

MODIFICACIONES ARQUITECTURA UNAIRLINES



UNIVERSIDAD NACIONAL DE COLOMBIA SEDE BOGOTÁ

Presentado por:
DIEGO CABALLERO
ALEJANDRO GIRALDO
JHONATAN GUZMÁN
CRISTIAN PARADA
CHRISTIAN ZULUAGA

Presentado a:
HENRY ROBERTO UMAÑA

BOGOTÁ D.C
2015

Para cumplir con los requerimientos de alta disponibilidad y tolerancia a Fallos se hace uso del modelo de actividades descrito por Rozanski, esto con el fin de que las tareas críticas que se puedan realizar por medio de las funcionalidades ofrecidas por el sistema de Aerolíneas UNAirlines (pago de tickets, registro de usuarios, edición de vuelos, etc), puedan realizarse de manera exitosa sin ningún inconveniente o fallo.

En el laboratorio anterior, se realizó las respectivas modificaciones a la arquitectura del sistema de la aplicación UNAirlines con el fin de soportar un alto número de solicitudes en una corta cantidad de tiempo. Sin embargo, tal arquitectura con la que se cuenta actualmente, no está garantizado la realización de tareas críticas sin interrupciones, las cuales pueden ser ocasionadas por factores externos que afectan el funcionamiento de la aplicación como por ejemplo que uno de los nodos de la aplicación deje de funcionar. en los que está desplegada la aplicación deje de funcionar.

De esta forma, mostraremos a continuación las características de la arquitectura trabajada para el laboratorio de escalabilidad y sus respectivos cambios para implementar el requerimiento no funcional de alta disponibilidad y tolerancia a fallos en nuestra aplicación, donde tendremos nuevamente el cluster con las tres máquinas físicas usadas anteriormente, en las que una cumple el rol de balanceador de carga.

- **Nodo 1:**
Procesador: Intel Core i5-4200U CPU 1.60 GHz
Memoria instalada(RAM): 8.00 GB
Tipo de sistema: sistema operativo de 64 bits, procesador x64
- **Nodo 2:**
Procesador: Intel Core i5-2450U CPU 2.50 GHz
Memoria instalada(RAM): 4.00 GB
Tipo de sistema: sistema operativo de 64 bits, procesador x64
- **Balanceador de carga:**
Procesador: Intel Core i3 CPU 2.53 GHz
Memoria instalada(RAM): 4.00 GB
Tipo de sistema: sistema operativo de 64 bits, procesador x64

Las características a nivel de software son las siguientes:

- **Balanceador de Carga:**
Sistema Operativo: Kubuntu 14.04
IP: 192.168.0.1
JDK 8u40
Apache HTTP Server 2.2.25
- **Nodo 1:**
Sistema Operativo: Kubuntu 14.04
IP: 192.168.0.2
JDK 8u40
Red Hat Jboss Enterprise Application Platform

- Nodo 2:**
 Sistema Operativo: Kubuntu 14.04
 IP: 192.168.0.3
 JDK 8u40
 Red Hat Jboss Enterprise Application Platform

Cambios en la arquitectura:

Como se especificó anteriormente, para cumplir con el requerimiento de alta disponibilidad y tolerancia a fallos, se contará con el mismo clúster con el que se trabajó para cubrir el requerimiento de escalabilidad. Sin embargo a nivel de software si se realizaron cambios en algunos archivos ubicados en archivos de configuración del balanceador y los nodos, al igual que la inclusión de algunas líneas de código en archivos fuente de la aplicación, al igual que de uno de sus archivos de configuración. Al realizar estos cambios, tuvimos que desplegar nuevamente la aplicación en cada uno de los nodos para poder visualizar los nuevos cambios.

Cambios en los Nodos:

Para cada uno de los nodos se cambió el nombre de la máquina, cambiando las líneas que contienen el nombre actual de la máquina en los archivos `/etc/hosts` y `/etc/hostname` por `nodo1` y `nodo2` respectivamente con el fin de poder identificar la máquina a la cual se accede por medio del balanceador de carga al entrar a la aplicación. Además en el archivo de configuraciones del servidor `standalone-ha.xml` se agregó el atributo **`instance-id="nodo1"`** en la etiqueta `subsystem`.

Cambios en el Balanceador:

En el laboratorio anterior se realizó la instalación del servidor web Apache junto con los módulos `mod_jk` y `mod_lbmethod_bybusyness`. El siguiente paso que se realizó fue la modificación del archivo de configuración del servicio `httpd` para que la máquina física pudiera funcionar como balanceador de carga. Así mismo se generó el archivo `workers.properties` en el que se especificó la forma en la que estaría configurado el balanceo de cargas a los nodos que se comunicarían con el balanceador. Para este laboratorio, se incluyó al final de dicho documento la propiedad **`worker.loadbalancer.sticky_session=1`**, con el fin de que el balanceador de carga sea capaz de redirigir las peticiones a un determinado nodo una vez que se ha establecido la conexión.

Cambios en la aplicación UNAAirlines:

En el `index.jsp` se agrega después de la etiqueta `body` las líneas: Backend JBoss:
`<%=InetAddress.getLocalHost().getCanonicalHostName()%>` y
 Session ID: `<%=request.getSession().getId()%>` además de una variable contador. Con esto logramos varias cosas, al acceder a la aplicación por medio del balanceador reconocer cuál nodo está usando en esa sesión, obtener el id de la sesión, y con el contador suma las veces que se accede a la aplicación.

