



Universidad de los Andes

**MODELADO, OPTIMIZACIÓN Y SIMULACIÓN DE BALANCEO DE FLUJO DE INFORMACIÓN ENTRE IoT Y
FOG COMPUTING**

Proyecto de grado

Enero - Mayo de 2017

Asesor: Yezid Enrique Donoso Meisel

Tatiana Andrea Barbosa Salgado

Código: 201313195

Jairo Ivan Bernal Acosta

Código: 201317123

CONTENIDO

0 Resumen	3
1 Introducción	3
2 Descripción general	4
2.1 Objetivos	4
2.2 Antecedentes	4
2.3 Identificación del problema y de su importancia	4
3 Diseño y especificaciones	5
3.1 Definición del problema	5
3.2 Especificaciones	5
3.2.1 Requerimientos funcionales	5
3.2.2 Requerimientos no funcionales	6
3.2.3 Aceptación de la solución	6
3.3 Restricciones	6
4 Desarrollo del diseño	6
4.1 Recolección de Información	7
4.2 Alternativas de diseño	7
5 Implementación	7
5.1 Descripción de la implementación	8
5.2 Resultados esperados	9
6 Validación - SIMULACIÓN	10
6.1 Métodos	10
6.2 Validación de resultados	11
7 Conclusiones	16
7.1 Discusión	16
7.2 Trabajo futuro	16
8 Referencias	17

0 RESUMEN

FOG computing es una alternativa para cumplir las exigencia de latencia y procesamiento que el internet de las cosas IoT tiene. Sin embargo, aun teniendo la capacidad de procesamiento necesaria el direccionamiento de la información juega un papel fundamental para asegurar el máximo aprovechamiento de FOG computing a al vez del mínimo tráfico en los enlaces. Se propone un modelo matemático para el balanceo de carga de dispositivos IoT a FOG utilizando GAMS y un simulador hecho a la medida en Python para comprobar el comportamiento de la arquitectura optimizada en el tiempo. Se logró tener una pérdida de información significativamente menor al utilizar el direccionamiento obtenido con GAMS en comparación con distribución equitativa y aleatoria.

1 INTRODUCCIÓN

Cloud computing con su modelo de “pagar lo que se consume” permite a las grandes empresas del mundo de hoy manejar sus datos sin pagar datacenter privados y costos. Su escalabilidad permite extender sus servicios sin degradar el desempeño. Sin embargo, aplicaciones que requieran latencia baja y dispositivos en su vecindad para cumplir con sus requerimientos no pueden apoyarse en Cloud ya que por su modelo centralización latencias bajas son costosas(Bonomi, Milito, Zhu, & Addepalli, 2012). Para 2020 se esperan 50 mil millones de dispositivos IoT, la mayoría con aplicaciones de automatización del hogar, ahorro de energía, cuidado de adultos mayores, educación, localización entre otras innumerables funciones. El verdadero valor de los datos generados por estos dispositivos se da cuando son conectados a FOG computing, permitiendo la combinación de información de varias fuentes (Andersson,2014).

Conociendo la arquitectura de conexión de los IoT como FOG así como la capacidad y requerimientos de todos los dispositivos es posible realizar una optimización para asegurar que todas las necesidad se satisfagan y que a la vez todos los dispositivos sean usados lo máximo posible. Todo esto, asegurando que posibles conexiones imprevistas no colapsen la comunicación. Por todo lo anterior el presente proyecto se enfoca en maximizar el uso de los canales de comunicación entre dispositivos IoT y FOG computing asegurando que la carga de cada uno de estos sea equitativa. Esto, con el fin de disminuir la cantidad de información perdida en la transferencia de datos.

La solución propuesta consiste en modelar matemáticamente una arquitectura generalizada de red en GAMS, maximizar el balance de carga en los enlaces y obtener como respuesta el porcentaje de información a enviar desde cada IoT a cada FOG. Con esto, se simuló la arquitectura en el cual cada IOT género aleatoriamente flujos de datos. Esta simulación se compara con otra cuyo balanceo de carga fue aleatorio. La ocupación de la red y la información perdida fue significativamente menor al utilizar el resultado de la simulación para el direccionamiento de información en al red.

El presente documento consiste en primer lugar de a descripción general, los objetivos, el estado del arte y la importancia del proyecto. A continuación, se define el problema a resolver con su especificación y restricciones. Tras esto, se discute el proceso de diseño de la solución al problema y la implementación de la alternativa escogida con su análisis. Luego se validan los resultados obtenidos y finalmente se concluye del trabajo realizado. Para un mejor entendimiento del documento se recomienda tener conocimientos en arquitectura de redes o al menos un contexto

general en FOG e internet de las cosas (IoT).

2 DESCRIPCIÓN GENERAL

2.1 OBJETIVOS

Debido a los avances tecnológico de la época y al desarrollo de herramientas que permiten integrar la mayoría de dispositivos por medio de internet (Internet de las cosas), la cantidad de envío y recepción de datos aumenta cada día. Por esto, el objetivo del presente proyecto es optimizar el balanceo de carga que generan los flujos entre dispositivos de IoT y Fog computing enfocándose en la máxima utilización de los canales de comunicación existentes. De ahí nacen otros objetivos como diseñar el modelo matemático acorde con con el problema, optimizar la carga de los enlaces, simular y comprobar la efectividad de la optimización y su impacto.

2.2 ANTECEDENTES

En cuanto a los trabajos previos relacionados con este problema, encontramos un artículo en el cual Hamid Reza Arkiana, Abolfazl Diyanatb y Atefe Pourkhalilia presentan MIST, un “esquema de análisis de datos basado en Fog con enfoque de optimización de aprovisionamiento de recursos que se pueden utilizar para los propósitos de detección de público de IoT” sin embargo, en este trabajo se realiza una optimización de los recursos lo cual puede responder al problema en cuestión, pero no lo abarca desde el balance de carga de los datos compartidos entre IoT y Fog.

Por otro lado, Yan Sun y Nan Zhang proponen una estructura para Fog computing y presentan un algoritmo que permite optimizar la integración esporádica de recursos de repuesto en la red. No obstante, este trabajo ayuda a mejorar el desempeño de la red pero desde el punto de vista de integración.

2.3 IDENTIFICACIÓN DEL PROBLEMA Y DE SU IMPORTANCIA

Internet de las cosas (IoT) ha revolucionado la computación permitiendo que millones de dispositivos se conecten entre sí predominantemente a través de Wireless networks. Esto tiene como consecuencia, una alta cantidad de datos que deben ser procesados y conducidos a través de un variado grupo de dispositivos hacia Cloud computing. Por lo tanto, el tráfico en la red puede congestionarse con la cantidad de datos enviados. De allí nace la idea de Fog computing. “Fog Computing es una plataforma altamente virtualizada que proporciona servicios de computación, almacenamiento y redes entre dispositivos finales y centros de datos tradicionales de Cloud Computing, normalmente, pero no exclusivamente, situados en el borde de la red” (Bonomi et al., 2012). Es así como se disminuye el tráfico hacia la nube dado que los datos se envían a varios dispositivos de fog, allí se centralizan y se envían a Cloud. Sin embargo, asegurar sus requerimientos de calidad como la capacidad, la memoria y el delay necesarios para la transmisión de datos desde los dispositivos IoT a Fog es una ardua tarea y este es el problema a tratar dado que sin alguno de estos recursos se pueden generar pérdidas de datos. .

3 DISEÑO Y ESPECIFICACIONES

3.1 DEFINICIÓN DEL PROBLEMA

Se desea disminuir la cantidad de información perdida entre IoT-FOG y minimizar la utilización de los enlaces. Esto, considerando que pocos FOG deben atender a miles de IoT. Para este primer estudio se supone: La latencia de todos los FOG es igual, los enlaces son directos entre IoT y FOG. El procesamiento necesario para un flujo de información es igual en todos los FOG y es determinado por el IoT. El tiempo de procesamiento de un flujo depende solo de su tamaño y de la capacidad del procesamiento del FOG.

3.2 ESPECIFICACIONES

3.2.1 REQUERIMIENTOS FUNCIONALES

ID	RF1
Nombre	Diseño de conexiones de red
Resumen	Partiendo de una arquitectura inicial modificar las conexiones con existentes con el fin de perder la menor cantidad de información posible y mantener los canales libres.
Entradas	Arquitectura inicial, dispositivos IoT, FOG, conexiones entre estos y flujos.
Resultados	Matriz con porcentajes de envío de cada flujo desde cada IoT a cada FOG.

ID	RF2
Nombre	Simulación de arquitectura
Resumen	Simular el comportamiento dinámico de una arquitectura en el tiempo.
Entradas	Arquitectura inicial, dispositivos IoT, FOG, conexiones entre estos, flujos y matriz con porcentajes de envío de cada flujo desde cada IoT a cada FOG.
Resultados	Procesamiento, memoria e información perdida en cada minuto en cada FOG.

3.2.2 REQUERIMIENTOS NO FUNCIONALES

Escalabilidad: La solución construida debe ser adaptable a cualquier cantidad de dispositivos IoT y FOG. El desempeño (cumplimiento de restricciones) de las conexiones encontradas debe por lo menos ser igual a él de una arquitectura existente. Deseando que sea mejor.

Interoperabilidad: Se espera que la comunicación entre el simulador y los resultados del diseño de la arquitectura sea fluida.

Replicabilidad: El diseño de la arquitectura debe apoyarse en modelos matemáticos optimizables. Adicionalmente el método de optimización debe permitir resultados replicables.

3.2.3 ACEPTACIÓN DE LA SOLUCIÓN

Los requerimientos de flujo y procesamiento de todos los IoT deben ser satisfechos en su estado inicial. Es posible que en la simulación la arquitectura no cumpla con este requerimiento ya que los dispositivos IoT tienen una tasa de generación de datos variable y los FOG una memoria y capacidad de procesamiento fijos. En esta situación la solución aceptable es la que menos pérdidas tiene. Existe la posibilidad de que la arquitectura sea robusta y sin importar el direccionamiento de la información sea capaz de ejecutar la tarea. Sin embargo este caso no se tendrá en cuenta ya que en la vida real los recursos son limitados.

3.3 RESTRICCIONES

Teniendo en cuenta el problema presentado anteriormente, se quiere diseñar un modelo matemático con el cual se pueda optimizar el balanceo de carga del flujo de datos entre los dispositivos IoT y Fog basándose en la máxima utilizations de los enlaces. Para la optimización, se tiene como restricción principal la licencia requerida por las diferentes herramientas de optimización. Como opciones se tuvieron en cuenta Xpress y Gams. Sin embargo, se utilizó Gams dado que se tuvo acceso a la licencia necesaria para la optimización de casos de tipo real, es decir, con una cantidad de dispositivos IoT mayor a 1000.

Por otro lado, se tuvo una restricción en cuanto a la herramienta usada para la validación por medio de simulación. Se necesitó una herramienta que permitiera poner diferentes parámetros a los dispositivos como lo son la capacidad de almacenamiento y de procesamiento para los dispositivos Fog y la frecuencia para los dispositivos IoT. Adicionalmente, era necesario que la herramienta permitiera manejar la utilización de cada enlace. Teniendo en cuenta estas restricciones, se hicieron pruebas con varios simuladores como ns3 iFogSim y Opnet no obstante, los simuladores no cumplían con los requerimiento por lo que se desarrolló un simulador hecho a medida para el proyecto.

4 DESARROLLO DEL DISEÑO

En primer lugar se definieron los conjuntos N : dispositivos IoT, M : dispositivos Fog y F : Flujos. Luego, se crean algunas variables necesarias para definir las restricciones:

$c(i,j) \rightarrow$ capacidad del enlace entre i y j

$d(i,f) \rightarrow$ Peso del flujo f que sale de i

$alm(j) \rightarrow$ capacidad de almacenamiento de j

$proc(j) \rightarrow$ capacidad de procesamiento de j

$frec(i) \rightarrow$ frecuencia de envío de flujos de i

Además, se definió la variable $x(i,j,f)$ que toma un valor entre 0 y 1 dependiendo del porcentaje de flujo f enviado por el enlace en i y j .

Más adelante, se estableció la función objetivo, la cual, es el máximo de la sumatoria de la suma de la memoria requerida de todos los flujos multiplicada por $x(i,j,f)$ dividido la capacidad de cada Fog, de esta manera:

$$\alpha = \text{Max}(\alpha(i,j))$$

$$c(i,j) * \alpha(i,j) = \sum_f d(i,f) * x(i,j,f)$$

Finalmente, se definen las restricciones. Primero, la restricción de memoria: la memoria disponible en el dispositivo Fog debe ser mayor que la memoria requerida por el flujo. En segundo lugar, la restricción de capacidad de procesamiento: la capacidad de procesamiento en el dispositivo de niebla debe ser mayor que la frecuencia de los flujos. Por último, la suma del peso de los flujos que pasa por el enlace entre i y j debe ser menor a la capacidad de enlace

$$\text{almacenamiento} = d(i,f) * x(i,j,f) \leq \text{alm}(j) * x(i,j,f)$$

$$\text{procesamiento} = \text{frec}(i) * x(i,j,f) \leq \text{proc}(j) * x(i,j,f)$$

$$\text{tamaño del flujo} = \sum_i \sum_j \sum_f (d(i,f) * x(i,j,f)) \leq c(i,j)$$

4.1 RECOLECCIÓN DE INFORMACIÓN

Como fuente de información para el diseño se tuvieron en cuenta los conocimientos adquiridos en la clase Modelado, simulación y optimización dictada por el asesor, Yezid Donoso así como las diapositivas usadas como guía en la clase.

4.2 ALTERNATIVAS DE DISEÑO

Para el diseño se realizó un proceso iterativo, teniendo un modelo se revisaba y se iba cambiando lo elementos que no fuera acorde con las necesidades del problema. Primero, se diseñó un modelo que cumpliera con las leyes de Kirchhoff aplicadas y tomaba la arquitectura como un grafo. Sin embargo, el uso de estas leyes era incorrecto dado que el problema no se puede representar con un grafo estándar. Por otro lado, se creó otro modelo que tenía en cuenta las restricciones pero la función objetivo no iba acorde con las necesidades del problema, no representaba la máxima utilización de los enlaces. Finalmente, se construyó un modelo matemático que tomaba los elementos correctos de los modelos anteriores y se mejoró agregando la restricción del tamaño de los flujos

5 IMPLEMENTACIÓN

En cuanto a la implementación de la optimización, se plasmó el modelo matemático diseñado anteriormente en Gams. Para comprobar el modelo, primero se realizaron pruebas con un caso pequeño de 2 dispositivos Fog y 5 dispositivos IoT. Al hacer las correcciones correspondientes y corroborar el buen funcionamiento del modelo, se ingresó un caso real con más de mil dispositivos

IoT.

5.1 DESCRIPCIÓN DE LA IMPLEMENTACIÓN

Primero, se definieron los conjuntos de dispositivos IoT, Fog y los flujos (Imagen 1).

```
Sets
i   Dispositivos Iot /n1*n2000/
j   Dispositivos Fog / m1*m3 /
f   Flujos           / f1*f4 /;
```

Imagen 1. Conjuntos

Enseguida, se agregaron los parámetros requeridos para las restricciones como se puede ver en la Imagen 2:

```
Table cap(i,j) Capacidad del Enlace
           m1      m2
n1         500    1024
n2         1024    556
n3         800     700
n4         956     256
n5         534    1024;

Parameters
alm(j) almacenamiento de dispositivos Fog en KB
/
m1      8000
m2     20000
/

proc(j) procesamiento de dispositivos Fog en Hz
/
m1      1.5
m2       2
/

frec(i) frecuencia de envio de dispositivos Iot
/
n1      0.1
n2      1.2
n3       1
n4      0.5
n5       1
/;
```

Imagen 2. Parámetros

Luego, se definieron las variables mostradas en la Imagen 3:

Variables	
<code>x(i,j,f)</code>	Indica el porcentaje de f que se debe enviar por el enlace i j
<code>maxij(i,j)</code>	Maxima Utilizacion de cada Enlace
<code>alpha</code>	Minimizacion de maxima utilizacion;

Imagen 3. Variables

Después, se especificó la función objetivo:

```
carga .. alpha =e= smax((i,j),maxij(i,j));
```

Imagen 4. Función objetivo

Posteriormente, se fijaron las restricciones como se muestra en la Imagen 5:

```
alph(i,j) .. c(i,j)* maxij(i,j) =e= sum((f),d(i,f)* x(i,j,f));
almacenamiento(j) .. sum((i,f),d(i,f)* x(i,j,f)) =l= alm(j);
procesamiento(j) .. sum((i,f), frec(i)* x(i,j,f)) =l= proc(j);
flujo(i,j) .. sum((f),x(i,j,f)* d(i,f)) =l= c(i,j) ;
```

Imagen 5. Restricciones

Por último, se seleccionó el solver a usar:

```
Solve transport using dnlp minimizing alpha;
```

Imagen 6. Solver

5.2 RESULTADOS ESPERADOS

Como caso base definimos una arquitectura con 2 dispositivos Fog, 5 dispositivos IoT y cada dispositivo enviaba un flujo con los siguientes pesos (Imagen 7):

```
Flujo 1 → 675 MB
Flujo 2 → 512 MB
Flujo 3 → 500 MB
Flujo 4 → 856 MB
Flujo 5 → 1000MB
```

Imagen 7. Flujos

Con esta arquitectura y el modelo anterior, se obtuvo el resultado mostrado en la Imagen 8:

```

---- 106 VARIABLE x.L Indica el porcentaje de flujo que se debe enviar por
      1 enlace i j

```

	f1	f2	f3	f4	f5
n1.m1	0.682	+INF	+INF	+INF	+INF
n1.m2	0.318	+INF	+INF	+INF	+INF
n2.m2	+INF	1.000	+INF	+INF	+INF
n3.m1	+INF	+INF	1.000	+INF	+INF
n3.m2	+INF	+INF		+INF	+INF
n4.m1	+INF	+INF	+INF	0.921	+INF
n4.m2	+INF	+INF	+INF	0.079	+INF
n5.m1	+INF	+INF	+INF	+INF	0.492
n5.m2	+INF	+INF	+INF	+INF	0.508

Imagen 8. Resultados

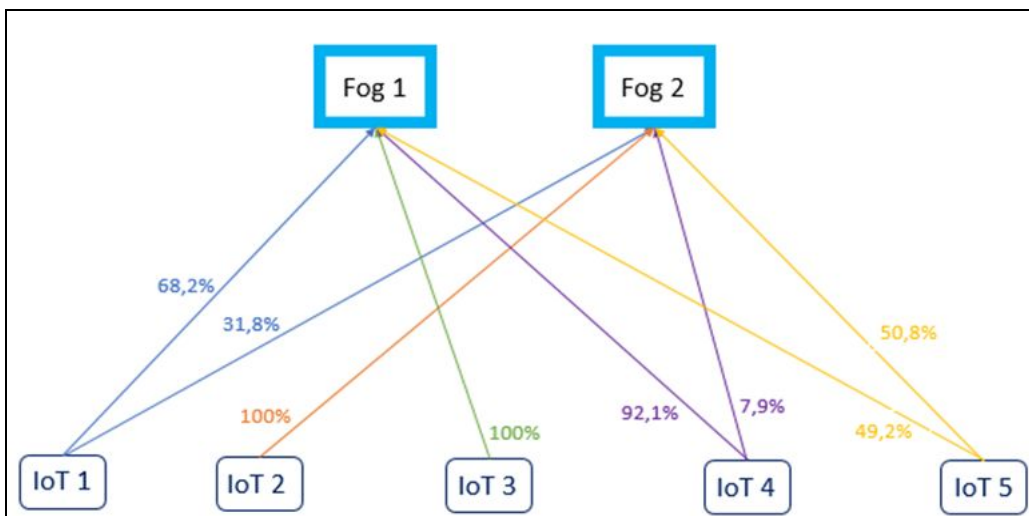


Imagen 9. Diagrama de resultados

En la imagen anterior, se diagrama de forma clara los resultados obtenidos para analizarlos y así comprobar su correcto funcionamiento. Allí, se pueden ver los enlaces entre lo dispositivo y el respectivo porcentaje de flujo que se debe enviar por cada canal para cumplir con la optimización. De este modo se pudo validar la implementación y luego se usó en un caso más grande. Los resultados obtenidos de este caso, se pueden ver mejor reflejado en la simulación.

6 VALIDACIÓN - SIMULACIÓN

6.1 MÉTODOS

Utilizando Python se creó una aplicación cuyos parámetros son los mismo utilizados en el modelo de GAMS, flujos, IoT y fogs con todas sus características. El último parámetro es una matriz con los porcentajes que el flujo generado por el IoT n debe mandar al FOG n.

En el simulador un IoT es definido por un nombre único, un procesamiento necesario y un intervalo de tiempo de envío de datos. Un dispositivo FOG por un nombre único, un poder de procesamiento y memoria. Un flujo representa el intercambio de datos entre un IoT y un FOG y está definido por un porcentaje, un tamaño y un origen y un destino. La simulación inicia con todos los IoT

generando una señal en el tiempo, si el tiempo de simulación es igual al tiempo de la señal esta es transferida a los FOG a través de un flujo. El dispositivo FOG atiende su cola pendiente hasta que su tiempo sea igual al de la simulación o hasta que acabe, tras esto comprueba si tiene suficientes recursos para atender a la petición, en caso afirmativo se almacena en una cola de prioridad ordenada por el tiempo de llegada, si no se agrega a las pérdidas. La ilustración 10 muestra la arquitectura del simulador implementado. El simulador retorna el promedio de las pérdidas, uso de CPU y uso de memoria en cada fog para cada minuto de la simulación en un archivo de excel y en gráficas.

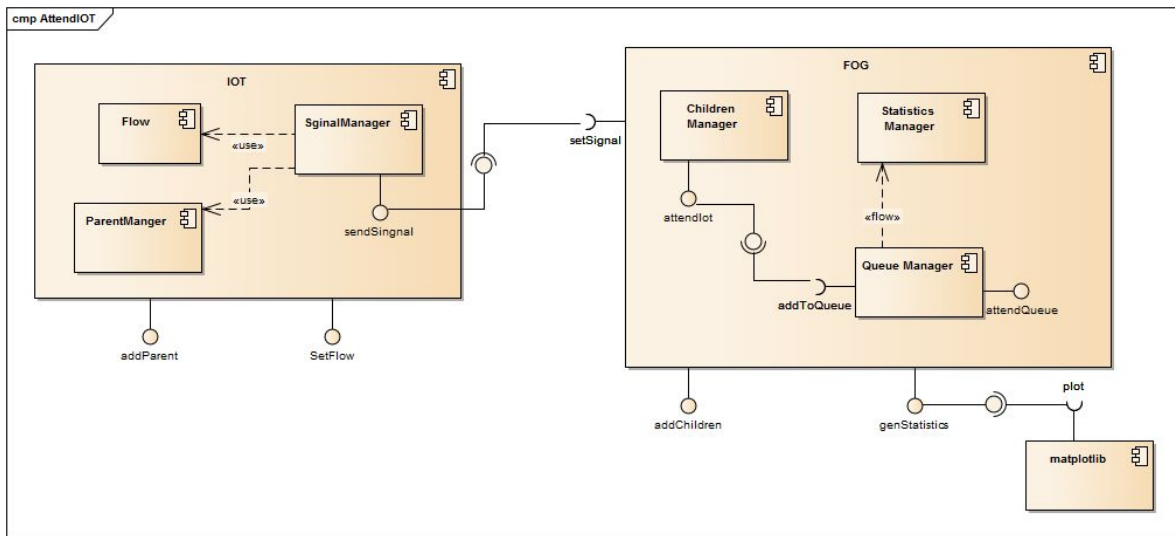


Imagen 10. Arquitectura del simulador

Se harán 3 pruebas de validación. La primera consiste en dirigir los flujos completos de un grupo uniforme de IoT a un fog específico. La segunda es aleatorizar el porcentaje de cada flujo a enviar desde cada IoT a cada FOG. El último es utilizar la respuesta dada por la optimización. Se espera que el uso del modelo matemático reduzca significativamente las pérdidas de las demás simulaciones. Cada simulación se hará 10 veces y una prueba z con una significancia del 95% será utilizada para comparar si la media de pérdidas en las 3 pruebas es igual o si se puede decir que las pérdidas utilizando la optimización son menores. Cada IoT genera datos en un intervalo de 1 a 5 minutos.

6.2 VALIDACIÓN DE RESULTADOS

El resultado de la optimización fue obtenido utilizando CONOPT permitiendo la replicabilidad del mismo. Cada simulación se ejecutó por 100 minutos. De cada una se guardó el total de KB perdidos, el valor de la función objetivo, el promedio de el uso de memoria y CPU en cada minuto. Se utilizaron 2000 IoT, con 4 flujos cada uno, 3 FOG, m1, m2, m3. Existen enlaces de todos los IoTs a todos los FOGs.

CASO I: El resultado de la optimización para el caso a gran escala se presenta en las imágenes 11 y 12 partes a y b.

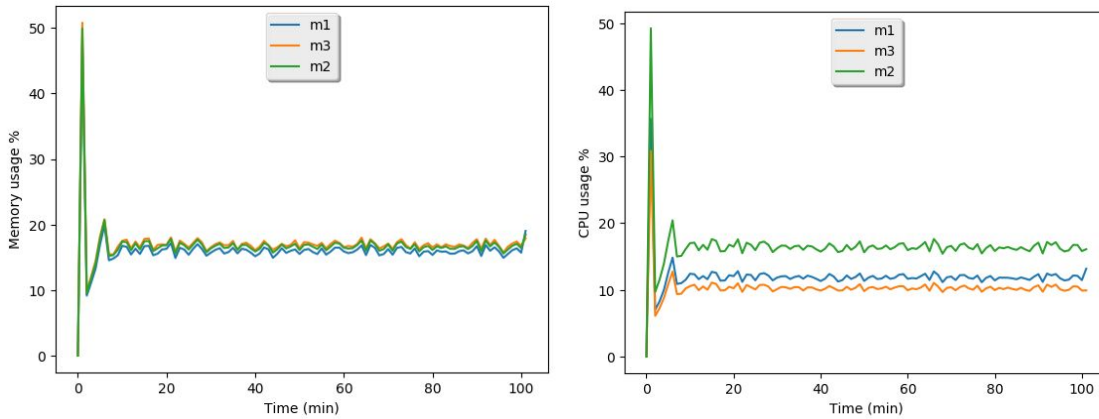


Imagen 11. Parte A a la izquierda presenta el uso de memoria en los FOG, parte B a la derecha el uso de CPU. Para una de las simulaciones del CASO I.

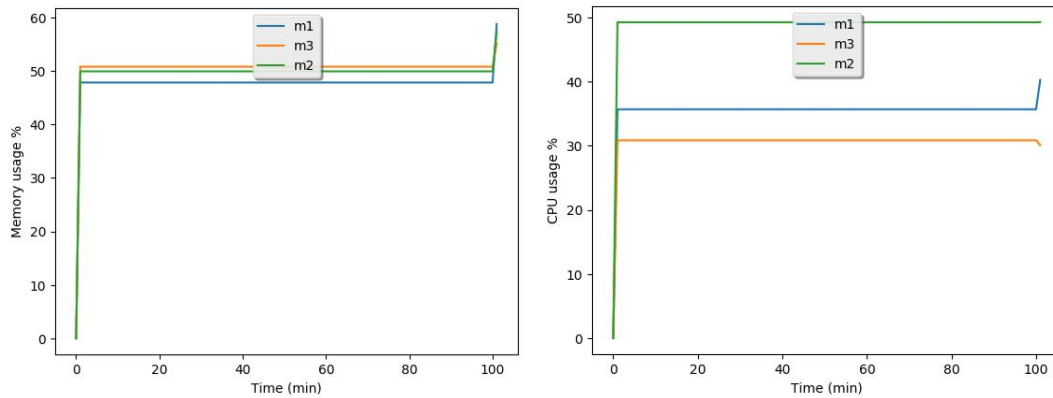


Imagen 12. Parte A a la izquierda presenta el uso de memoria en los FOG, parte B a la derecha el uso de CPU. Para una de las simulaciones del CASO I.

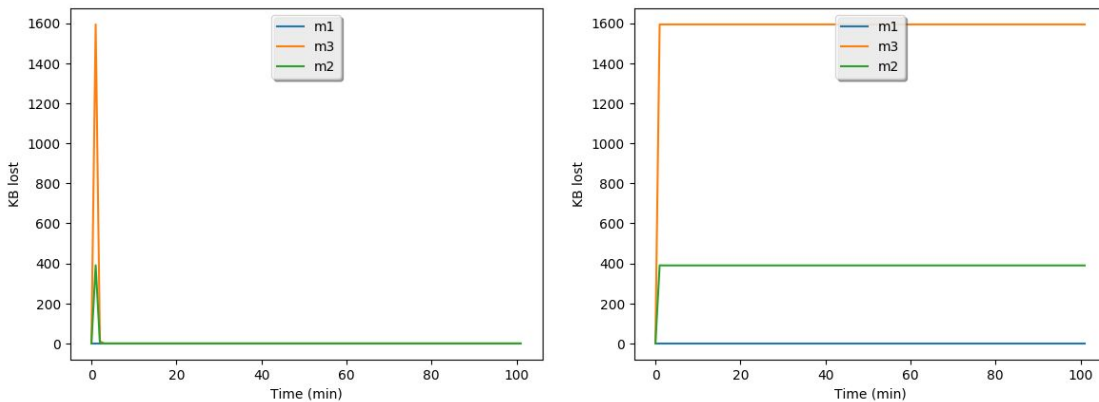


Imagen 13. KB perdidos en dos de las simulaciones para el caso del CASO I.

CASO II: Asignando $\frac{1}{3}$ del flujo de cada IoT a cada FOG se obtuvieron los siguientes resultados

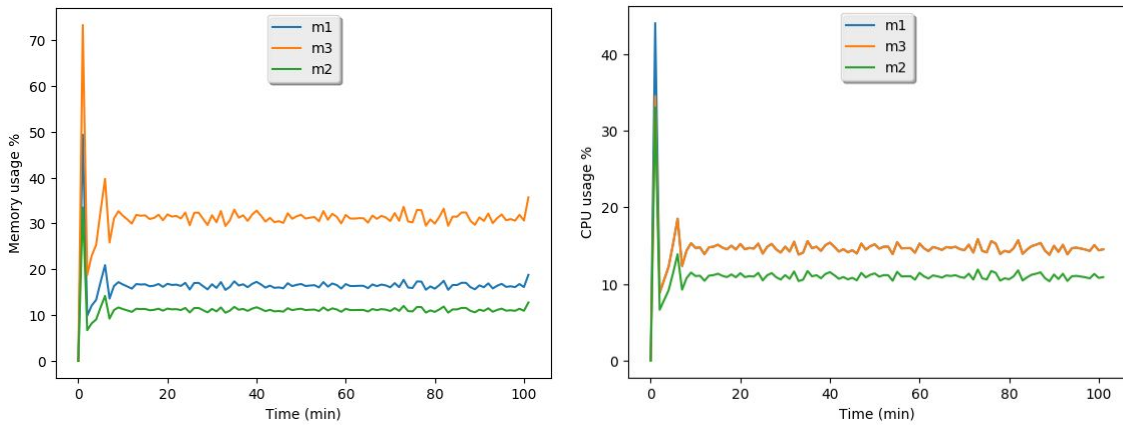


Imagen 14. Parte A a la izquierda presenta el uso de memoria en los FOG, parte B a la derecha el uso de CPU. Para una de las simulaciones del CASO II.

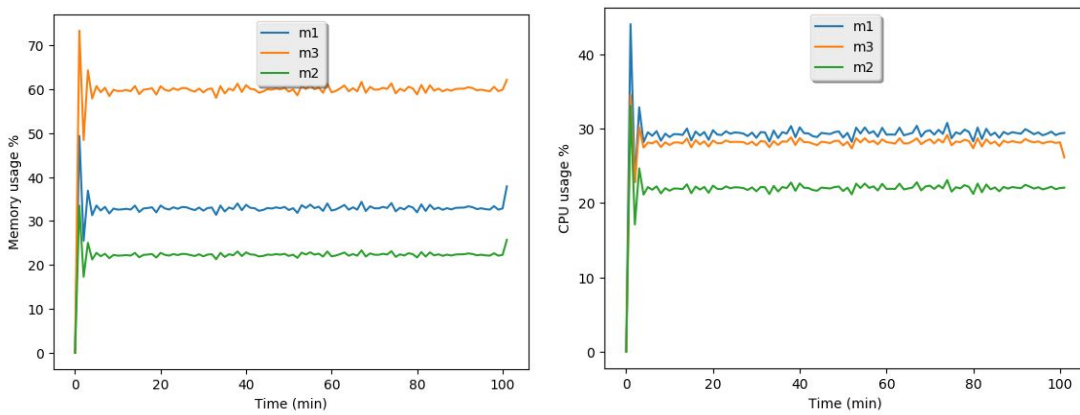


Imagen 15. Parte A a la izquierda presenta el uso de memoria en los FOG, parte B a la derecha el uso de CPU. Para una de las simulaciones del CASO II.

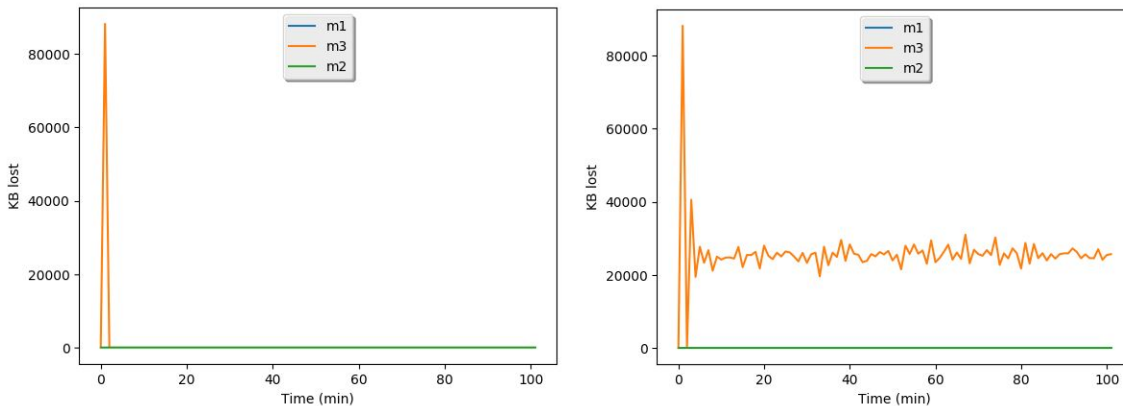


Imagen 16. KB perdidos en dos de las simulaciones para el caso del CASO II.

CASO III: Aleatorizando el porcentaje de cada flujo enviado desde un IoT a cada FOG se obtuvieron los siguientes resultados

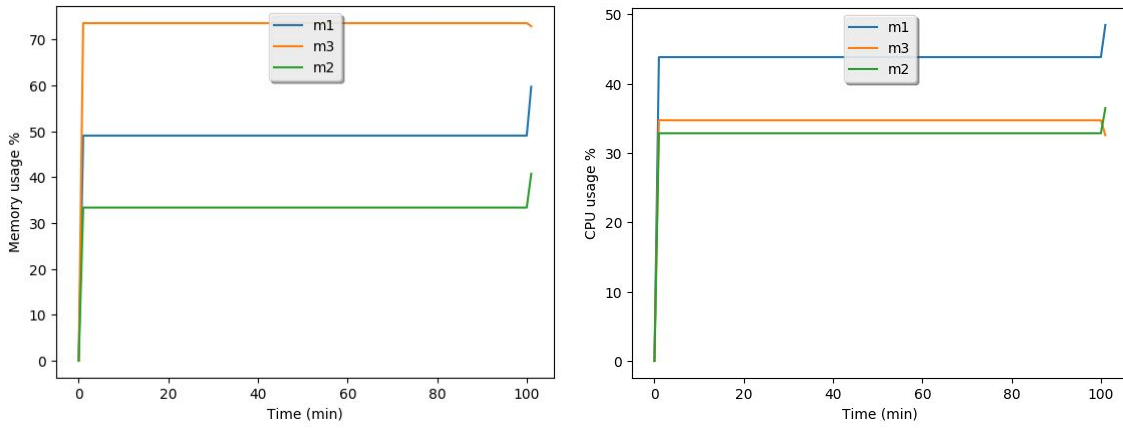


Imagen 17. Parte A a la izquierda presenta el uso de memoria en los FOG, parte B a la derecha el uso de CPU. Para una de las simulaciones del CASO III.

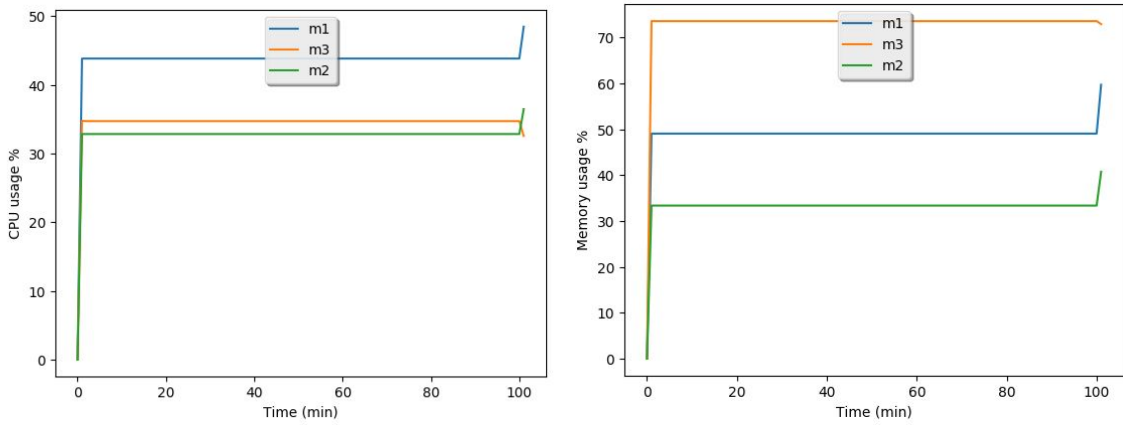


Imagen 18. Parte A a la izquierda presenta el uso de memoria en los FOG, parte B a la derecha el uso de CPU. Para una de las simulaciones del CASO III.

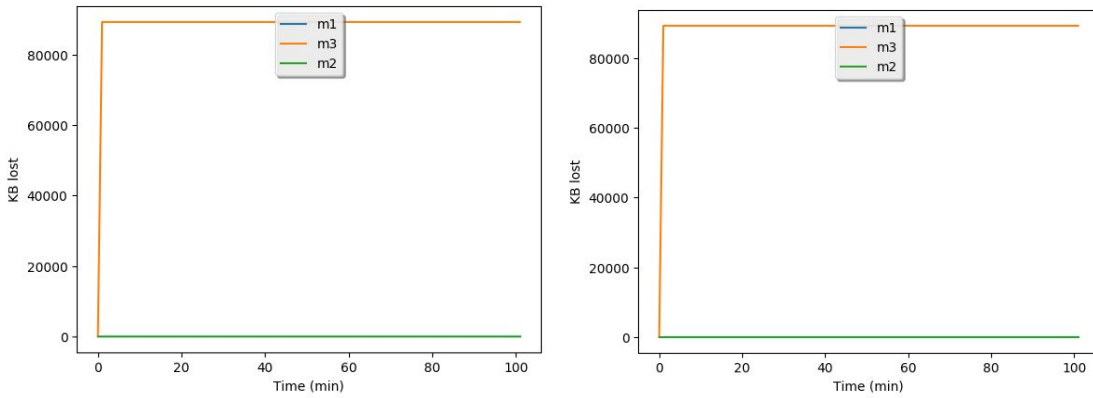


Imagen 19. KB perdidos en dos de las simulaciones para el CASO III .

La tabla 1 presenta el valor de la función objetivo para cada caso. La imagen 19 las pérdidas totales para cada simulación

Tabla 1. Valor de la función objetivo para cada caso.

Caso	α
CASO I	0.405
CASO II	0.266
CASO III	0.469

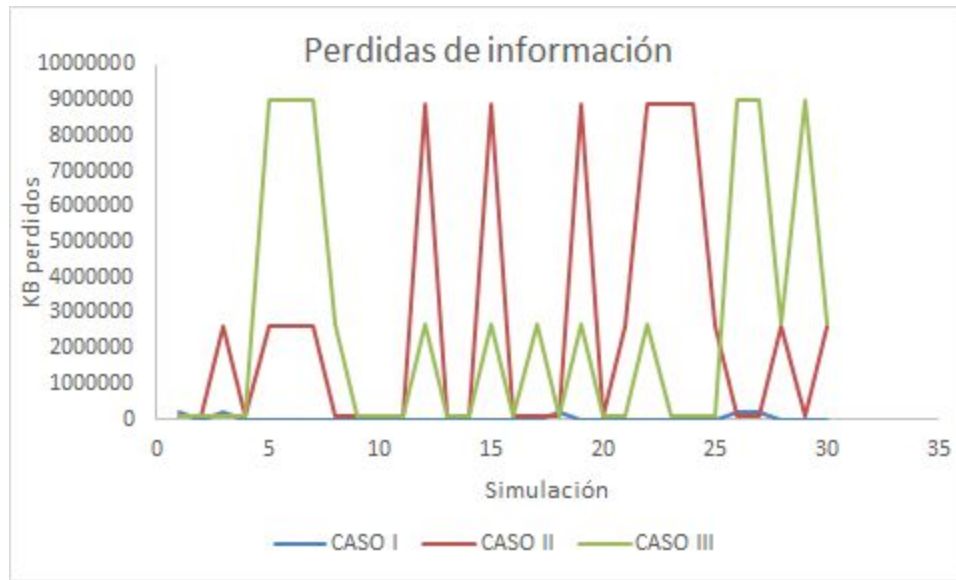


Imagen 19. KB perdidos en total para todas las simulaciones.

Teniendo como hipótesis nula H_0 y como hipótesis alterna H_1 se realizó una prueba t con una confianza del 95%. La tabla 2 presenta los resultados al comparar la media de pérdidas del caso optimizado con las otras dos casos. Se utilizó el software Minitab para obtener los resultados.

$$H_0 = \mu_{opti} - \mu_{muestra} = 0$$

$$H_1 = \mu_{opti} - \mu_{muestra} < 0$$

Tabla 2. Resultados de las pruebas t.

Caso	p-value	conclusión
CASO I vs CASO III	0	La optimización es mejor
CASO I vs CASO II	0	La optimización es mejor

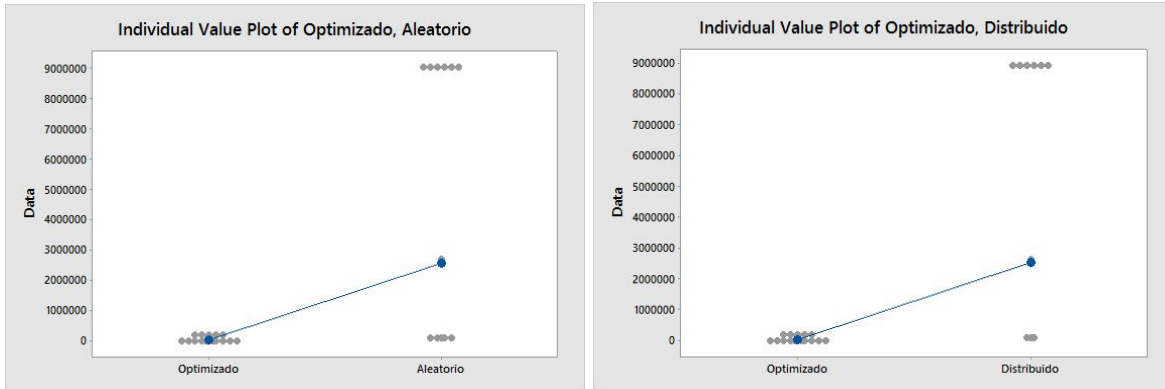


Imagen 20. Parte A a la izquierda gráfico individual para la comparación de medias del CASO I vs CASO III. Parte B a la derecha CASO I vs CASO II.

7 CONCLUSIONES

7.1 DISCUSIÓN

Este proyecto se dividió en tres etapas principales, la primera, fue el modelado matemático del problema. Esta parte del proyecto se desarrolló de forma iterativa: se diseñaba un modelo, se hacía una revisión con el asesor mirando aspectos correctos e incorrectos del modelo y se volvía a diseñar hasta que se obtuvo el modelo correcto. La segunda etapa fue la optimización. En esta fase del proyecto se implementó el modelo de la parte anterior en Gams, esta optimización se realizó con un caso base de 2 Fog y 5 dispositivos IoT, se fueron comprobando los resultados y se fue ajustando el modelo para conseguir los óptimos esperados. Al final de esta fase, se obtuvo un modelo que permite optimizar casos de cualquier tamaño. La última etapa fue la de validación que se realizó por medio de simulación. Para la simulación, se tuvo que desarrollar un simulador hecho a la medida dado que los simuladores encontrados como Opnet y ns3 no permiten ingresar los parámetros necesarios o llevar un registro de las medidas requeridas para comprobar el estado de la simulación. En cuanto a los resultados, el promedio del uso de los recursos en los FOG se estabiliza tras un tiempo. Adicionalmente existen pérdidas en todos los casos tras el minuto 1. Esto puede deberse a que paquetes pequeños que tardan menos tiempo en procesar son generados con más frecuencia que otros paquetes cuyo requerimientos de memoria y procesamiento son mucho mayores. A pesar de esto son máximo 1600 KB por minuto.

7.2 TRABAJO FUTURO

El presente trabajo logró demostrar que el direccionamiento de la información en al arquitectura teniendo en cuenta las restricciones de esta es una mejor alternativa a repartir equitativa o aleatoriamente el trabajo. Sin embargo, las suposiciones que se hicieron pueden afectar significativamente la respuesta obtenida por lo cual se propone hacer un modelo matemático más robusto para ser más fiel a la realidad. Adicionalmente, buscar simuladores capaces de soportar la especificación más robusta del problema. Finalmente utilizar una arquitectura real para comprobar el modelo matemático y la aplicación de este a la industria.

8 REFERENCIAS

Andersson, M. (2014). Short range low power wireless devices and Internet of Things (IoT). *Contributed By Digi-Key Corporation*, 01-16.

Arkian, H. R., Diyanat, A., & Pourkhalili, A. (2017). MIST: Fog-based data analytics scheme with cost-efficient resource provisioning for IoT crowdsensing applications. *Journal of Network and Computer Applications*, 82, 152–165. <https://doi.org/10.1016/j.jnca.2017.01.012>

Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog Computing and Its Role in the Internet of Things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing* (pp. 13–16). New York, NY, USA: ACM. <https://doi.org/10.1145/2342509.2342513>

Chatterjee, U. (2017). A pUF-Based secure communication protocol for IoT. *Acm Transactions on Embedded Computing Systems*, 16(3), 1-25. doi:10.1145/300571

Chen S, Zhang T, and Shi W. 2017. "Fog Computing." *Ieee Internet Computing* 21 (2): 4–6. doi:10.1109/MIC.2017.39.

Díaz, M., Martín, C., & Rubio, B. (2016). State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. *Journal of Network and Computer Applications*, 67, 99–117. <https://doi.org/10.1016/j.jnca.2016.01.010>

Donoso, Y. 2016. Capitulo 2 optimización. [Presentación de Power Point] Obtenido de https://sicuaplus.uniandes.edu.co/webapps/blackboard/execute/content/file?cmd=view&content_id=_1594427_1&course_id=_104600_1

Sun Y. and Zhang N. (2017). A resource-sharing model based on a repeated game in fog computing. *Saudi Journal of Biological Sciences*, 24(3), 687-694. doi:10.1016/j.sjbs.2017.01.043