de estas dos tecnologías presenta un gran potencial. Sin embargo, el fuerte incremento de servicios IoT e IIoT y su implementación en nuevos sectores con requisitos más estrictos, pueden llevar a situaciones en las que el Cloud Computing no sea suficiente. En este contexto el Fog Computing ha surgido como su extensión natural que acerca los recursos computacionales a los dispositivos.

El Cloud Computing proporciona alta disponibilidad de recursos computacionales con un consumo de energía relativamente alto (centro de datos), mientras que el Fog Computing proporciona disponibilidad moderada de recursos con un consumo de energía más bajo (servidores pequeños, routers, switches, gateways, etc.). Los entornos Cloud y Fog se pueden usar de manera independiente, pero ambas soluciones se complementan entre sí y, por tanto, la cooperación entre ellas conduce a un uso óptimo de los recursos. Este enfoque conduce a arquitecturas de tres niveles (IoT-Fog-Cloud).

Las principales contribuciones de este trabajo se resumen brevemente a continuación:

1) Utilizando una arquitectura IoT-Fog-Cloud de tres niveles, se propone un esquema de offloading, asignación dinámica de cargas de trabajo, que considera el consumo de energía y el coste monetario en un entorno aleatorio y no controlado.

[1]https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/iot-connections-outlook

2) Se aborda el problema de optimización estocástica resultante mediante la aplicación de la Teoría de Lyapunov, de modo que se reduce a un problema de estabilización del sistema de colas, que puede resolverse con el algoritmo "drift-plus-penalty", esto es, una secuencia de problemas de programación lineal entera (ILP, por sus siglas en inglés).
3) Se realiza un análisis exhaustivo del esquema propuesto en diferentes escenarios y bajo condiciones heterogéneas.

El resto del documento se estructura de la siguiente manera. En la Sección II, se discuten los trabajos existentes relacionados con la combinación de IoT, Fog y Cloud, y otros algoritmos de offloading, señalando en qué se diferencia la propuesta que aquí se presenta. En la Sección III, se describe el modelo del sistema y la solución propuesta para el algoritmo de offloading. A continuación, en la Sección IV se describe la plataforma desplegada para llevar a cabo la evaluación, mientras que en la Sección V se analiza el rendimiento de la solución propuesta. Por último, la Sección VI concluye el documento, resumiendo los aspectos fundamentales, y proporciona una perspectiva del trabajo futuro que surge a partir de la metodología diseñada.

II. TRABAJOS RELACIONADOS

La combinación de IoT e IIoT con Fog y Cloud Computing ha atraído recientemente la atención de la comunidad científica desde diferentes ángulos. Con una perspectiva global, algunos trabajos han propuesto arquitecturas adaptadas a estos entornos [1]–[3]. Aunque estos trabajos comparten el mismo escenario de aplicación que el presentado en este documento, su ámbito de aplicación se sitúa a nivel de arquitectura, mientras que el principal interés de la propuesta descrita se sitúa en la lógica de offloading del procesado de tareas de cómputo y soluciones algorítmicas para obtener comportamientos óptimos. Otros trabajos se han centrado más específicamente en enfoques de offloading [4]–[8].

A diferencia de estos trabajos, la propuesta aquí descrita se centra no sólo en el consumo energético, sino también en el coste monetario, manteniendo la estabilidad de las colas y reduciendo así el retardo. Además, al igual que otras soluciones que utilizan la Teoría de Lyapunov, también tiene en cuenta la evolución temporal del escenario, pudiendo producirse eventos aleatorios. En la Tabla I se compara la propuesta de este trabajo con enfoques similares de la literatura, al menos en sus objetivos. Se seleccionan aquellas soluciones que asumen entornos aleatorios (no controlados) y proponen técnicas para proporcionar una adaptación instantánea. La comparación se realiza en términos del algoritmo de decisión y de los parámetros y métricas de rendimiento que considera. Se incluyen los que se enumeran a continuación:

- **Retardo.** Se refiere al retardo que sufren las tareas o servicios de computación, desde que se generan hasta que se procesan completamente.

Tabla I: Parámetros y métricas de rendimiento considerados en algoritmos de la literatura reciente con escenarios similares al algoritmo propuesto.

	Algoritmo	Retardo	Energía	Coste	Estabilidad
[4]	Optimización distribuida basada en ADMM.	✓	✓	✗	✗
[5]	Programación estocástica basada en Lyapunov.	✓	✗	✗	✓
[6]	Programación estocástica basada en Lyapunov.	✓	✓	✗	✓
[8]	Offloading predictivo.	✓	✓	✗	✓
Prop.	Programación estocástica basada en Lyapunov.	✓	✓	✓	✓

- **Consumo de energía.** Es consecuencia, principalmente, del procesado. Se suele tener en cuenta para los nodos Fog, que tienen capacidades más limitadas.
- **Coste monetario.** Corresponde al coste de utilizar la capacidad de procesamiento del Cloud. Se considera un modelo de "pago por uso", puesto que es lo que ofrecen la mayoría de los proveedores (Amazon, Microsoft, IBM, Google, etc.).
- **Estabilidad.** Hace referencia a la estabilidad de las colas de memoria en el sistema global.

Como puede observarse, la solución propuesta es la única que considera conjuntamente la energía (en los nodos Fog) y el coste monetario (en los nodos Cloud), manteniendo la estabilidad del sistema. Por ello, es posible concluir que complementa y amplía el estado del arte relacionado con la distribución de tareas de computación en despliegues Fog-Cloud.

III. MODELO DE SISTEMA

Esta sección describe el modelo del sistema, así como el diseño del algoritmo propuesto basado en la Teoría de Lyapunov. En la Tabla II se resumen los símbolos utilizados en el modelo propuesto y su significado. Se utiliza el término "servicio" para referirse a conjuntos de paquetes sobre los que se necesita aplicar cierta computación.

Se considera un sistema compuesto por nodos Fog y Cloud con diferentes capacidades de procesamiento. En este escenario, múltiples aplicaciones de usuario generan servicios, compuestos por paquetes, que se envían a los nodos Fog. A continuación, los servicios pueden procesarse localmente (en los mismos nodos Fog) o ser enviados al Cloud. El sistema incluye también un orquestador, o nodo Master, que toma las decisiones de offloading, dependiendo de la política particular implementada.

Sea M el número de aplicaciones que generan servicios. Se supone que el tiempo transcurre en slots y que cada aplicación genera un servicio en cada slot. A su vez, el número de paquetes por servicio sigue una

Tabla II: Símbolos y variables del modelo del sistema.

Notación	Descripción
N	número de puntos de procesamiento, tales como CPUs en el Fog e instancias Cloud.
M	número de aplicaciones independientes generando servicios para ser procesados.
$a_m(t)$	llegadas a la cola de la aplicación m en el slot t, medido en paquetes.
$b_m(t)$	salidas desde la cola de la aplicación m en el slot t, medido en paquetes.
$Q_m(t)$	tamaño de la cola de la aplicación m en el slot t, medido en paquetes.
$\alpha_{m,n}(t)$	decisión para la aplicación m y la CPU n en el slot t, medido en número de paquetes.
$\alpha(\mathbf{t})$	$M \times N$ matriz de las variables de decisión.
$\mathcal{A}(t)$	conjunto de decisiones admisibles en el slot t.
$\omega_n(t)$	tasa de transferencia de la opción de procesamiento n en el slot t. Refleja la variación de la capacidad de procesamiento disponible.
$g_m(t)$	complejidad de procesado de los servicios de la aplicación n en el slot t.
$C(t)$	coste monetario por el uso del Cloud en el slot t.
$k_n(t)$	coste genérico de usar el punto de procesamiento en el slot t.
$E_n(t)$	coste de energía del punto de procesamiento n en el slot t.
E_nth	umbral de energía del punto de procesamiento n.
$G_n(t)$	cola virtual relativa al consumo de energía en el punto de procesamiento n en el slot t.
$\hat{\bullet}$	se utiliza para indicar una función arbitraria que produce una variable $\bullet$.

cierta distribución aleatoria. Los paquetes de los servicios generados se almacenan localmente en las colas de las aplicaciones. Se supone que el escenario tiene N alternativas de procesamiento, incluyendo procesadores locales (alternativas $1, \ldots, N-1$) y un Cloud (alternativa N). En cada slot el nodo Master establece la cantidad de datos de cada aplicación a procesar en cada alternativa, satisfaciendo algunas restricciones. La política de offloading debe garantizar que las colas de aplicaciones permanezcan estables, con el fin de evitar incrementar el retardo. En este escenario, se propone usar la Teoría de Lyapunov, ampliamente utilizada en optimización estocástica para garantizar la estabilidad del sistema.

Sea $a_m(t)$ la cantidad de paquetes que llegan a la cola de la aplicación m, $m \in \{1, \ldots, M\}$, en el slot t y $b_m(t)$ el número de paquetes salientes como consecuencia de la política que se aplica. La dinámica de colas viene dada por la Ec. (1), donde $Q_m(t+1)$ es la cola de espera de la aplicación m en el slot t.

$$Q_m(t+1) = \max[Q_m(t) - b_m(t), 0] + a_m(t) \quad (1)$$

El objetivo es garantizar la estabilidad (promedio) de las colas de aplicaciones, según se describe en la Definición 1.

Definición 1 (Estabilidad de la tasa media): Una cola es estable en tasa media si:

$$\lim_{T\to\infty} \frac{1}{T}\,\mathbb{E}\{Q(t)\} = 0 \quad (2)$$

donde $Q(t)$ es el tamaño de la cola en el slot t y $\mathbb{E}$ es la esperanza matemática.

Sea $\alpha(t)$ una matriz MxN, tal que el elemento $\alpha_{m,n}(t)$ se corresponde con la cantidad de datos de la aplicación m que se asigna al procesador n en el slot t. En cada slot se toma una decisión $\alpha(t)$, dentro de un conjunto $\mathcal{A}(t)$ de posibles elecciones. Además, se supone que existe una tasa de servicio (velocidad de procesado de datos) para cada opción, que dicta cuántos bytes se pueden aceptar en un slot determinado. Es posible definir $b_m(t)$ según la Ec. (3), donde $\omega(t)$ es la tasa de servicio de cada procesador en el slot t, en bytes por slot. En general, se supone que la tasa varía con el tiempo, siguiendo una distribución arbitraria. Como puede observarse, la cantidad de datos drenados por cada aplicación es función de la decisión y de la tasa de servicio de los procesadores. Cabe señalar que esta última puede verse influida tanto por la capacidad de cálculo de la CPU como por la capacidad de comunicación entre la cola de aplicación y el procesador. Por ejemplo, en el procesamiento local la tasa de servicio estaría dominada por la capacidad de cálculo, mientras que en el caso de un procesado remoto (datos que deben enviarse al Cloud) estaría limitada por la capacidad de comunicación.

$$b_m(t) = \hat{b}(\alpha(t), w_1(t), w_2(t), \ldots) = \hat{b}(\alpha(t), \omega(t)) \quad (3)$$

Para evitar la asignación de paquetes inexistentes, en cada slot se impone que la cantidad total de bytes asignados de una cola de aplicación i en la slot t no supere los bytes que hay en dicho instante, como se indica en la Ec. (4).

Además, se asegura que la asignación no supere la tasa de servicio, con la restricción definida en la Ec. (5), donde $g_i(t)$ denota un factor de escala genérico. Por ejemplo, en el caso de una tasa de servicio limitada por la capacidad de cálculo, este parámetro estaría relacionado con la complejidad de dicho cálculo. De este modo, los servicios más complejos producirían tiempos de cálculo más lentos para el mismo número de bytes, lo que se representa escalando el número de bytes, mientras se mantiene constante la capacidad de cálculo.

$$\sum_{j=1}^{N} \alpha_{ij}(t) \leq Q_i(t) \quad \forall i \in \{1, \ldots, M\}, \forall t \quad (4)$$

$$\sum_{i=1}^{M} g_i(t) \cdot \alpha_{ij}(t) \leq w_j(t) \quad \forall t, \forall j \quad (5)$$

Ahora se puede reescribir la salida $b_m(t)$, como se muestra en la Ec. (6).

$$b_m(t) = \hat{b}(\alpha(t), \omega(t)) = \sum_{j=1}^{N} \alpha_{m,j}(t) \quad (6)$$

También se consideran las penalizaciones por utilizar las distintas alternativas de procesamiento. Se utiliza el símbolo $k_i(t)$ para denotar el coste de utilizar la alternativa de procesamiento i en el slot t. Las penalizaciones se definen para las CPU del Cloud y del Fog de formas distintas. Para el procesamiento en el Cloud el objetivo

es minimizar el coste monetario, que está relacionado con la potencia de cálculo necesaria, tal y como se define en la Ec. (7).

$$C(t) = \hat{C}\left(\sum_{i=1}^{M} g_i(t) \cdot k_N(t) \cdot \alpha_{i,N}(t)\right) \tag{7}$$

donde $g_i(t)$ se corresponde, como ya se ha mencionado, con la complejidad de cálculo de los servicios generados por la aplicación i, mientras que $k_N(t)$ es el coste monetario de utilizar el Cloud en el slot t. Como puede verse, el coste es proporcional a la cantidad de tráfico enviado a la instancia del Cloud, escalado por la complejidad computacional. En general, se supone que tanto la complejidad computacional de los servicios de cada aplicación como la tarifa del Cloud pueden variar con el tiempo, siguiendo distribuciones aleatorias arbitrarias.

En cambio, el procesamiento local en el Fog se ve penalizado por el consumo de energía. En este caso, no se busca minimizar este parámetro, sino garantizar que, en promedio, se mantenga por debajo de un determinado valor. Esto sería necesario, por ejemplo, para los dispositivos alimentados por baterías que pueden recargarse periódicamente. Se define la restricción energética de la Ec. (8),

$$E_j(t) = \sum_{i=1}^{M} g_i(t) \cdot k_j \cdot \alpha_{i,j}(t) \quad \forall j \neq N \tag{8}$$

donde k_j representa un mapeo general entre el número de bytes que hay que procesar, escalado por la complejidad del procesamiento, y la energía necesaria para dicho procesamiento. A diferencia del coste del Cloud, el coste asociado a la energía (k_j) dependería principalmente de las características del hardware del procesador, por lo que se supone que no varía con el tiempo: $k_j(t) = k_j \ \forall t \, \forall j \neq N$. En ambos casos, se tienen en cuenta las penalizaciones a lo largo del tiempo, por lo que se utilizan sus expectativas temporales promedio, $\overline{C}$ y $\overline{E}$, que se definen en las Ec. (9) y (10), respectivamente.

$$\overline{C} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \mathbb{E}\{C(t)\} \tag{9}$$

$$\overline{E} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \mathbb{E}\{E(t)\} \tag{10}$$

En conjunto, se busca una política de control que minimice el problema de optimización enunciado en el Problema 1.

Problema 1:

$$\min_{\alpha(t)} \quad \overline{C} \tag{11}$$

$$\textbf{s.t.} \quad \overline{E}_j \leq E_j^{Th} \quad \forall j \in \{1, \ldots, N-1\} \tag{12}$$

$$\alpha(t) \in \mathcal{A}(t) \tag{13}$$

donde E_j^{Th} es el umbral de energía definido para cada procesador de un nodo Fog y $\mathcal{A}(t)$ se cumple para el conjunto de restricciones definidas en las Ec. (4) y (5) en cada slot. Utilizando el entorno de optimización estocástica desarrollado en [9], las desigualdades relacionadas con la limitación del consumo de energía pueden convertirse en colas virtuales, al igual que las colas de aplicaciones definidas anteriormente. Con ello, la actualización de la cola virtual asociada a la energía del procesador de Fog j, G_j, se define en la Ec. (14).

$$G_j(t+1) = \max\{G_j(t) + (E_j(t) - E_j^{Th}), 0\} \tag{14}$$

La cola virtual se introduce como un método para garantizar que se satisface la restricción de consumo promedio de energía. Así, se puede definir el conjunto de colas (de aplicaciones y la cola virtual) como $\Theta(t)$. La función de Lyapunov $L(\Theta(t))$ y su deriva (drift) $\Delta(\Theta(t))$ se definen como se muestra en (15) y (16).

$$L(\Theta(t)) = \frac{1}{2}\left(\sum_{j=1}^{N} G_j(t) + \sum_{i=1}^{M} Q_i(t)\right) \tag{15}$$

$$\Delta(\Theta(t)) = \mathbb{E}\{L(\Theta(t+1)) - L(\Theta(t)) | \Theta(t)\} \tag{16}$$

La solución al Problema 1 es el algoritmo drift-plus-penalty. En cada slot t, se observa el estado de las colas, y se toma una decisión que resuelve el Problema 2, donde V es un factor de ponderación positivo que establece el compromiso entre la deriva y la penalización. Se trata de un problema de programación lineal entera (ILP), que puede resolverse con herramientas existentes.

Problema 2:

$$\min_{\alpha(t)} \quad V \cdot C(t) + \sum_{i=1}^{M} Q_i(t)[a_i(t) - b_i(t)] + \tag{17}$$

$$\sum_{j=1}^{N} G_j(t)(E_j(t) - E_j^{Th}) \tag{18}$$

$$\textbf{s.t.} \quad \sum_{j=1}^{N} \alpha_{ij}(t) \leq Q_i(t) \quad \forall i \in \{1, \ldots, M\}, \forall t \tag{19}$$

$$\sum_{i=1}^{M} g_i(t) \cdot \alpha_{ij}(t) \leq w_j(t) \quad \forall t, \forall j \tag{20}$$

IV. PLATAFORMA DE EVALUACIÓN

Existen varias alternativas para desplegar y gestionar instancias de Fog y Cloud. La mayoría de las grandes empresas tecnológicas proporcionan servicios en la nube, como AWS, Azure, Linode, etc. Por otro lado, existen alternativas que permiten el despliegue de instancias Fog-Cloud propietarias y autogestionables, tanto comerciales (ej. VmWare) como open source (ej. OpenStack, Apache CloudStack, Proxmox). Sin embargo, estas tecnologías no están diseñadas para probar o evaluar el rendimiento de diferentes soluciones de orquestación, sino para gestionar servicios en ejecución. En este sentido, es necesario desarrollar frameworks que llenen el vacío existente entre la evaluación analítica y la planificación, y la evaluación del

rendimiento esperado en entornos controlados. Existen algunos trabajos relacionados en los que se han desarrollado plataformas de arquitectura de tres niveles como [10]–[12], pero tenían algunas limitaciones para los objetivos anteriormente descritos.

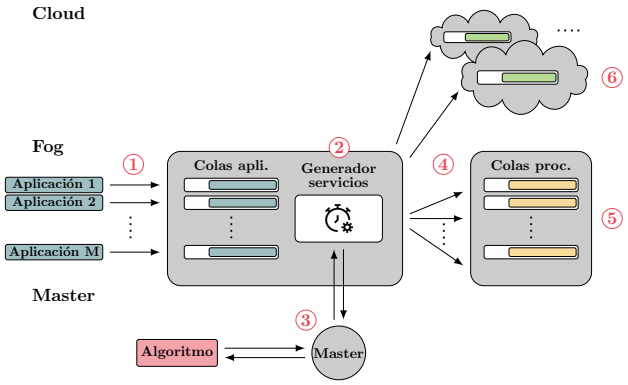


Fig. 1: Vista general de la plataforma Fog-Cloud.

Por ello, se ha desarrollado una plataforma, cuyo diseño se muestra Fig. 1. Abarca tres tipos de elementos que imitan Fog, Cloud y un nodo Master. Los nodos Fog generan flujos de tráfico sintéticos independientes, pertenecientes a diferentes aplicaciones, para lo que se utiliza distribuciones aleatorias configurables (Poisson, uniforme, Lognormal, etc.). El tráfico generado de cada aplicación se almacena en un búfer de entrada, que se representa en la Fig. 1 como el paso 1. A partir del tráfico generado, los nodos Fog definen servicios (tareas de procesamiento) que engloban una serie de paquetes. La generación de servicios, representada en la Fig. 1 con el paso 2, también es configurable, utilizando distribuciones aleatorias. Cuando se definen los servicios, el nodo Fog consulta al Master y este indica el punto de procesado según el algoritmo implementado (paso 3). A continuación, el nodo Fog envía los datos del servicio a procesadores locales o a instancias remotas (paso 4). Por último, se generan registros del rendimiento del sistema (pasos 5 y 6) para cada servicio y la evolución temporal de los estados de todos los dispositivos. Cabe destacar que la plataforma implementa una interfaz genérica para comunicarse con el nodo Master, que es independiente del algoritmo de decisión adoptado. Además, el algoritmo del nodo Master se implementa como un plugin, fácilmente modificable, permitiendo así la comparación de diferentes esquemas de decisión bajo las mismas circunstancias.

Para garantizar una plataforma escalable y ligera todos los nodos se han desplegado en contenedores utilizando Docker. Esto permite a los usuarios desplegar rápidamente múltiples contenedores personalizados en el mismo host. En esta plataforma, cada uno de los contenedores funciona como un nodo Fog, Cloud o Master.

V. RESULTADOS

En esta sección se lleva a cabo una campaña de experimentos utilizando la plataforma desarrollada para analizar el rendimiento de la propuesta de algoritmo de offloading descrito previamente. En primer lugar, se estudian las propiedades del algoritmo en dos escenarios sintéticos, con y sin opción de procesamiento en el Cloud. De esta forma, el primer escenario se centra en el equilibrio entre las limitaciones relacionadas con el consumo de energía y el coste monetario, mientras que la última configuración presta especial atención al impacto que la limitación de energía puede tener sobre el tráfico entrante de la aplicación. A continuación, se evalúa el esquema propuesto en un tercer escenario más realista, en términos de capacidad de procesamiento y generación de tráfico.

Tabla III: Configuración común de la plataforma para la campaña de simulaciones.

Parámetro	Valor
Slots simulados	1000
Duración del slot	1 s
Capacidad de procesamiento de una CPU en el Fog	1 kB/s
Capacidad de procesamiento en un Cloud	100 kB/s
Longitud de un paquete	$200 + 12$ bytes
Tasa de tráfico media agregada	Poisson $[6, \ldots, 30]$ pkt/s
Tasa de generación de servicios	1 serv/s
Factor de energía del Fog (k_j)	1
Coste del Cloud (k_N)	1
Complejidad de la aplicación (g_m)	1
Umbral de energía	$[1, \ldots, 5]$
Número de aplicaciones	3
Número de CPUs en un nodo Fog	2

En estas tres configuraciones, el rendimiento de la solución propuesta se compara con el observado con un algoritmo simple basado en round-robin. Para todas las pruebas se opta por configurar un único nodo Fog con tres aplicaciones y dos procesadores, un nodo Cloud, y un nodo Master que ejecuta los algoritmos. Los detalles de la configuración base que se ha utilizado para las dos primeras pruebas se muestran en la Tabla III. Como puede observarse, en todos los casos se realizan ejecuciones que abarcan 1000 slots de 1 segundo cada uno. En cada slot, las aplicaciones generan un servicio consistente en un número aleatorio de paquetes. En la Tabla III se muestra la tasa media agregada de las aplicaciones y, en cada escenario, se especificará la tasa particular de cada aplicación. Como se puede ver, algunas variables que en el modelo pueden ser aleatorias se definen como constantes, para simplificar la interpretación de los resultados, aunque la utilización de otras opciones no supondría ninguna modificación de la solución propuesta.

A. *Colaboración Fog y Cloud*

Con la primera configuración se pretende analizar el equilibrio de procesamiento entre los nodos Fog y Cloud en diferentes configuraciones. Cabe destacar que la capacidad de procesamiento del nodo Cloud, teniendo en cuenta el tráfico generado por las aplicaciones y la alta capacidad que típicamente tienen los centros de datos, se considera infinita. Se refleja así que, en cualquier caso, será siempre significativamente mayor que la de los nodos Fog.

En primer lugar, se procede a evaluar el impacto del parámetro V, el cual ajusta el compromiso entre con-

sumo de energía y coste monetario. La Fig. 2 muestra la proporción de tráfico enviado al nodo Cloud en función de distintos valores de la tasa de tráfico media agregada. En esta configuración, se fija el valor umbral de energía, E_{th}, en 2. Además, se realizan simulaciones para distintos valores de V. La figura también muestra, con líneas discontinuas, los resultados obtenidos al utilizar el algoritmo round-robin (RR) y una variante de round-robin que tiene en cuenta el umbral de energía prefijado (RRe). Así, la primera opción de round-robin consume toda la capacidad de procesamiento del Fog, enviando el excedente al Cloud. La segunda hace uso de los procesadores del nodo Fog sin superar el umbral de energía, enviando el resto de paquetes al Cloud.

Como era de esperar, se puede observar que una mayor tasa de tráfico conlleva un mayor uso del Cloud. Además, a medida que aumenta el valor de V, vemos una tendencia decreciente en el uso del Cloud. Se identifican 3 regiones de operación en función de dicho parámetro, delimitadas por los algoritmos round-robin. En la primera región, se puede ver que se envía más tráfico al Cloud que con el algoritmo RRe. Esto ocurre con valores de V inferiores a 1, donde se da muy poco peso al coste monetario. El resultado es que no se utiliza la máxima capacidad de procesamiento disponible en el Fog, aunque no se alcance el umbral de energía. La segunda región corresponde a valores de V comprendidos entre 2 y 1000, y el rendimiento observado se sitúa entre las dos versiones round-robin. En esta región, los valores más altos de V reducen significativamente el uso del Cloud y, a su vez, conducen a una eventual saturación de los procesadores del Fog. Con el objetivo de no rebasar el umbral de energía, las soluciones propuestas mantienen el tráfico en las colas de aplicaciones y equilibran las decisiones para garantizar la estabilidad del sistema, teniendo en cuenta las colas de aplicaciones, la energía y el coste.

Este efecto se analiza además estudiando cómo afectan las distintas configuraciones al indicador de rendimiento energético. En la Fig. 3 se representa, con un diagrama de barras, el consumo energético promedio producido por las soluciones propuestas para distintas configuraciones de la tasa de tráfico agregada y para diversos valores de V. Como puede observarse, cuando V se encuentra dentro de la primera región observada en la Fig. 2 ($V = 1$), el consumo está muy por debajo del umbral, independientemente de la tasa de tráfico. A medida que se aumenta el valor de V se observa que el esquema propuesto no es capaz de mantener la energía por debajo del umbral, debido al alto coste de uso de la instancia Cloud. Además, los resultados muestran que el impacto de V también varía con la tasa de tráfico. En este sentido, el consumo de energía se satura con $V = 1e3$ para la tasa de tráfico más baja (6 $\texttt{pkt/s}$), mientras que este valor de saturación aumenta para tasas más altas.

Este primer conjunto de resultados valida el adecuado comportamiento del esquema propuesto, mostrando que es capaz de equilibrar la carga computacional, considerando diferentes parámetros (energía, coste y colas de aplicaciones). Además, se puede configurar para fomentar diferentes comportamientos, gracias al parámetro de operación V.

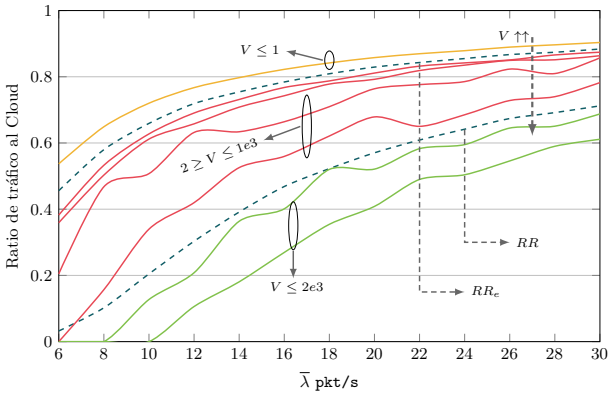


Fig. 2: Uso del Cloud frente a la variación de la tasa de tráfico agregada.

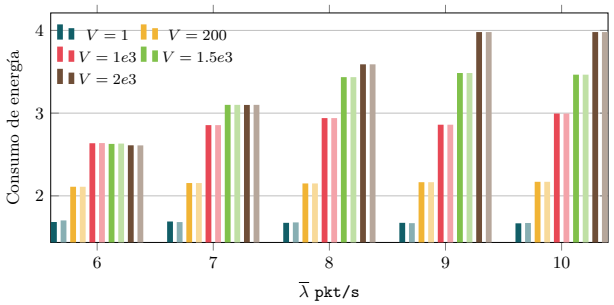
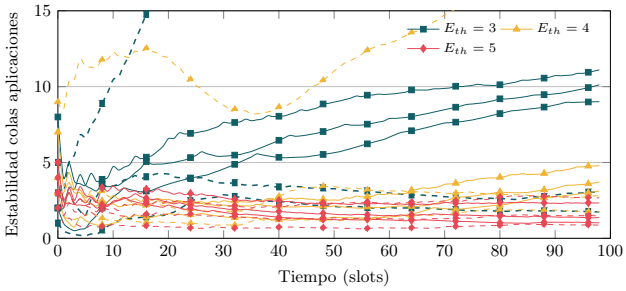


Fig. 3: Coste de energía promedio frente a la variación de la tasa de tráfico agregada.

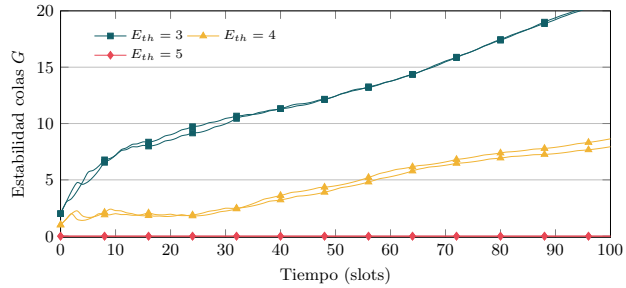
B. Análisis de rendimiento en Fog

Los resultados que se presentan a continuación se centran, con mayor detalle, en el impacto que tienen las distintas configuraciones sobre la energía y las colas de aplicaciones en el Fog. Como se ha visto en la sección anterior, valores bajos de V evitarían la sobrecarga de los nodos Fog, puesto que obligan a enviar muchos servicios al Cloud. Teniendo esto en cuenta, se considera una situación en la que el Cloud y su alta capacidad de procesamiento no estén disponibles, lo que se correspondería con un valor de V alto. Sin el Cloud, es posible evaluar configuraciones más críticas en términos de capacidad de procesamiento, lo que permite observar más de cerca la estabilidad de las colas de aplicaciones.

En este caso, la tasa media agregada de tráfico se fija en 7 paquetes por slot. En concreto, la primera, segunda y tercera aplicación generan 1, 2 y 4 paquetes por slot, respectivamente. La Fig. 4 muestra la estabilidad temporal de las colas de aplicaciones y energía, utilizando la Ec. (2), para distintos valores del umbral de energía. Cabe señalar que umbrales altos son equivalentes a no imponer ningún límite al consumo de energía. Para una mejor representación de los resultados, se muestran los valores observados en los 100 primeros slots. La figura ilustra la estabilidad de la cola de las aplicaciones obtenida



(a) Colas de las aplicaciones. round-robin está representado con líneas discontinuas.



(b) Colas virtuales de energía.

Fig. 4: Evolución de las colas debido a la variación del umbral de energía.

con el algoritmo propuesto y la obtenida con RRe (líneas discontinuas). Los distintos colores corresponden a las estabilidades de las distintas colas de las aplicaciones.

En general, se puede observar que el algoritmo propuesto consigue mantener la estabilidad de todas las colas de las aplicaciones, independientemente de los límites establecidos en el consumo de energía. Como se pone de manifiesto, cuando se utiliza round-robin con el umbral de energía fijado en 3 y 4, hay una cola muy inestable, que corresponde a la aplicación con mayor tasa, mientras que las demás muestran valores bastante bajos. Por otra parte, la solución propuesta es capaz de adaptarse a las tasas de tráfico, como demuestra el hecho de que todas las colas presenten valores similares. Cuando se utiliza un umbral de 3, no se puede garantizar la estabilidad, pero, con restricciones energéticas más suaves, la estabilidad se alcanza rápidamente. La Fig. 4b muestra la estabilidad de la cola virtual de energía (G_j) de cada CPU del nodo Fog. Los resultados muestran una tendencia similar a la observada para las colas de aplicaciones. A medida que se relaja la restricción energética, el esquema propuesto es capaz de estabilizar ambos tipos de colas, mientras que penaliza aquellas con requisitos más estrictos.

C. Entorno realista de generación de tráfico

A continuación se amplía la evaluación utilizando un entorno más realista. En concreto, se ajustan las distribuciones de tráfico de las aplicaciones y, por tanto, las capacidades de los dispositivos. La configuración concreta se muestra en la Tabla IV. Los valores se han obtenido de [13], donde los autores caracterizaron la carga de trabajo de entrada/salida y la distribución de datos de AliCloud, uno de los mayores proveedores de Asia. Como se muestra en la Tabla IV, el tráfico sigue una distribución lognormal. En concreto, el valor esperado y la varianza de la distribución normal subyacente se fijan en 4.5 y 0.8, respectivamente. A su vez, se obtiene una tasa media de tráfico de 125 `pkt/s`[2] Además, se escala la capacidad del nodo Fog en función de las nuevas tasas de tráfico, de forma que pueda hacer frente cómodamente, en media, al tráfico generado por las tres aplicaciones. Así, el escenario configurado permite que el umbral de energía desempeñe un papel importante en los resultados de la simulación.

Tabla IV: Configuración de la simulación del entorno realista.

Parámetro	Valor
Slots simulados	1000
Duración del slot	1 `s`
Capacidad de procesamiento de una CPU en el Fog	100 `kB/s`
Capacidad de procesamiento en un Cloud	10^6 `kB/s`
Umbral de energía	$[75, \ldots, 120]$
Distribución de tráfico para cada aplicación	Lognormal $(4.5, 0.8)$ $(\log pkt/s)$

Bajo esta configuración se pretende identificar configuraciones óptimas del algoritmo según las limitaciones de coste monetario y consumo de energía asequibles. En este sentido, la Fig. 5 muestra una representación en dos ejes del coste monetario (eje izquierdo) y el consumo de energía (eje derecho) con líneas sólidas y discontinuas, respectivamente. Las líneas representan el valor medio obtenido a partir de 30 experimentos independientes, cada uno de los cuales dura 1000 slots. Junto con los valores medios, también se representan el máximo y el mínimo obtenidos durante las simulaciones (fondo sombreado en cada una de las líneas). Se muestran los resultados a medida que se aumenta el valor del umbral de energía E_{th} y para distintos valores del parámetro V.

Como era de esperar, al relajar el umbral de energía el coste disminuye, ya que se procesan más datos en el Fog, mientras que el consumo de energía crece. Las intersecciones corresponden a los puntos (configuraciones) en los que ambos costes son iguales. En este sentido, el esquema propuesto permite establecer configuraciones (V y E_{th}) que igualan los costes de energía y monetarios. Como puede observarse, para el escenario considerado, E_{th} debe fijarse entre 90 y 115 para todo el rango de valores de V. En la Tabla V se indican los puntos de intersección de más valores de V que no se incluyeron en la Fig. 5, para simplificar su representación gráfica. Se puede realizar un análisis similar para diferentes relaciones entre el consumo de energía y el coste del Cloud, o fijando el umbral de energía en lugar del coste del Cloud.

[2]El valor esperado de la distribución lognormal $\mathcal{X}$ viene dado por $\mathbb{E}\{\mathcal{X}\} = \exp(\mu + \frac{\sigma^2}{2})$, donde μ y σ^2 son el valor esperado y la varianza de la distribución normal $\mathcal{N}$, de modo que $log(\mathcal{X}) \sim \mathcal{N}(\mu, \sigma^2)$.

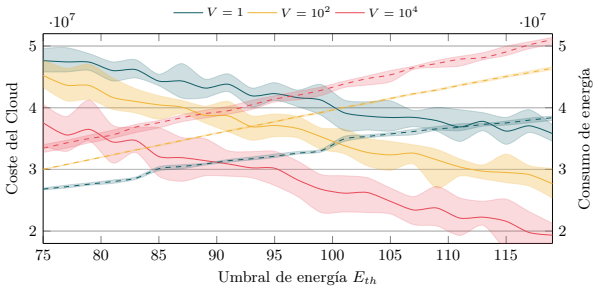


Fig. 5: Coste del Cloud y consumo de energía en función de V y E_{th}.

Tabla V: Puntos de intersección en función del parámetro V.

V	1	10	10^2	10^3	10^4
E_{th}	80.13	91.61	92.96	108.61	113.54
Coste ($\cdot 10^6$)	35.33	37.05	36.96	37.07	37.36

VI. CONCLUSIONES

El enorme volumen de datos generado por los dispositivos IoT precisa encontrar una arquitectura de sistema adecuada capaz de procesar y almacenar la gran cantidad de servicios desplegados. Si bien las arquitecturas basadas en el Cloud se utilizan actualmente con ese propósito, se vislumbra que el nuevo paradigma de Fog Computing permitirá escalar y optimizar las infraestructuras de IoT.

En este contexto se propone, en primer lugar, un modelo de sistema genérico que asume patrones de tráfico variables arbitrarios, capacidad de cálculo disponible y coste en el Cloud. A continuación, se formula un problema de optimización estocástica y se emplea la Teoría de Lyapunov para convertirlo en una secuencia temporal de problemas ILP, que pueden resolverse fácilmente con herramientas existentes. El esquema propuesto se aplica posteriormente a una variedad de escenarios Fog-Cloud, para validar su comportamiento bajo diferentes configuraciones del sistema. Los resultados demuestran que el esquema propuesto es capaz de equilibrar el uso de instancias de Fog y Cloud, consiguiendo regular el consumo de energía y el coste monetario debido al uso del Cloud. Además, se observa que la solución propuesta es capaz de adaptarse a cargas de tráfico desequilibradas, garantizando la estabilidad del sistema incluso bajo situaciones de elevada demanda o para restricciones de energía más estrictas.

Como líneas futuras se plantea ampliar el modelo de diferentes maneras. En primer lugar, se analizará el rendimiento al agregar funciones más complejas (ej. logarítmicas) al problema de optimización, para fomentar diferentes compromisos entre las limitaciones de costo y energía. Además, se analizará la posibilidad de tener en cuenta la ocupación de las colas de los procesadores dentro del modelo del sistema, haciendo que evolucione hacia una red de colas interconectadas, donde se puedan aplicar algoritmos de tipo back-pressure. En este sentido, otra línea futura que subyace es la aplicabilidad de esta estrategia, modificándola para que se base en el retardo de los servicios de computación.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad, Fondo Europeo de Desarrollo Regional, MINECO-FEDER, a través del proyecto SITED: *Semantically-enabled Interoperable Trustworthy Enriched Data-spaces (PID2021-125725OB-I00).*

REFERENCIAS

[1] M. Aazam, S. Zeadally, and K. A. Harras, "Deploying fog computing in industrial internet of things and industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4674–4682, 2018.

[2] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416–464, 2018.

[3] P. Bellavista, L. Foschini, and D. Scotece, "Converging mobile edge computing, fog computing, and iot quality requirements," in *2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2017, pp. 313–320.

[4] Y. Xiao and M. Krunz, "Qoe and power efficiency tradeoff for fog computing networks with fog node cooperation," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017, pp. 1–9.

[5] X. Duan, F. Xu, and Y. Sun, "Research on offloading strategy in edge computing of internet of things," in *2020 International Conference on Computer Network, Electronic and Automation (ICCNEA)*, 2020, pp. 206–210.

[6] J. Xu, L. Chen, and P. Zhou, "Joint service caching and task offloading for mobile edge computing in dense networks," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, pp. 207–215.

[7] Y. Qi, L. Tian, Y. Zhou, and J. Yuan, "Mobile edge computing-assisted admission control in vehicular networks: The convergence of communication and computation," *IEEE Vehicular Technology Magazine*, vol. 14, no. 1, pp. 37–44, 2019.

[8] X. Gao, X. Huang, S. Bian, Z. Shao, and Y. Yang, "Pora: Predictive offloading and resource allocation in dynamic fog computing systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 72–87, 2020.

[9] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*, ser. Synthesis Lectures on Communication Networks. Morgan & Claypool Publishers, 2010. [Online]. Available: http://dx.doi.org/10.2200/S00271ED1V01Y201006CNT007

[10] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.2509

[11] A. Kertész, T. Pflanzner, and T. Gyimothy, "A mobile iot device simulator for iot-fog-cloud systems," *Journal of Grid Computing*, vol. 17, 09 2019.

[12] I. Lera, C. Guerrero, and C. Juiz, "Yafs: A simulator for iot scenarios in fog computing," *IEEE Access*, vol. 7, pp. 91 745–91 758, 2019.

[13] Z. Ren, W. Shi, J. Wan, F. Cao, and J. Lin, "Realistic and scalable benchmarking cloud file systems: Practices and lessons from alicloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 11, pp. 3272–3285, 2017.