Revista Cubana de Ciencias Informáticas
ISSN: 2227-1899 | RNPS: 2301
Vol. 8, No. 4, Octubre-Diciembre, 2014
Págs. 157-170
http://rcci.uci.cu

Tipo de artículo: Artículo original
Temática: Desarrollo de aplicaciones informáticas
Recibido: 21/03/2014 | Aceptado: 8/10/2014

Sistema de distribución del proceso de búsqueda en bancos de datos de huellas dactilares

Distribution system of the search process in fingerprints databases

Royli Hernández Delgado [1*], Melvis Machin Armas [1],

[1] Centro de Identificación y Seguridad Digital. Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños, km 2 ½, Torrens, Boyeros, La Habana, Cuba. CP.: 19370.

[*] Autor para correspondencia: royli@uci.cu

Resumen

La búsqueda en el banco de datos de un sistema de identificación de huellas dactilares o AFIS, influye significativamente en la obtención de tiempos de respuesta adecuados y en la efectividad del sistema. Entre las estrategias de búsqueda están la centralización en un nodo que procese toda la información y la distribución del proceso en nodos que busquen paralelamente. En el Centro de Identificación y Seguridad Digital se desarrolló un AFIS para ser incluido en sus sistemas de control de acceso. Este sistema reporta una baja tolerancia a fallos y tiempos de respuesta ineficientes utilizando un banco de datos de solo 10 000 huellas, debido a la implementación de un proceso de búsqueda centralizado. Este trabajo propone erradicar estos inconvenientes con un sistema que brinde un servicio de distribución, realizando una búsqueda concurrente en varios nodos. Para ello se seleccionó el *middleware RabbitMQ* con el objetivo de establecer la comunicación entre los distintos nodos que participan en el proceso de búsqueda. La solución aporta una estrategia de monitoreo y control de fallos basada en la redistribución del banco de datos siempre que se detecte la caída de un nodo. Además, los resultados de las pruebas de rendimiento realizadas utilizando un banco de datos de 28 000 huellas dactilares, demostraron que al aumentar los datos a procesar, el modelo centralizado ralentizaba la identificación, mientras que con la utilización del servicio de distribución utilizando solo 3 servidores de búsqueda, los tiempos de respuesta del AFIS se redujeron en más de un 50%.

Palabras clave: AFIS, distribución, proceso de búsqueda, RabbitMQ.

Revista Cubana de Ciencias Informáticas
ISSN: 2227-1899 | RNPS: 2301
Vol. 8, No. 4, Octubre-Diciembre, 2014
Págs. 157-170
http://rcci.uci.cu

Abstract

The search process in a database system based on fingerprint identification (AFIS), has a significant influence in obtaining appropriate response times and system effectiveness. There are many search strategies such as centralization on a node to process all information and distribution process in parallel nodes participating in the search. In Identification and Security Digital Center an AFIS was developed to be included on their access control systems. This system reports low fault tolerance and time inefficient responses using a database of 10 000 fingerprints, due to the implementation of a centralized search process. To eliminate these problems with a system that provides a delivery service, performing a search on multiple nodes concurrently. RabbitMQ middleware was selected in order to establish communication between the different nodes participating in the search process. The solution provides a strategy for failure monitoring based on the redistribution of the database whenever a falling node is detected. Also, the results of performance testing using a database of 28 000 fingerprints showed that increasing the data processing, the centralized model identification slowed, while the use of the distribution service using only 3 search servers response times AFIS decreased by more than 50%.

Keywords: *AFIS, distribution, RabbitMQ, search process.*

Introducción

La seguridad se ha convertido en un asunto de actualidad que es motivo de preocupación en la protección de los bienes de cualquier entidad, de manera que la identificación de los individuos se convierte en factor fundamental. Para garantizar la identificación de un individuo existen diferentes métodos. Entre los más convencionales se encuentra el uso de tarjetas de identificación tales como el Documento Nacional de Identificación (DNI) o credenciales especiales creadas para un ambiente específico. Sin embargo, con el incremento de la potencia de cómputo en los últimos años y las ciencias informáticas en general, el uso de otros métodos como la biometría han sido aplicados a sistemas modernos que permiten identificar unívocamente a un individuo (Komarinski, 2005). Entre ellos la aplicación incremental y desarrollo vertiginoso de los AFISs han permitido utilizar las huellas dactilares como una de las características humanas más fiables utilizadas para la identificación (Muñoz, 2009) .

Uno de los componentes fundamentales de los AFISs es el algoritmo de comparación, encargado de comparar dos huellas y determinar el grado de similitud entre ellas. Para la identificación exitosa de un individuo resulta esencial un proceso de búsqueda que obtenga la similitud arrojada por este algoritmo entre la huella capturada por el sistema y cada una de las huellas almacenadas en el banco de datos. La robustez de este proceso influye significativamente para obtener tiempos de respuestas adecuados y en la efectividad del sistema, aún cuando pudiera ocurrir un crecimiento futuro del banco de datos que este maneja.

Debido al gran tamaño que suelen alcanzar los bancos de datos de los AFIS es necesaria la reducción del espacio de búsqueda. Por ejemplo una de las formas más conocidas de lograrlo es mediante la indexación (Maltoni, 2009). Por el contrario en muchos casos no se cuenta con algoritmos de indexación o con otras técnicas que agilicen el proceso de búsqueda. Entonces es necesario recurrir a un acercamiento más rudimentario y simple a través de la comparación de la huella introducida en el sistema con cada una de las huellas almacenadas en el banco de datos. Aunque resulte impracticable esta opción, la elección de una adecuada estrategia de búsqueda puede influir en la disminución de los tiempos del proceso.

La estrategia de búsqueda utilizada está estrechamente vinculada a la eficiencia del proceso en general, dentro de la cual se incluyen factores importantes como la arquitectura del sistema y las características del hardware, en cuanto a velocidad de procesamiento de CPU y memoria RAM se refiere. Aumentar la potencia de hardware de las unidades de cómputo u optar por una arquitectura distribuida con varios nodos de búsqueda, podrían ser alternativas a tener en cuenta cuando se plantea una estrategia. Sin embargo, cada una de estas propuestas tiene inconvenientes muy serios.

La centralización del proceso de búsqueda en un nodo con altas prestaciones de hardware permite fortalecer la seguridad y en la práctica suele ser inevitable (Rob, 2004). Este nodo procesa toda la información de forma lineal afectando directamente al tiempo de búsqueda, y su fallo podría provocar un colapso del sistema. Por otra parte, utilizar un modelo distribuido aportaría mayor flexibilidad en cuanto a la cantidad de unidades participantes en la búsqueda, de modo que el proceso ocurriría concurrentemente en todos los nodos disminuyendo teóricamente el tiempo de búsqueda. El fallo de un nodo no determina el fallo de la operación en general, ya que los demás podría completar el proceso de búsqueda. Sin embargo, este modelo es muy difícil de implementar (Rivero, 2004).

En el departamento de Biometría del Centro de Identificación y Seguridad Digital (CISED) de la Universidad se las Ciencias Informáticas (UCI) desarrolló un AFIS cuya estrategia de búsqueda de huellas dactilares en el banco de datos se basa en un modelo centralizado. Su tiempo de respuesta es de aproximadamente un segundo en identificar a un individuo utilizando un banco de datos de 10 000 huellas. Este tiempo de respuesta resulta ineficiente teniendo en cuenta que el número de huellas almacenadas no es significativo comparado con otros bancos de datos cuyas cifras se mueven en el orden de los millones. El problema se agudiza si se tiene en cuenta que el aumento futuro de huellas registradas influye directamente en el incremento de este tiempo, demorando la identificación, y por consiguiente, la pérdida de calidad del AFIS como producto.

Se podría pensar en la reutilización de componentes destinados al proceso de búsqueda en los AFIS más conocidos y con mejores resultados. Sin embargo, estos sistemas han sido desarrollados bajo licencias privativas y de ellos sólo se conoce la información referente a sus principios de funcionamiento y estrategias de búsqueda.

Revista Cubana de Ciencias Informáticas
ISSN: 2227-1899 | RNPS: 2301
Vol. 8, No. 4, Octubre-Diciembre, 2014
Págs. 157-170
http://rcci.uci.cu

Este trabajo se centra en el desarrollo de un sistema que permita distribuir el proceso de búsqueda de huellas dactilares en el banco de datos de un AFIS. Su objetivo es obtener mejores tiempos de respuesta en el proceso de identificación, teniendo en cuenta la escalabilidad del sistema y el crecimiento futuro del banco de datos.

Materiales y métodos

En la actualidad los AFISs y los sistemas de información en general que necesitan manejar grandes cantidades de datos en tiempos cada vez menores, así como compartir recursos para llevar a cabo tareas cada vez más complejas, utilizan el procesamiento distribuido. Este tipo de enfoque se refiere a varias computadoras autónomas conectadas mediante una red de comunicaciones y equipadas con programas que les permitan coordinar sus actividades y compartir recursos (Coulouris, 2001). Por ejemplo *ExpressID AFIS* de la empresa francesa *Innovatrics* es un sistema distribuido que presenta una arquitectura escalable, permitiendo a los componentes del servidor de comparación ser instalados en varios ordenadores para distribuir la carga de trabajo e incrementar la velocidad de comparación, siendo capaz de realizar búsquedas en bancos de datos de millones de huellas dactilares (Innovatrics, 2013).

Cuba también ha adoptado este concepto de distribución en la solución *BIOMESYS AFIS* comercializada por *DATYS, Tecnologías y Sistemas*. Se basa principalmente en un clúster de búsqueda y comparación. Este clúster tiene una infraestructura en la que se incluyen un servidor que distribuye la búsqueda (master), un servidor de contingencia (spare) y n servidores de comparación. La solución a las solicitudes de búsqueda se realiza de forma paralela íntegramente en memoria RAM en cada servidor de comparación, en los cuales está segmentado el banco de datos.

El aporte de la presente solución se basa en realizar esta distribución pero utilizando para ello las potencialidades de los middlewares distribuidores de mensajes. Entre los principales softwares de este tipo en el mercado destacan RabbitMQ, ApacheQpid y OpenAMQ. Son todos proyectos de código libre que implementan el protocolo AMQP[1] y cuyos resultados son satisfactorios para lograr un entorno distribuido. OpenAMQ posee una velocidad de 600 mil msj/seg[2] en un corredor sostenido por un día de trabajo completo y 300 ms de latencia, cifras adecuadas para la presente solución. Sin embargo, OpenAMQ tiene el inconveniente de no ser completamente fiel al estándar, pues deshecha parte de sus funcionalidades.

1 *Advanced Message Queue Protocol* es un protocolo de estándar abierto en la capa de aplicaciones de un sistema de comunicación a partir del cual se puede lograr interoperabilidad entre *middlewares* basados en mensajes. Este protocolo está caracterizado por el manejo de mensajes y colas, el enrutamiento (puede ser punto a punto o publicación-subscripción), la exactitud y la seguridad.

2 Msj/seg: mensajes que son transmitidos por segundo.

Revista Cubana de Ciencias Informáticas
ISSN: 2227-1899 | RNPS: 2301
Vol. 8, No. 4, Octubre-Diciembre, 2014
Págs. 157-170
http://rcci.uci.cu

ApacheQpid junto con RabbitMQ son los únicos servidores que implementan en su totalidad el protocolo. Ambos son líderes en su aplicación y han sido utilizados en proyectos de gran envergadura como OpenStack (Bryant, 2012) y el corredor Qpid C++ de Red Hat MRG (Foundation) con excelentes índices de distribución. Sin embargo, al emparejarlos se determinó que las potencialidades de OTP Erlang para la ejecución distribuida y conmutación ante errores presentes en RabbitMQ, aporta un mayor rendimiento al sistema a desarrollar. Otra de las razones por las que *RabbitMQ* se presentó como mejor candidato es la posibilidad que brinda de crear clústeres, característica que permite alcanzar una alta disponibilidad en las comunicaciones.

En cuanto a la administración de los distintos elementos del corredor, con *RabbitMQ* es mucho más sencilla que con *OpenAMQ* y *ApacheQpid*, pues cuenta con una interfaz web para el control de los mismos.

Resultados y discusión

La solución de esta investigación es un sistema que permite involucrar varios nodos a la búsqueda subdividiéndola, de forma tal que se realiza más rápidamente considerando el futuro incremento del banco de datos. El enfoque de la distribución aporta además de rapidez una mayor tolerancia ante los fallos (Coulouris, 2001), ya que la caída de un nodo de búsqueda no supone el colapso del AFIS, sino solo una nueva distribución entre los nodos disponibles en el momento. El sistema está conformado por dos módulos para lograr la distribución del proceso de búsqueda.

En el primer módulo dedicado a la distribución, se chequea periódicamente la disponibilidad de los distintos servidores de búsqueda. Es responsable de distribuir el banco de datos entre los servidores de búsqueda que han sido seleccionados para trabajar, balanceando la carga de forma tal que se maximice la utilización de los recursos y se obtenga el mejor desempeño del sistema. También atiende las solicitudes de identificación, delegando la búsqueda en los servidores destinados a esta tarea, para posteriormente procesar sus respuestas. Monitorea el estado de los servidores de búsqueda, así como genera *logs* sobre las solicitudes atendidas.

El segundo módulo lo constituyen los servidores de búsqueda, los cuales buscan en su porción del banco de datos el individuo al que corresponde la huella dactilar cuya comparación se requiere y responden en consecuencia al resultado obtenido.

El diseño del sistema está orientado a la extensibilidad y adaptabilidad (Cointe, 1999) con el objetivo de que pueda ser utilizado no solo por sistemas basados en el reconocimiento de huellas dactilares, sino también por otros sistemas biométricos. Para ello se definieron conceptos que propician dentro del negocio la abstracción y la extensibilidad, como por ejemplo: *Tipo de servidor de búsqueda* y *banco de datos*, debido a que por cada sistema biométrico existirá un tipo de servidor de búsqueda y un banco de datos al que consultar. La lógica de los servidores de búsqueda es definida en

dependencia de su tipo, es decir, el sistema biométrico que en el futuro utilice el servicio de distribución deberá implementar las peculiaridades de sus servidores de búsqueda, como está solución está dirigida a solucionar los inconvenientes detectados en el AFIS del CISED, el sistema desarrollado estructura, hasta el momento, el comportamiento de los servidores de búsqueda del mismo.

Arquitectura

Para el desarrollo del sistema se utilizó un estilo arquitectónico basado en la configuración cliente-servidor. En consecuencia con este modelo el sistema soporta el procesamiento distribuido, permitiendo la implementación de mecanismos eficientes de tolerancia a fallos (Kendall, 2005). La arquitectura propuesta (Ver Figura 1) garantiza en general expandir al AFIS para mejorar su rendimiento en forma dinámica, aumentando el tamaño del banco de datos y su carga de trabajo con el mínimo aumento de componentes, minimizando así el costo de las expansiones a largo plazo. Para la implementación de la arquitectura cliente-servidor se utilizó como patrón arquitectónico n-capas, específicamente 4 capas:

- Capa de Dominio: Contiene la interfaz de administración, así como toda la lógica del sistema, desde la atención de las solicitudes de distribución e identificación hasta la gestión de las entidades del sistema.
- Capa de Procesamiento: Contiene la lógica de los servidores de búsqueda, que incluye consultas al banco de datos biométrico y la búsqueda de la huella dactilar de un individuo que se desee identificar.
- Capa de Persistencia de datos: Está conformada por el banco de datos biométrico al que los servidores de búsqueda consultarán siempre que se requiera distribuirlo, así como por el banco de datos del sistema que almacenará toda la información persistente.
- Capa de Comunicación: Contiene un servidor distribuidor de mensajes encargado de distribuir mensajes de forma segura y eficiente a través del *middleware RabbitMQ*. Es la capa intermediaria en la comunicación entre: Publicador de solicitudes y distribuidor, Distribuidor y servidores de búsqueda.

Revista Cubana de Ciencias Informáticas
ISSN: 2227-1899 | RNPS: 2301
Vol. 8, No. 4, Octubre-Diciembre, 2014
Págs. 157-170
http://rcci.uci.cu

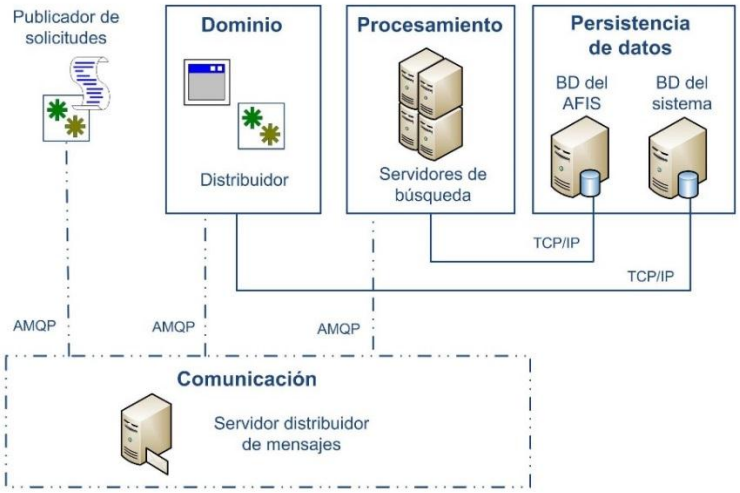

Figura 1 Arquitectura del sistema de distribución.

Entre las restricciones de la arquitectura se encuentra que el crecimiento del banco de datos está limitado por las condiciones de almacenamiento existentes en los distintos servidores de búsqueda y por la cantidad de solicitudes que se requiera alcanzar.

Método para el balanceo de carga

El balanceo de carga juega un papel muy importante en el sistema desarrollado, ya que es muy difícil predecir el tamaño de la porción del banco de datos que debe asignársele a cada servidor de búsqueda, de modo que la carga computacional se mantenga uniforme y se consiga la máxima velocidad de ejecución posible (Dormido, 2003). El método utilizado para realizar la asignación tiene en cuenta que tan apto es un servidor para ejecutar la búsqueda, tomando en consideración dos propiedades fundamentales:

- Memoria RAM disponible: Una vez asignada la porción del banco de datos cada servidor de búsqueda debe cargarla en su memoria RAM, por lo que debe tenerse la precaución de no colapsar el servidor con una asignación que no esté en correspondencia con su disponibilidad. Dado el caso de que un servidor no pueda asumir la porción de datos asignada, restricción de la arquitectura antes mencionada, el distribuidor manejará la excepción informando al usuario sobre las características del evento ocurrido. Es responsabilidad del usuario, teniendo en cuenta la información proporcionada, tomar las acciones convenientes para poder realizar la distribución exitosa del banco de datos.

Revista Cubana de Ciencias Informáticas
ISSN: 2227-1899 | RNPS: 2301
Vol. 8, No. 4, Octubre-Diciembre, 2014
Págs. 157-170
http://rcci.uci.cu

- Velocidad de procesamiento de CPU: Cada servidor buscará en la porción del banco de datos que se le haya asignado cuando se le solicite realizar una identificación, por lo que la velocidad del CPU influye en qué tan rápido se ejecute este proceso.

La función de aptitud está definida como la rapidez asociada a propiedades antes mencionadas (Gutiérrez, 2004). El cálculo se basa en la sumatoria entre estas propiedades, donde cada una de ellas está multiplicada por un peso que dependerá de su importancia para el desempeño del sistema: $Fa_i = a * dRAM_i + b * vCPU_i$

Donde:

$dRAM_i$: Memoria RAM disponible del servidor de búsqueda i.

$vCPU_i$: Velocidad de procesamiento del CPU del servidor de búsqueda i.

a y b: Pesos asignados a la memoria RAM disponible y a la velocidad de procesamiento de CPU respectivamente en una escala de [0; 1] (a = 0,6 y b = 0,4).

Para realizar el cálculo de la función de aptitud es necesario llevar las variables a una escala común, ya que si se trabaja con los valores originales la función de aptitud se volvería muy variada para cada servidor de búsqueda, debido a que la variable con mayor escala definiría por completo la función haciendo insignificante el valor total de la otra variable. En este caso la memoria RAM es expresada en MB y la velocidad de CPU en MHz, garantizando de esta forma que ambas variables se encuentren en la misma escala.

Una vez obtenida la función de aptitud se calcula la capacidad de carga de cada servidor de búsqueda. Consiste en el porciento que representa la función de aptitud del servidor i con respecto a la sumatoria de los valores de aptitud de todos los servidores activos: $CC_i = \frac{Fa_i * 100}{SFA}$ $SFA = \sum_{i=1}^{N} Fa_i$

Donde:

CC_i: Capacidad de carga del servidor de búsqueda i.

SFA: Sumatoria de los valores de aptitud de aptitud de todos los servidores activos.

Con los porcentajes de carga calculados ya se está en condiciones de determinar el tamaño de la porción del banco de datos que puede asignársele a cada servidor de búsqueda, estableciendo una correspondencia entre el porcentaje de carga y el tamaño del banco de datos: $PBD_i = \frac{CC_i * BD}{100}$

Donde:

PBD_i: Porción del banco de datos que debe asignársele al servidor de búsqueda i.

BD: Tamaño del banco de datos.

Revista Cubana de Ciencias Informáticas
ISSN: 2227-1899 | RNPS: 2301
Vol. 8, No. 4, Octubre-Diciembre, 2014
Págs. 157-170
http://rcci.uci.cu

PBD_i: Representa la cantidad máxima de usuarios que puede ser asignada al servidor i para la búsqueda. Si se le asigna un valor mayor que este se estaría sobrecargando al servidor, mientras que si se le asigna un valor menor se desaprovecha su capacidad.

Tratamiento de fallos

Para tratar un fallo primeramente es necesario tener conocimiento de qué está ocurriendo un evento adverso. Para ello el sistema cuenta con varias vías para detectar que un servidor de búsqueda ha colapsado y para informar cuándo una funcionalidad no ha podido ser ejecutada.

La primera de ellas está asociada a la detección de los servidores de búsqueda que realizará el distribuidor frecuentemente en intervalos de tiempo. Esta inspección permitirá no solo conocer cuándo ha dejado de ofrecer servicios un servidor, sino también, detectar nuevos servidores de búsqueda.

Por el contrario el caso más crítico que puede presentarse es la caída de un servidor durante el proceso de búsqueda. Como cada servidor tiene cargada en su memoria RAM la porción del banco de datos que se la ha asignado, un fallo significa que existe un rango de datos en el cual no se realiza la búsqueda y que en el peor de los casos contiene el individuo a identificar. Cuando esta situación se presenta, otro servidor, una vez que ha terminado su búsqueda, asume el trabajo que no se ha realizado y envía una notificación al distribuidor informando el nombre en la red del servidor que ha interrumpido sus servicios para que posteriormente se realice una redistribución del banco de datos.

Pruebas

Debido a la necesidad de medir el rendimiento del sistema con un banco de datos de tamaño representativo para la validación de los resultados de la investigación, se utilizó un banco de datos con 14 mil usuarios para un total de 28 mil huellas dactilares almacenadas, pues por cada usuario se tomaron las impresiones dactilares de sus dos pulgares. En la Figura 2 se pueden observar los resultados obtenidos al identificar a individuos en distintos intervalos del banco de datos. Para ello el sistema de distribución empleó 3 y 6 servidores de búsqueda respectivamente.

Revista Cubana de Ciencias Informáticas
ISSN: 2227-1899 | RNPS: 2301
Vol. 8, No. 4, Octubre-Diciembre, 2014
Págs. 157-170
http://rcci.uci.cu

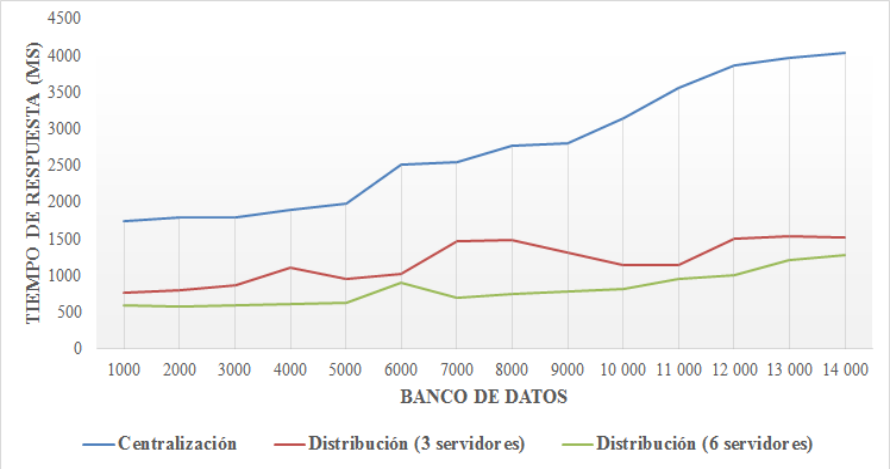

Figura 2 Comparación de tiempos de respuesta del AFIS centralizado y distribuido.

Al aumentar los datos a procesar (aumento del número de iteraciones) el modelo centralizado ralentiza la identificación con un incremento promedio de 117,0449 ms por cada 2 000 huellas a comparar, o sea, 1 000 individuos a analizar. Por el contrario, con el sistema de distribución los tiempos de respuesta del AFIS se reducen en más de un 50 % con solo 3 servidores de búsqueda. Nótese que al aumentar el número de servidores de búsqueda disminuyeron aún más los tiempos de respuesta. Debe tenerse en consideración que los tiempos de respuesta del AFIS con el sistema de distribución están afectados por la latencia de la infraestructura de red sobre la que se ejecute el sistema. Es por ello que en la Figura 2 se perciben incrementos y decrementos bruscos en los tiempos de respuesta, en lugar de una función creciente.

Aunque los resultados anteriores demuestran la efectividad de la distribución para optimizar el proceso de búsqueda, a continuación se presentarán los tiempos de búsqueda locales obviando la latencia de la red, o sea, el tiempo que demora el servidor de búsqueda donde se hace la identificación positiva[3] en culminar el proceso, obviando el tiempo que transcurre en la notificación del resultado a la capa del AFIS encargada de procesarlo (Ver Figura 3). El promedio de tiempo perdido por la latencia de la infraestructura de red (Mesa, 2009) donde se realizaron las pruebas fue de aproximadamente 254,054 ms. Sin embargo, como se ha podido apreciar, este tiempo de demora no afectó en gran medida los tiempos de respuesta reportados por el AFIS con el sistema de distribución.

3 Se refiere al servidor de búsqueda donde se encuentra la huella dactilar del individuo a identificar.

Revista Cubana de Ciencias Informáticas
ISSN: 2227-1899 | RNPS: 2301
Vol. 8, No. 4, Octubre-Diciembre, 2014
Págs. 157-170
http://rcci.uci.cu

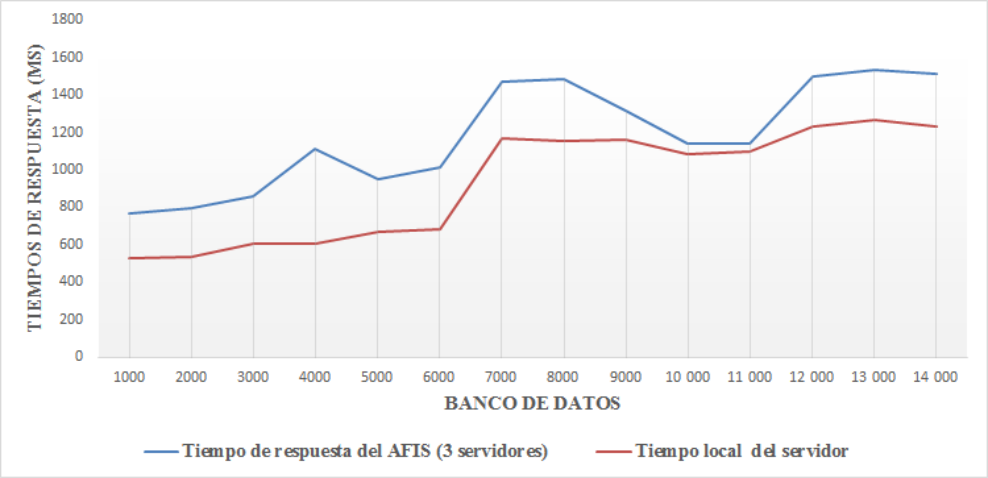

Figura 3 Tiempos de respuesta del AFIS y tiempos locales en los servidores de búsqueda.

Si se analizan las comparaciones por segundo que realiza el sistema de distribución, con solo seis servidores de búsqueda este es capaz de comparar 23 133,2 huellas. Aunque no supera los resultados de sistemas como el *ExpressID AFIS* y el *BIOMESYS AFIS* con 85 millones y más de un millón de comparaciones por segundo respectivamente, constituye un paso de avance para el desarrollo de sistemas biométricos en Cuba. Debe tenerse en cuenta que la cifra alcanzada no resulta modesta si se considera que solo estuvieron involucrados seis servidores de búsqueda *Dual Core* con 1GB de memoria RAM, características inferiores comparadas con los clústeres de búsqueda de los sistemas antes mencionados. Otras pruebas realizadas fueron las pruebas de estrés para validar el mecanismo de tolerancia a fallos implementado. Para ello se simuló un ambiente de despliegue inestable, interrumpiendo el funcionamiento de uno de los servidores de búsqueda intencionalmente durante el procesamiento de las solicitudes. Los resultados obtenidos demostraron que cuando colapsa un servidor de búsqueda durante el proceso de identificación, aunque la operación no se detiene, los tiempos de respuesta tienden a aumentar en la medida en que sea mayor la porción de datos cargada en cada servidor de búsqueda.

El tiempo de respuesta cuando está fallando un servidor de búsqueda (Ver Figura 4) se duplica en comparación con el funcionamiento del sistema en condiciones normales. Sin embargo, la estrategia de recuperación ante fallos utilizada garantiza una respuesta ante este evento que incluye seguir procesando nuevas solicitudes de identificación. Como puede observarse en la gráfica el tiempo de recuperación promedio es de 795.42 ms, pero vale destacar que puede reducirse si se adicionan más servidores de búsqueda.

Revista Cubana de Ciencias Informáticas
ISSN: 2227-1899 | RNPS: 2301
Vol. 8, No. 4, Octubre-Diciembre, 2014
Págs. 157-170
http://rcci.uci.cu

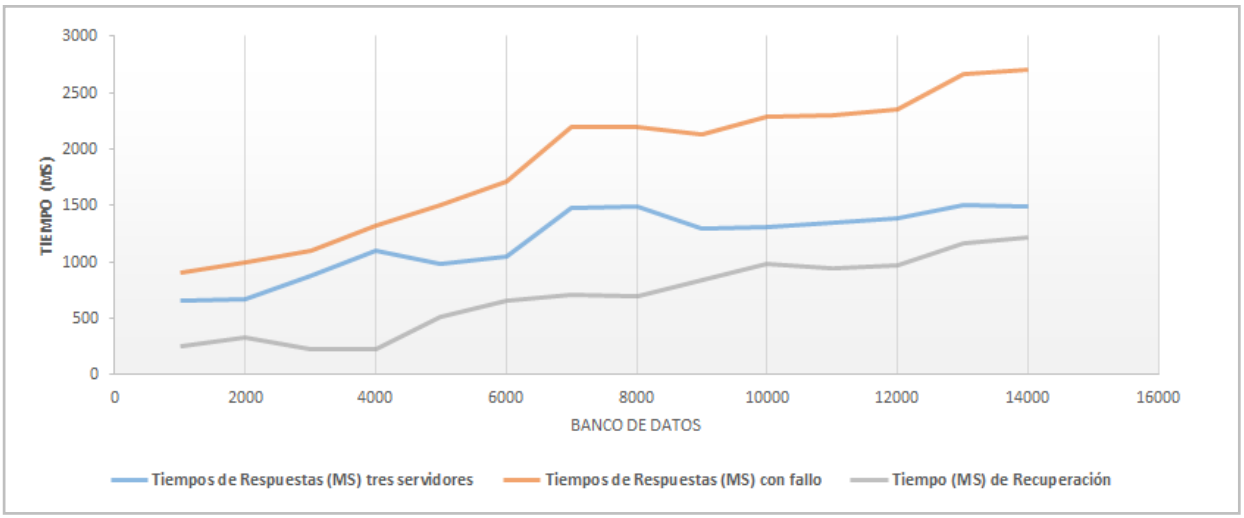

Figura 4 Tiempo de respuesta ante fallo y tiempo de recuperación para 3 servidores de búsqueda.

Conclusiones

Durante el transcurso de esta investigación se comprobó que el sistema desarrollado para la distribución del proceso de búsqueda de huellas dactilares, cobra especial importancia por el aporte económico que supone la inclusión de un AFIS eficiente en las soluciones informáticas del CISED, y particularmente su utilización en los sistemas de control de acceso dentro de la Universidad y el país. Algunas de las conclusiones obtenidas luego de terminada la primera versión del sistema para la distribución de la búsqueda son:

1. La definición de la arquitectura cliente-servidor y el uso del patrón n-capas permitieron que el sistema soporte el procesamiento distribuido, previendo el crecimiento modular y la alta escalabilidad, así como la implementación de mecanismos eficientes de tolerancia a fallos.
2. El componente encargado de la distribución que forma parte del sistema final, tiene un alto nivel de independencia, que puede ser usado por otros sistemas biométricos que solo tendrían que implementar la arquitectura especifica de sus servidores de búsqueda.
3. Las pruebas de rendimiento permitieron validar la hipótesis planteada, ya que el AFIS del CISED haciendo uso del sistema de distribución, reporta mejores tiempos de respuesta y garantiza que cuando un servidor colapse durante el proceso de identificación el sistema continúe funcionando.

Referencias

- BOUYSSOUNOUSE, B. 2005. Telecomunication Software Infracstructure. Embedded Systems Design: The ARTIST Roadmap for Research and Development. s.l.: Springer, 2005, pp. 131-132.
- BRYANT, R. 2012. [Online] 2012. [Cited: 18 enero 2013.] http://fedoraproject.org/wiki/Features/OpenStack_using_Qpid.
- COINTE, P. 1999. Meta-Level Architectures and Reflection: Second International Conference, Reflection'99 Saint-Malo, France, July 19-21, 1999 Proceedings. s.l.: Springer, 1999.
- COULOURIS, G. 2001. Sistemas Distribuidos. Madrid: Addison Wesley, 2001.
- DORMIDO, S. 2003. Procesamiento paralelo: teoría y programación. Madrid: s.n., 2003.
- FOUNDATION, APACHE SOFTWARE. 2012. ApacheQpid. [Online] [Cited: 18 enero 2012.] https://cwiki.apache.org/qpid/faq.html.
- GUTIÉRREZ, G, R. 2004. Estrategias para el balanceo dinámico de la carga en sistemas distribuidos. 2004.
- INNOVATRICS. 2013. ExpressID AFIS. [Online] 2013. [Cited: 16 enero 2013.] http://www.innovatrics.com/products/expressid-afis.
- KENDALL, K. E. Y KENDALL, J. E. 2005. Tecnología Cliente Servidor. Análisis y diseño. s.l.: Pearson Educación, 2005, pp. 622-624.
- KOMARINSKI, P. 2005. History of Automated Identification Systems. Automated Fingerprint Identification System (AFIS). s.l.: Elsevier Academic Press, 2005.
- MALTONI, D Y MAIO, D. 2009. Handbook of fingerprint recognition. Second Edition. Londres: Springer, 2009.
- MESA, A. 2009. Variables que afectan el desempeño de los cluster. Método para el manejo del balanceo de carga en sistemas de cómputo distribuido de alto desempeño. Colombia: s.n., 2009, p. 9.
- MUÑOZ, A L. 2009. Tesis Doctoral: Contribución al reconocimiento de huellas dactilares mediante técnicas de correlación y arquitecturas hardware para el aumento de prestaciones. Madrid: Universidad Carlos III de Madrid. Departamento de Tecnología Electrónica, 2009.
- RABBITMQ. RabbitMQ. [Online] RabbitMQ. [Cited: 19 Enero 2013.] http://www.rabbitmq.com/features.html..
- RIVERO, E C, GUARDIA, CARLOS R, REIG, J C. HERNÁNDEZ. 2004. Bases de datos distribuidas. Bases de datos relacionales: diseño físico. s.l.: Univ Pontifica Comillas, 2004.
- ROB, P Y CORONEL, C. 2004. Sistemas de bases de datos: diseño, implementación y administración. s.l. : Thomson, 2004.

– ROEBUCK, K. 2011. Advanced Message Queuing Protocol (Amqp): High-Impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors. 2011.
– VIDELA, A Y WILLIAMS, J.W. 2012. RabbitMQ in Action Distributed Messaging For Enyone. s.l.: Manning Publications Company, 2012.