

APLICACIÓN WEB PARA CONTROL, MONITOREO Y AVISO DE PROCESOS EN LA SUPERCOMPUTADORA

Martínez Paniagua, Carlos Alberto (1), Guryev, Igor (2)

1 [Ingeniería en Sistemas Computacionales, Universidad de Guanajuato] | [ca.martinezpaniagua@ugto.mx]

2 [DICIS, Ingenierías, Irapuato-Salamanca, Universidad de Guanajuato] | [guryev@ugto.mx]

Resumen

Piritakua, es el nombre que se le dio a la supercomputadora de la Universidad de Guanajuato ubicada en el departamento de estudios multidisciplinarios del campus Irapuato – Salamanca. Este equipo cuenta con un sistema operativo CentOS, que permite realizar miles de instrucciones por segundo. Todos los códigos ejecutados son desde modo consola y no se cuenta con una interfaz gráfica de usuario. Inspeccionar el progreso de los códigos puede llegar a ser un poco complicado ya que se tiene que realizar de forma presencial o en forma remota desde la consola dentro de la red de la propia Universidad. Se creó una aplicación web para que los códigos puedan enviar actualizaciones del progreso de sus ejecuciones. Se utilizan los servicios de Realtime Database de Firebase para recibir dichas actualizaciones. Cuando se detectan cambios en la base de datos se manda una notificación al dispositivo Android mostrando el progreso de la ejecución.

Abstract

Piritakua, is the name given to the supercomputer of the University of Guanajuato which is located on the department of multidisciplinary studies, Irapuato – Salamanca campus. The computer has a CentOS operating system, which allows to execute thousands of instructions per second. All codes are executed from console mode given that it does not count with a graphical user interface. Checking the progress of the codes can be complicated and it must be done personally or remotely from the console within the University itself. A web application was created to make possible for the codes to send updates of their progress. Services such as the Realtime Database of Firebase were used to receive these updates. When changes are detected in the database, a notification is sent to the Android device showing the progress of the execution.

Palabras Clave

Supercomputadora; aplicación web; aplicación Android; notificaciones de ejecución.

INTRODUCCIÓN

Una de las supercomputadoras con la que cuenta la Universidad de Guanajuato, en específico la del departamento de estudios multidisciplinarios, ubicado en Yuriria, Guanajuato, es utilizada por muchos investigadores que requieren una gran capacidad de cómputo.

Todos los códigos se ejecutan en forma de consola ya que el equipo no cuenta con una interfaz gráfica de usuario, si el usuario quiere ver el progreso tiene que revisarlo de forma presencial en la consola. El usuario solo podrá revisar el progreso si se escribieron líneas de código para que lo imprima en la consola. Si no lo hizo, el usuario no podrá ver el progreso y únicamente tendrá que esperar hasta que termine el código. También tendrá que estar revisando constantemente para saber cuándo terminó de ejecutarse.

Justificación

Se decidió desarrollar herramientas que ayudaran a los investigadores a mostrar sus progresos o mensajes en los dispositivos Android.

Se utilizaron herramientas de Firebase para disminuir la carga de procesamiento de la supercomputadora y que no afecte en el rendimiento. Otra ventaja, es que Firebase es multiplataforma y queda la opción de escalar el proyecto hacia otras plataformas o lenguajes.

MATERIALES Y MÉTODOS

El desarrollo de este trabajo está conformado por dos partes: crear el servidor en la supercomputadora y la aplicación para mostrar las notificaciones. El servidor es necesario para poder conectarse a Firebase por medio de PHP y enviar la información a la base de datos. La aplicación es para tener un servicio que nos este informando cuando haya modificaciones en la base de datos.

Para que el usuario pueda recibir las notificaciones deberá contar con un dispositivo Android con una versión API 19 correspondiente a Android 4.4 Kitkat o posterior y deberá contar con los servicios de Google Play 15.0.0 o posterior. El dispositivo y el servidor deberán contar con conexión a Internet.

Servidor computo

Para la creación del servidor se utilizó un servidor HTTP (Hypertext Transfer Protocol) Apache. El servidor HTTP Apache es un servidor web de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras.

Se instaló el servidor apache ejecutando las siguientes instrucciones en consola:

- `sudo yum install httpd`
- `sudo service httpd start`

Se instaló PHP con sus módulos correspondientes, para recibir el mensaje proveniente de los códigos:

- `sudo yum install php php-mcrypt php-cli php-gd php-curl php-mysql php-ldap php-zip php-fileinfo`

También se utiliza un SDK (Software Development Kit) de Firebase para enviar los datos:

- `<script src="https://www.gstatic.com/firebasejs/5.2.0/firebase.js"></script>`

El desarrollo del Proyecto está alojado en Github, donde se puede acceder al código:

- <https://github.com/CharlyMtzP/FirebaseUG>

Aplicación

La aplicación Android fue desarrollada en Android Studio 3.1.3 Build #AI-173.4819257. Se utilizaron dos servicios de Firebase: Authentication y Realtime Database.

Autenticación

Firebase Auth ofrece varios métodos para autenticar, entre ellos correo electrónico y contraseña, proveedores externos como Google o Facebook, o tu sistema de cuentas existente de manera directa. También ofrece verificación con el número de teléfono que es la que utilizamos nosotros para validar el teléfono y que la información esté asegurada.

Base de datos

Realtime Database Nos sirve para almacenar y sincronizar datos entre usuarios y dispositivos en tiempo real a través de una base de datos NoSQL alojada en la nube. Los datos actualizados se sincronizan entre distintos dispositivos conectados en milisegundos y permanecen disponibles si la app pierde la conexión a la red, lo que brinda una experiencia del usuario de alta calidad sin importar el estado de la conectividad.

RESULTADOS

Uso del servicio en el servidor

Dependiendo el lenguaje que se esté utilizando, el usuario tendrá que implementar lo necesario para poder abrir una dirección URL desde su código. También es aconsejable tener en consideración los tiempos en que se van a estar enviando las notificaciones, para evitar el exceso de avisos y no afectar el rendimiento del equipo.

La aplicación utilizará el número de teléfono del celular como identificador del progreso de código.

Para enviar una notificación el usuario debe ingresar la siguiente URL:

- `http://localhost/FirebaseUG/api/api.php?ID=NúmeroDeTeléfono&val=Mensaje`

NúmeroDeTeléfono: Deberá ser ingresado el número de celular con todo y su prefijo telefónico.

Mensaje: Aquí se deberá escribir el progreso del código o mensaje que el usuario necesite notificar.

Ejemplo:

- `http://localhost/FirebaseUG/api/api.php?ID=+524451111111&val=53%25`

Si se desean utilizar caracteres especiales en la URL se deberá poner el prefijo % seguido del número hexadecimal correspondiente al símbolo a enviar en la tabla de código ASCII.

Autenticación

La interfaz cuenta con dos campos, en una donde se debe escribir el prefijo telefónico y otro donde se debe ingresar el número de teléfono de 10 números.



IMAGEN 1: Actividad para inicio de sesión

Después de haber ingresado los campos y presionar el botón de iniciar sesión, la aplicación se encarga de verificar que el número sea correcto enviando un mensaje a ese mismo número. En cuanto se confirma el número automáticamente cambia a la actividad principal.

Base de datos

Cuando el usuario ya realizó la autenticación, se pasa a la actividad principal. En esta parte se inician los servicios de Realtime Database y se establece la conexión con la base de datos. El dispositivo está a la espera de recibir notificaciones por parte de la supercomputadora.



También se inicia un servicio en segundo plano para que el usuario no tenga que tener la aplicación abierta siempre que quiera ver las notificaciones. El servicio en segundo plano mostrará notificaciones cada vez que haya cambios en la base de datos.

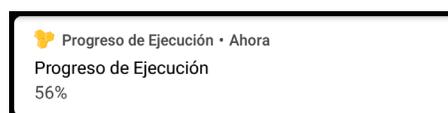


IMAGEN 2: Actividad principal

IMAGEN 3: Mensajes de notificación

CONCLUSIONES

La solución propuesta para mostrar el progreso de las ejecuciones es muy efectiva. El consumo de ancho de banda es muy corto en las aplicaciones, <0.6 KB para recepción y < 0.3KB para confirmación de recepción. Además de que el consumo es únicamente por modificación a la base de datos.

El tiempo de notificación desde que se modifica en la base de datos es de 1 segundo aproximadamente.

El desarrollo de este trabajo queda abierto a escalabilidad, de plataformas y lenguajes.

AGRADECIMIENTOS

Agradezco todo el apoyo brindado de la Universidad de Guanajuato y al Departamento de Estudios Multidisciplinarios.

REFERENCIAS

- Google Developers. (2018). Realtime Database para Firebase. firebase.google. Recuperado de <https://firebase.google.com/docs/database/>
- Google Developers. (2018). Firebase Authentication. firebase.google. Recuperado de <https://firebase.google.com/docs/auth/>
- DigitalOcean™ Inc. (2018). How To Install Linux, Apache, MySQL, PHP (LAMP) stack On CentOS 6. Recuperado de <https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-centos-6>
- PHP. (2018). Manual de PHP. Recuperado de <http://php.net/manual/es/>
- Mozilla y colaboradores individuales. (2005-2018). MDN web docs. Recuperado de <https://developer.mozilla.org/es/>