Diseño de un Sistema Cloud Aplicado a e-Health

Jordi Vilaplana[1], Francesc Solsona[1], Francesc Abella[2], Jaume Celma[2]

Resumen— **La computación cloud es un nuevo paradigma que está cambiando la ubicación de las infraestructuras informáticas hacia Internet. Con ello se consigue que las organizaciones (empresas) no tengan que mantener ni sus propios servidores ni tampoco su software, ahorrando de este modo energía, espacio físico y personal técnico. Además, la nube ofrece un gran rendimiento en términos de escalabilidad, mantenibilidad y procesamiento masivo de datos en entornos dinámicos y de necesidades cambiantes.**

En este artículo se presenta una plataforma cloud aplicada a e-Health con el fin de optimizar los recursos utilizados. La escalabilidad es un factor a tener en cuenta puesto que la plataforma ha de garantizar unos niveles de QoS (Quality of Service) aceptables. Para su modelización, se ha diseñado un modelo basado en sistemas de colas que consiste en la unión de dos colas M/M/m de forma serie. Los resultados obtenidos son muy esperanzadores, aunque falta corroborar el modelo en una plataforma real.

Palabras clave—**computación cloud, e-Health, modelado del rendimiento, sistemas de colas, escalabilidad.**

I. Introducción

La computación cloud es un nuevo paradigma que está cambiando la ubicación de las infraestructuras informáticas hacia Internet. Con ello se consigue que las organizaciones (empresas) no tengan que gestionar ni sus propios servidores ni tampoco su software, ahorrando de este modo energía, espacio físico y personal técnico. Además, los sistemas cloud ofrecen un gran rendimiento en términos de escalabilidad, mantenibilidad y procesamiento masivo de datos en entornos dinámicos y de necesidades cambiantes [5]. La computación cloud está teniendo una adopción masiva [2]. Plataformas Xen [21] o VMWare [22] están ofreciendo entornos de computación virtuales que permiten gestionar y configurar un sistema cloud, sin embargo no ofrecen las facilidades de gestionar de forma dinámica (creación y eliminación) los recursos computacionales (básicamente las máquinas/servidores virtuales) según una calidad de servicio preestablecida.

La computación cloud ha atraído la atención de un considerable número de investigadores, pero sólo una pequeña porción del trabajo realizado hasta ahora ha abordado los problemas de rendimiento [3]. En [13], se obtuvo la distribución del tiempo de respuesta de un sistema cloud modelado según una red abierta clásica $M/M/m$, en el supuesto que los tiempos entre llegadas y de servicio seguían una función de densidad exponencial. Mediante el uso de la distribución del tiempo de respuesta, se determinó el nivel de servicio óptimo así como la relación entre el número máximo de tareas y el número mínimo de recurso (máquinas virtuales). El tiempo de respuesta se desglosa en el tiempo de espera y de servicio. En [14], se obtuvo la distribución del tiempo de respuesta para un sistema cloud para un modelo $M/m/m+r$. Tanto los tiempos entre llegadas y servicio se suponían con una distribución exponencial y el sistema tenía un número finito de buffers de tamaño $m+r$. El análisis del caso donde el tiempo entre llegadas y/o tiempo de servicio no son exponenciales es más compleja, como es el caso de los modelos G/M/m, M/G/m y G/G/m. Muchos análisis teóricos se han basado en una amplia investigación en la evaluación del rendimiento, entre ellos destacamos los que analizan el modelo M/G/m (p.e. [15]). La dificultad radica en que las distribuciones de probabilidad del tiempo de respuesta no puede obtenerse mediante una fórmula cerrada, y por lo tanto requiere la búsqueda de formulaciones aproximadas. Sin embargo, nuestro trabajo no se centra en la investigación de modelos de colas concretos sinó en la utilización de modelos ya existentes para modelar sistemas cloud para su aplicación en e-Health.

En este artículo se presenta una plataforma cloud aplicada a e-Health y con QoS (basado en el tiempo de espera por los servicios), con el fin de optimizar los recursos utilizados. En el Hospital Santa Maria de Lleida se ha probado con éxito una aplicación telemática para mejorar el tratamiento de la deshabituación tabáquica. Se ha demostrado que el uso de aplicaciones de seguimiento telemático mejoran los resultados en el tratamiento de los pacientes fumadores. También se ha demostrado que el mismo método es válido para ser aplicado en tratamientos de pacientes hipertensos [16][17] y en pacientes con enfermedades crónicas en general [18]. Esto no es mas que un ejemplo de las grandes posibilidades que ofrece la aplicación de computación cloud a e-Health. La plataforma de e-Health que presentamos contempla muchos centros hospitalarios con sus respectivas especialidades compuestas por personal sanitario así como los respectivos pacientes.

Para el diseño del sistema nos hemos basado en el modelaje del mismo mediante un sistema de colas con el fin de determinar y luego controlar el tiempo de espera de los usuarios. El modelado de un sistema de colas permite que dada una plataforma de virtualización, el sistema cloud pueda escalar automáticamente de una forma óptima respetando los márgenes de QoS (i.e. tiempo de espera), planificando adecuadamente el despliegue y reducción de máquinas

[1]Dept. d'Informàtica i Enginyeria Industrial, Univ. de Lleida, Jaume II 69, 25001 Lleida, e-mail: jordi,francesc@diei.udl.cat.

[2]Unitat de Tabaquisme de l'Hospital Santa Maria de Lleida, Alcalde Rovira Roure, 44, 25198 Lleida, e-mail: abella,jcelma@gss.scs.es.

virtuales según la carga del sistema [12]. Esto se consigue mediante la utilización del software de gestión de máquinas virtuales de código abierto OpenStack [6].

En la sección *Conceptos Previos* (Sec. II) se introduce el concepto de computación *cloud* en más detalle y se presentan los elementos que lo componen, tanto hardware como software. En la sección *Análisis y Diseño del Sistema* (Sec. III) se detallan los requerimientos de nuestra plataforma cloud y luego según el análisis realizado se da un diseño del sistema. En la sección de *Modelado* (Sec. IV) se propone un modelo del sistema e-Health compuesto por dos colas M/M/m conectadas en serie, así como un algoritmo de control del tiempo de espera medio en las colas (criterio de QoS escogido). En la sección Resultados (Sección V) se analizan los resultados que se obtiene en la modelización del sistema e-Health. Finalmente las Secciones VI y VII presentan las principales conclusiones y el trabajo futuro.

II. Conceptos Previos

Un sistema cloud es una red de servidores bajo demanda, escalable y flexible. Los servicios que pueden ofrecer los sistema cloud se agrupan en tres categorías. Infrastructure as a Service (IaaS), que ofrece hardware, almacenamiento y dispositivos físicos a través de Internet; Software as a Service (SaaS), ofrece software y aplicaciones alojadas en Internet; y la combinación de ambas, Platform as a Service (PaaS), que implica ofrecer tanto hardware como software [3]. Los servicios cloud ofrecen múltiples beneficios para las aplicaciones con una alta conectividad con todo tipo de dispositivos (véase Fig. 1): smartphones, tablets, PCs, netbooks, portátiles, etc.

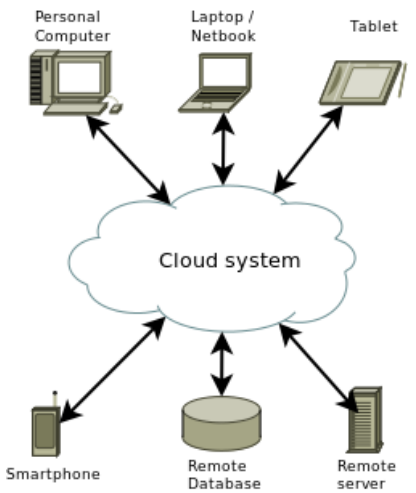

Fig. 1. Conectividad de una plataforma cloud.

Los sistemas de computación cloud de alta escalabilidad (línea de investigación del presente trabajo), ofrecen la ilusión de disponer de recursos computacionales infinitos bajo demanda, permitiendo la expansión de los recursos cuando es necesario. Los recursos se gestionan de un modo más eficiente que en otras infraestructuras HPC ya que éstos se pueden crear y eliminar de forma dinámica [4].

A. Arquitectura Cloud

La arquitectura de un sistema cloud se compone de dos partes: Front- i Back- end (véase Fig. 2).

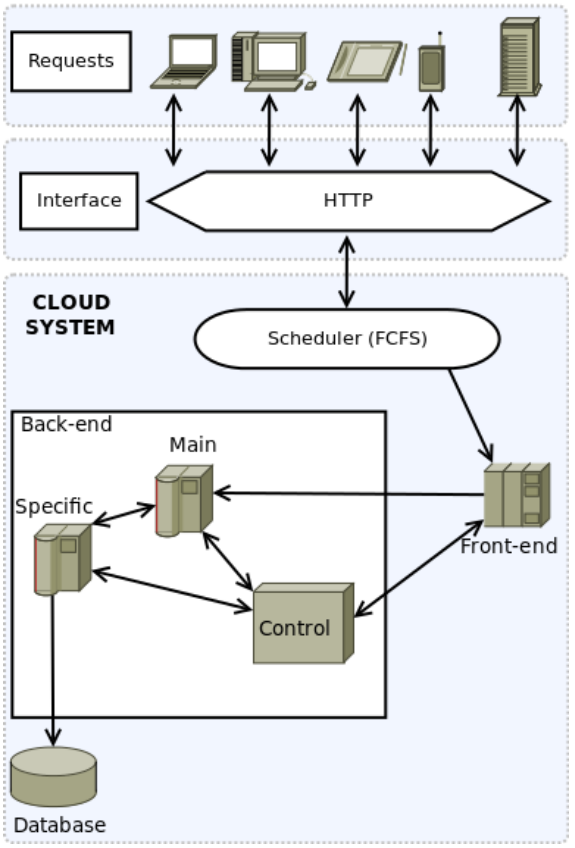

Fig. 2. Modelado del sistema cloud.

Front-end

El Front-end es la puerta de entrada al cloud y consiste en las aplicaciones e interfaces necesarios para poder conectarse mediante aplicaciones remotas autorizadas. Suelen disponer de protocolos web estándar para acceder al sistema. Todas las peticiones se procesarán en el planificador de tareas, que suponemos con una política FCFS, el cual envía las tareas a la cola del Back-end.

Back-end

Las funciones del Back-end incluyen la gestión de la cola de trabajos, los servidores y máquinas virtuales, los sistemas de almacenamiento y el sistema de gestión de la base de datos. Todas las peticiones provenientes del Front-end se gestionan mediante un planificador de trabajos para ser dispuestos en una cola. El sistema de servidores se compone de múltiples máquinas virtuales gestionadas mediante OpenStack y conectadas a un servidor de bases de datos.

El Back-end está compuesto de los siguientes tipos de servidores:

- Servidores principales: máquinas virtuales que ejecutan el núcleo de la aplicación. Son los encargados de ejecutar la mayor parte del cómputo.

- Servidores específicos: máquinas virtuales cuyo principal cometido es realizar cálculos específicos y encargarse del interface con el Front-end, así como de la comunicación con la base de datos y otros servidores (incluso con los servidores principales).
- Servidor de control: máquina virtual encargada de monitorizar el estado del sistema y de los demás servidores. Este servidor es el responsable de crear/eliminar máquinas virtuales de forma dinámica.

B. *OpenStack*

OpenStack[6] es un framework de virtualización de código abierto altamente interoperable y escalable. OpenStack va más allá de un hyper-visor clásico (i.e. VirtualBox[20], Xen[21], VMware[22]), el cual permite crear máquinas virtuales bajo demanda a medida que se requieren más recursos. Con ello se garantiza la calidad del servicio durante picos de tráfico periódicos y/o esporádicos, cuando el ratio de llegada de peticiones se incrementa notablemente. OpenStack puede configurarse para crear nuevas instancias cuando los nodos actuales se vean desbordados y para detenerlos cuando el tráfico vuelve a su ratio normal. Esta funcionalidad permite asegurar que el número de instancias activas en el cloud escale cuando el sistema crezca, y es especialmente útil para aplicaciones que experimentan variaciones significativas en su uso.

III. Análisis y Diseño del Sistema

En este trabajo estamos interesados en el diseño del Back-end, compuesto de los servidores principales, específicos y de control. Este diseño ha de tener en cuenta el análisis de requerimientos, que en nuestro caso solo se centrará en caracterizar los usuarios que puede dar servicio nuestra plataforma cloud.

Volviendo al caso práctico que nos ocupa, el sistema de e-Health ha de tener suficientes prestaciones como para poder dar servicio a muchos usuarios (principalmente personal sanitario y pacientes de varios hospitales), aunque no infinitos. Teniendo en cuenta la arquitectura cloud (explicada en la sección II-A), los servidores principales del Back-end serán los encargados de servir las peticiones de los usuarios de la plataforma.

Además, los datos se guardarán en un servidor de bases de datos. Se reservarán varios nodos específicos encargados de la comunicación con la base de datos.

Finalmente, el servidor de control será el encargado de controlar la creación/eliminación de servidores principales y específicos. Para controlar el sistema, se propone la utilización de un sistema de colas que modele el rendimiento del sistema. Este modelo se presenta en la siguiente sección (Modelado, Sección IV).

La Figura 2 muestra el diseño final del sistema cloud. En ella se puede apreciar como las peticiones de servicio, son planificadas por el "Scheduler" mediante una cola FCFS. A continuación, las peticiones se pasan al Front-end, encargado de enviar las tareas al Back-end. Dentro del Back-end se muestra la comunicación entre sus componentes.

IV. Modelado

En esta sección nos centramos únicamente en el Back-end, controlado por el servidor de control cuya función básica es crear/eliminar servidores principales y específicos. Estas decisiones las toma teniendo en cuenta el tiempo de espera de los trabajos, la cual se modela mediante un sistema basado en colas.

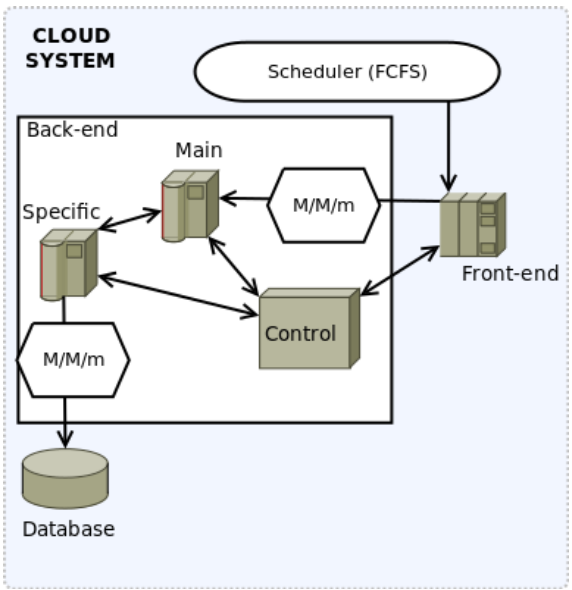

Fig. 3. Modelado

Como puede verse en la Figura 3, en el sistema van a haber dos colas iguales, que en una primera aproximación, se ha decidido que sean del tipo $M/M/m$. Esto indica que tanto el tiempo entre llegadas de peticiones como el tiempo de servicio de las dos colas se distribuye de forma exponencial con una media λ y μ respectivamente, y con m servidores que siguen una política de planificación del tipo FCFS. La primera cola modela los servidores principales y la segunda los servidores específicos que interaccionan con la base de datos.

Abstrayendo el problema, se puede considerar una modelización mediante un sistema de colas compuesto por dos colas $M/M/m$ en serie, tal y como se muestra en la Figura 4. En este sistema, las peticiones entran por la primera cola. Con una probabilidad d pasan a la segunda cola $M/M/m$, que representa el sistema de base de datos. Con una probabilidad $(1-d)$, la petición se irá del sistema sin pasar por la segunda cola. De esta forma estamos modelando un sistema en el que cada petición de servicio requerirá de un cómputo, más un acceso a la base de datos con una probabilidad d.

Según el teorema de Burke [1], la salida de una cola M/M/m estable con un parámetro de entrada λ, y μ como parámetro de servicio para cada uno de los m servidores, es otro proceso de Poisson con el

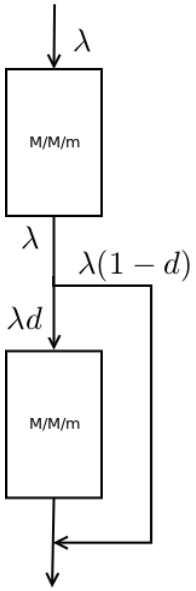

Fig. 4. 2 servidores M/M/m en serie.

mismo parámetro de entrada λ. Esto significa que la conexión en serie de dos sistemas M/M/m (sin ciclos) es independiente entre ellos y dichos sistemas preservan las mismas distribuciones de densidad, tanto de entrada como de servicio.

A. M/M/m

En este apartado estudiamos el sistema de colas M/M/m, con m servidores y dos funciones de densidad, que representan la media entre arribos (λ) y de servicio por servidor (μ), tal y como se muestra en la Fig. 5.

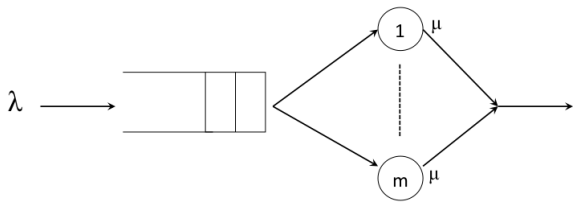

Fig. 5. Esquema cola M/M/m.

En la Fig. 6 se muestra el diagrama de transición de estados en condiciones de equilibrio del sistema, así como las ecuaciones que lo definen.

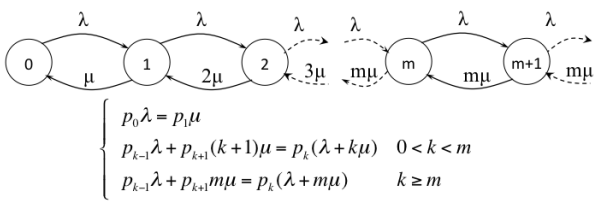

Fig. 6. Diagrama de transición de estados y ecuaciones en equilibrio de la cola M/M/m.

Resolviendo el sistema de ecuaciones se obtiene p_k (probabilidad de que en el sistema haya exactamente k usuarios):

$$p_k = \begin{cases} p_0 \frac{(mp)^k}{k!} & k \leq m \\ p_0 \frac{m^m \rho^k}{m!} & k \geq m \end{cases} \tag{1}$$

donde el factor de utilización (ρ) es:

$$\rho = \frac{\lambda}{m\mu} < 1 \tag{2}$$

Teniendo en cuenta que:

$$\sum_{k=0}^{\infty} p_k = 1, \tag{3}$$

obtenemos p_0 (probabilidad de que no haya ningún usuario en el sistema):

$$p_0 = \left[\sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} + \frac{(m\rho)^m}{m!(1-\rho)} \right]^{-1} \tag{4}$$

El número medio de usuarios en la cola de espera (N_W) es:

$$\begin{aligned} N_W &= \sum_{k=0}^{\infty} k p_{k+m} \\ &= \sum_{k=0}^{\infty} k p_0 \frac{m^m \rho^{k+m}}{m!} = \frac{p_0 (m\rho)^m}{m!} \sum_{k=0}^{\infty} k \rho^k \\ &= \frac{p_0 (m\rho)^m}{m!} \frac{\rho}{(1-\rho)^2} \end{aligned} \tag{5}$$

El tiempo medio de espera en la cola W (parámetro de QoS escogido en este trabajo) se define como:

$$W = \frac{N_W}{\lambda} \tag{6}$$

B. Calidad de Servicio

El criterio de calidad de servicio QoS que se quiere ofrecer es el tiempo de respuesta en la cola. El tiempo de espera depende del factor de utilización ρ. En un sistema de colas M/M/m, $\rho = \frac{\lambda}{m\mu}$.

Algorithm 1 Control de QoS

Ensure: $W_{max} = 750ms$
Ensure: $W_{min} = 150ms$
  while TRUE **do**
    if $W > W_{max}$ **then**
      while $W > W_{max}$ **do**
        $start - virtual - server$
      end while
    else
      if $W < W_{min}$ **then**
        while $W < W_{min}$ **do**
          $stop - virtual - server$
        end while
      end if
    end if
    $sleep(T)$
  end while

Según [19], un sistema se encuentra dentro de unos márgenes aceptables de interactividad cuando

el tiempo de espera medio (W) no supera $750ms$ ($W_{max} = 750ms$). También se considera que el usuario no aprecia la mejora de la interactividad cuando el sistema ofrece un tiempo de espera medio por debajo de los 150ms ($W_{min} = 150ms$). Por consiguiente, podemos establecer que si el tiempo de espera medio de nuestro sistema supera W_{max}, será necesario crear nuevas máquinas virtuales, es decir, aumentar el número de servidores principales o específicos (según el caso), hasta que W se sitúe por debajo del umbral W_{max}. Por el contrario, si W cae por debajo de W_{min}, se pueden liberar recursos, que en nuestro caso se corresponde a eliminar servidores principales (o específicos), hasta que W vuelva a superar el límite inferior W_{min}. Este proceso se muestra en el Algoritmo 1, el cual controla de forma continua e ininterrumpida la necesidad de crear y eliminar cada cierto tiempo T, servidores virtuales dependiendo del valor de W, el cual tiene que estar comprendido en el rango $W_{min} \leq W \leq W_{max}$. El tiempo T estará preestablecido por el administrador del sistema. Seria interesante obtener T en función de ρ. Del mismo modo, también seria de gran interés incorporar en el algoritmo controles para determinar el tipo de máquinas virtuales a crear/eliminar (principales o específicas). No obstante, lo reservamos para un trabajo futuro.

V. Resultados

A continuación se analiza como afecta al tiempo de respuesta el hecho de incrementar el número de servidores en una cola $M/M/m$. En la Fig. 7 vemos como aumenta el tiempo de espera de la primera cola incrementando el ratio de ocupación ρ para uno, dos, diez y cien servidores. Estos valores han sido obtenidos mediante el simulador de sistemas de colas Queue 2.0 [7].

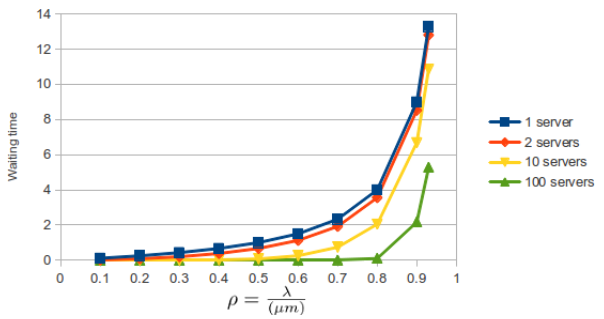

Fig. 7. Tiempo de espera en la cola 1 (M/M/m).

Para la segunda cola $M/M/m$, el ratio de entrada está determinado por λd. Hemos supuesto el valor $d = 0.9$ como la probabilidad de que una petición acceda a los servidores de bases de datos. La Fig. 8 muestra los mismos resultados que la Fig. 7, pero esta vez para la segunda cola. Para otros valores de d se obtienen resultados con proporciones similares a los obtenidos. Como cabía esperar, en la cola 2, el tiempo de espera es menor que en la cola 1. Se puede observar como el modelo propuesto es válido puesto que las gráficas respetan las mismas proporciones, aunque faltaría corroborarlo en un sistema real. Con un modelo simple, hemos demostrado que puede servir para modelar sistemas de computación cloud.

El resultado mas significativo es que para valores bajos del número de servidores, el segundo la relación entre los tiempos de espera no varia, manteniéndose en niveles constantes. Para grandes sistemas computacionales, con grandes requerimientos de unidades de procesamiento (servidores virtuales), aumentando el parámetro ρ, el ratio entre el tiempo de espera de la primera y segunda etapa, tiende a estabilizarse. Además esta relación disminuye de forma muy significativa en grandes sistemas (con gran número de máquinas virtuales). Este hecho se muestra en la Fig. 9.

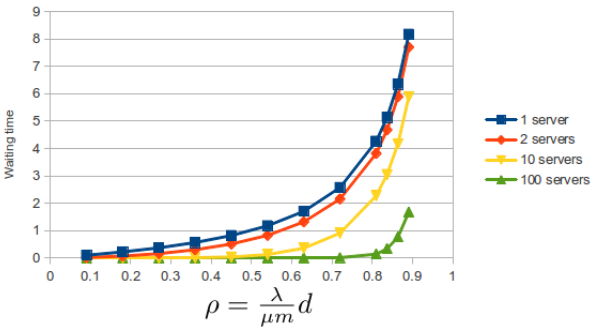

Fig. 8. Tiempo de espera en la cola 2 (M/M/m).

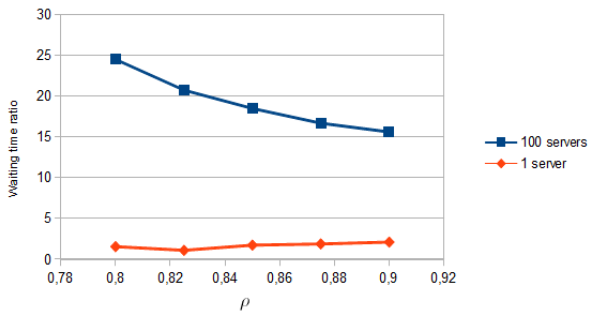

Fig. 9. Ratio del tiempo de espera entre la primera y segunda cola.

VI. Conclusiones

Este artículo ha proporcionado una visión del paradigma de computación cloud mediante el diseño de un sistema aplicable a e-Health. El diseño de un sistema cloud requiere centrarse en arquitecturas escalables (aunque centralizadas), flexibles, dinámicas y con un alto nivel de acoplamiento. En este trabajo, hemos escogido los sistemas de colas para modelar su rendimiento. De esta forma, el dinamismo del sistema basado en la creación/eliminación de servidores virtuales, se controla mediante el modelo de sistema de colas presentado.

Mediante un análisis preliminar, se ha propuesto un diseño para una arquitectura cloud adecuada a los requerimientos y necesidades del sistema e-Health. Para su modelización se ha propuesto la unión de dos sistemas M/M/m en serie, la primera para dar

servicios de cómputo y la segunda para dar servicio de accesos a un gestor de bases de datos. En su estudio se ha comprobado que para ofrecer una calidad de servicio en cuanto a tiempos de respuesta medios, dentro de unos márgenes aceptables, la relación del tiempo de espera del primer sistema respecto al segundo tiende a estabilizarse. La reducción es bastante más significativa cuanto mayor sea el número de máquinas virtuales.

VII. Trabajo futuro

Tal y como ya se ha mencionado anteriormente, se pretende ampliar el Algoritmo 1 para determinar el valor de T en función de ρ. Del mismo modo, también seria de gran interés incorporar mecanismos para decidir el tipo de máquinas virtuales a crear/eliminar (principales o específicas). También se pretende reemplazar las dos colas por un modelo más realista $M/M/m/m+r/K$, con m servidores, un número máximo de conexiones de usuarios $m+r$ (número máximo de usuarios en el sistema, es decir, usuarios recibiendo servicio, que como máximo es m, mas los que están en espera, cuyo máximo es r), y un número máximo de K usuarios, tal y como se presenta en [3].

Se tiene previsto desarrollar una aplicación mediante OpenStack que emule los requerimientos de la unidad de tabaquismo del Hospital Santa Maria de Lleida, con datos reales sobre la cantidad y requerimientos de los usuarios de la misma. Con ello se podrá hacer una estimación de los recursos computacionales que necesitaría dicha unidad y por extensión (sabiendo los usuarios del hospital) se pueden hacer diseños de sistemas cloud aplicados a e-Health para un hospital concreto. Extrapolando, se puede ampliar la aplicación para emular el comportamiento del sistema suponiendo un número variable de hospitales.

Los diferentes modelos de colas propuestos serán validados con la aplicación desarrollada con OpenStack.

Agradecimientos

Este trabajo está financiado por el Ministerio de Ciencia e Innovación mediante el contrato TIN2011-28689-C02-02. Los autores son miembros del grupo de investigación consolidado de la Generalitat de Cataluña 2009 SGR145.

Referencias

[1] P. Burke. *The Output of a Queuing System.* Operations Research, 4, 699-704. 2010.

[2] D. Smith et al. *Hype Cycle for Cloud Computing.* Gartner RAS Core Research Note G00201557. 2010.

[3] H. Khazaei, J. Misic, V. Misic. *Performance Analysis of Cloud Computing Centers Using M/G/m/m+r.Queuing Systems.* IEEE Transactions on parallel and distributed systems, vol. 23, n. 5. 2012.

[4] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia. *Above the Clouds: A Berkeley View of Cloud Computing.* Technical Report No. UCB/EECS-2009-28. 2009.

[5] J. Varia. *Architection for the Cloud: Best Practices.* Amazon Web Services. 2010.

[6] OpenStack Website. *OpenStack: An Overview. Available: http://openstack.org/.* Acceso 1 de Junio 2012.

[7] Queue 2.0 Website. *http://www.win.tue.nl/cow/Q2/.* Acceso 1 de Junio 2012.

[8] F. Tzelepis, C. Paul, J. Wiggers, R. Walsh, J. Knight, S. Duncan, C. Lecathelinais, A. Girgis, J. Daly. *A randomised controlled trial of proactive telephone counselling on cold-called smokers' cessation rates.* Tob Control 20: 40-46. 2011.

[9] C. Free, R. Whittaker, R. Knight, T. Abramsky, A. Rodgers and I. Roberts. *Txt2stop: a pilot randomised controlled trial of mobile phone-based smoking cessation support.* Tob Control 18, pp. 88-91. 2009.

[10] C. Free, R. Knight, S. Robertson, R. Whittaker, P. Edwards, W. Zhou, A. Rodgers, J. Cairns, M. Kenward, I. Roberts. *Smoking cessation support delivered via mobile phone text messaging (txt2stop): a single-blind, randomised trial.* The Lancet 378-9785: pp. 49-55. 2011.

[11] T. Sai Sowjanya, D.Praveen, K.Satish, A.Rahiman. *The Queueing Theory in Cloud Computing to Reduce the Waiting Time.* IJCSET, Vol 1, Issue 3,110-112. 2011.

[12] M. Mao, J. Li, M. Humphrey. *Cloud Auto-scaling with Deadline and Budget Constraints.* Proceedings of GRID. pp 41-48. 2010.

[13] K. Xiong, H. Perros. *Service Performance and Analysis in Cloud Computing.* Proc. IEEE World Conf. Services, pp. 693-700. 2009.

[14] B. Yang, F. Tan, Y. Dai, S. Guo. *Performance Evaluation of Cloud Service Considering Fault Recovery.* Proc. First Int'l Conf. Cloud Computing (CloudCom'09), pp. 571-576. 2009.

[15] N. Ma, J. Mark. *Approximation of the Mean Queue Length of an M/G/c Queueing System.* Operations Research, vol. 43, pp. 158-165. 1998.

[16] T. Ohkubo, et al. *Home blood pressure measurement has a stronger predictive power for mortality than does screening blood pressure measurement: a population-based observation in Ohasama.* Japan. J Hypertens 16: 971-975.. 1998.

[17] G. Bobrie, G. Chatellier, N. Genes, et al. *Cardiovascular prognosis of "masked hypertension" detected by blood pressure self-measurement in elderly treated hypertensive patients.* JAMA 291: 1342-1349. 2004.

[18] A. León, et al. *A New Multidisciplinary Home Care Telemedicine System to Monitor Stable Chronic Human Immunodeficiency Virus-Infected Patients: A Randomized Study.* PLoS ONE 6(1): e14515. doi:10.1371/journal.pone.0014515. 2011.

[19] F. Nah. *A study on tolerable waiting time: how long are Web users willing to wait?* Behaviour & Information Technology, forthcoming. 2004.

[20] VirtualBox Website. *http://www.virtualbox.org/.* Acceso 1 Junio 2012.

[21] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. *Xen and the art of virtualization.* SIGOPS Oper. Syst. Rev., vol. 37, no. 5, pp. 164-177. 2003.

[22] WMWare Staff. *Virtualization overview.* White Paper, http://www.vmware.com/pdf/virtualization.pdf. Acceso 1 Junio 2012.