

Proyecto Final

1. Objetivos

- 1.1. Complementar los conceptos vistos en clase a través del estudio de tecnologías de tendencia que soportan los servicios de telecomunicaciones estudiados
- 1.2. Desarrollar la habilidad del estudiante para enfrentarse al estudio y configuración de nuevas tecnologías por su cuenta
- 1.3. Compartir la experiencia a través de una presentación en clase.

2. Introducción

Este documento presenta un listado de opciones de proyectos que pueden ser seleccionados como proyecto final de curso. Estos proyectos se clasifican en los siguientes grupos:

- LISTA DE PROYECTOS 1: balanceo de carga de servidores web. A este grupo pertenecen los proyectos 1 y 2
- LISTA DE PROYECTOS 2: balanceo de carga de bases de datos. A este grupo pertenecen los proyectos 3, 4, 5, 6 y 7.
- LISTA DE PROYECTOS 3: monitoreo de infraestructura. A este grupo pertenecen los proyectos 8 y 9.
- LISTA DE PROYECTOS 4: investigación y desarrollo de tecnologías como soporte a sistemas tolerantes a fallos (caso de estudio: atención de desastres). A este grupo pertenecen los proyectos 10 y 11.
- OTROS PROYECTOS: proyectos propuestos por el estudiante, los cuales deben ser aprobados por el profesor.

Los proyectos se desarrollarán en grupos de 3 (min) o 4 (max) estudiantes. Cada grupo deberá registrar el proyecto seleccionado en la plataforma classroom del curso. No podrá haber proyectos repetidos para dos grupos de estudiantes diferentes.

3. LISTA DE PROYECTOS 1: balanceo de carga de servidores web

Herramientas a utilizar

Apache mod_proxy_balancer, NGINX, Vagrant, VirtualBox, Artillery
<https://www.artillery.io/>

Requerimientos

Este proyecto consiste en la implementación de tres requerimientos principales:

1. Clúster de servidores web con balanceo de carga
2. Configuración de ambiente para pruebas de carga

PARTE 1: Clúster de servidores web con balanceo de carga

Se requiere implementar un clúster de servidores web con balanceo de carga. El balanceador de carga funcionará como Frontend del servidor web. Cada vez que se envíe una petición al balanceador de carga, éste se encargará de redirigir la petición a uno de los servidores del clúster. Dichos servidores alojan los recursos necesarios para resolver las peticiones.

En ningún caso las peticiones realizarán directamente a los servidores web, sino que el balanceador de carga decidirá que servidor será el encargado de procesar la petición. Los servidores en el backend deben correr un servicio web como. En la siguiente figura se muestra un ejemplo de la configuración a realizar.

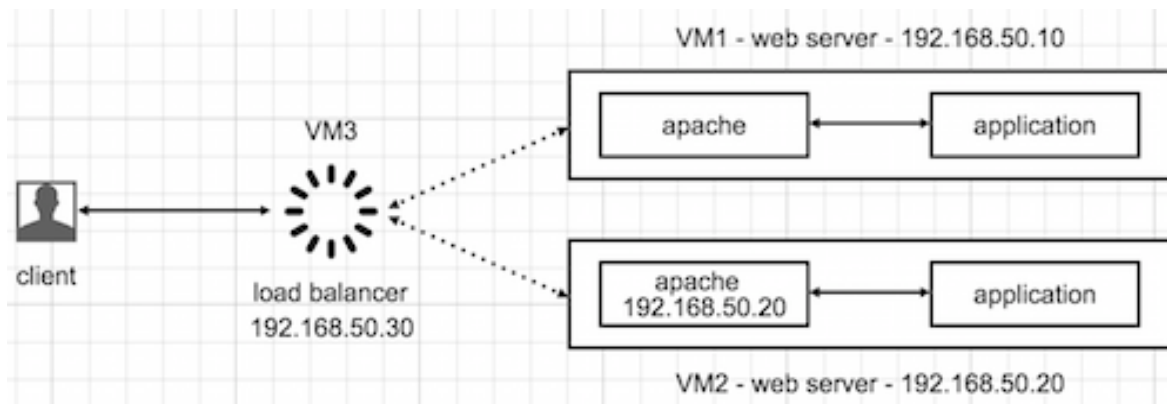


Figura 1 Configuración Requerida

PARTE 2. Pruebas de Carga

Se requiere probar la ejecución de múltiples peticiones simultáneas al clúster de servidores web. Con estas pruebas podrá verificar el número de usuarios concurrentes que la aplicación soportará y qué carga hará que la aplicación deje de funcionar.

Para las pruebas de carga se sugiere usar <https://www.artillery.io/> . Una aplicación de fuente abierta para pruebas de desempeño.

Pruebas Mínimas Esperadas

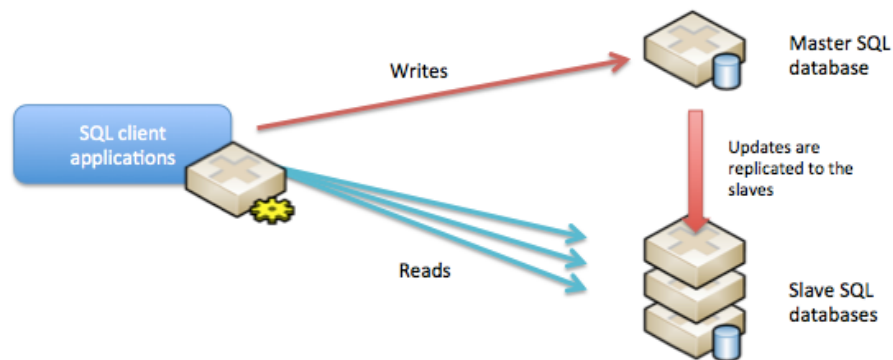
Después de realizar todas las configuraciones requeridas en los puntos anteriores, realice al menos las siguientes pruebas:

1. Realice pruebas de carga variando los parámetros de las pruebas y analice qué sucede
2. Agregue diferentes números de servidores web y verifique cómo responde el sistema
3. Explore las diferentes estadísticas que arroje el balanceador de carga. Tales como número de peticiones resueltas, tasa de llegada de peticiones, porcentaje de fallas en la atención de peticiones, etc.

Proyectos

- **Proyecto 1:** Balanceador de carga usando el modulo de apache mod_proxy_balancer + Pruebas de Carga con Artillery (<https://www.artillery.io/>)
- **Proyecto 2:** Balanceador de carga usando NGINX + Pruebas de Carga con Artillery (<https://www.artillery.io/>)

4. LISTA DE PROYECTOS 2: Balanceo de Carga de Bases de datos



Herramientas a utilizar

Vagrant, VirtualBox, MySQL, otros.

Requerimientos

Implementar un balanceador de bases de datos MySQL con el objetivo de alcanzar una mayor eficiencia y velocidad en lo referente a escritura y lectura de datos, incrementar la tolerancia a fallos y obtener una alta disponibilidad. Para esto se debe configurar un pool de servidores de bases de datos.

Pruebas Mínimas Esperadas

Se deben realizar al menos las siguientes pruebas:

1. Comprobar el funcionamiento con queries de lectura
2. Comprobar el funcionamiento básico con queries de escritura

3. Realizar pruebas de carga usando herramientas como Sysbench o similar.

- **Proyecto 3:** Balanceo de Carga de bases de datos con MySQL y Nginx
- **Proyecto 4:** Balanceo de Carga de bases de datos con MySQL y ProxySQL
- **Proyecto 5:** Balanceo de Carga de bases de datos con MySQL y MySQL Router
- **Proyecto 6:** Balanceo de Carga de bases de datos con MySQL y HAProxy
- **Proyecto 7:** Balanceo de Carga de bases de datos con MariaDB MaxScale

5. LISTA DE PROYECTOS 3: monitoreo de infraestructura

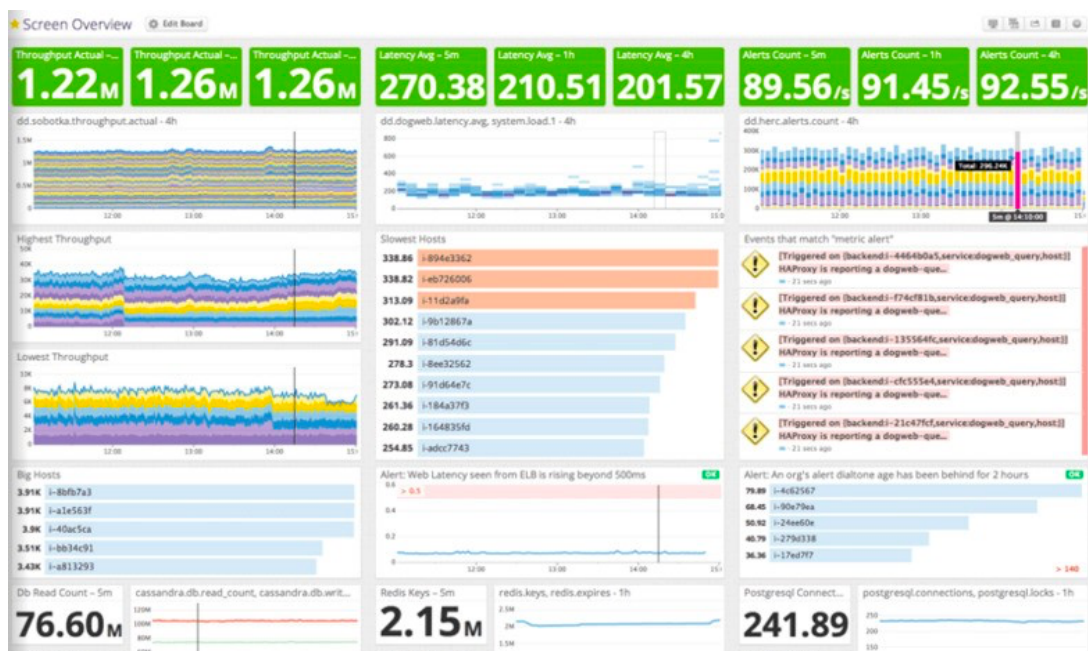


Figura 2 Monitoreo de Infraestructura

Herramientas a utilizar

Vagrant, VirtualBox, Apache Nagios, DataDog

Requerimientos

Este proyecto consiste en la implementación de tres requerimientos principales:

1. Montar una red de equipos
2. Usar una herramienta de nube para el monitoreo de infraestructura de red montada
3. Mostrar los resultados de monitoreo haciendo uso de los Dashboards que provee la herramienta usada.

Pruebas Mínimas Esperadas

Después de realizar todas las configuraciones requeridas en los puntos anteriores, realice al menos las siguientes pruebas:

4. Verificación de monitoreo de equipos a través del dashboard que provee la herramienta seleccionada
5. Verificación de cambios en tiempo real del estado mostrado de los equipos de acuerdo a cambios en su estado

Proyectos

- **Proyecto 8.** Monitoreo de infraestructura con Nagios. Se requiere monitorear una red usando la plataforma Nagios. Para esto se requiere que los estudiantes monten una red real y realicen su monitoreo. Enlace Herramienta: <https://www.nagios.org>
- **Proyecto 9.** Monitoreo de Cluster con DataDog. Se requiere realizar monitoreo completo de un cluster computacional usando la plataforma DataDog. Enlace Herramienta: <https://www.datadoghq.com/>

6. LISTA DE PROYECTOS 4: investigación y desarrollo de tecnologías como soporte a sistemas tolerantes a fallos (caso de estudio: atención de desastres)

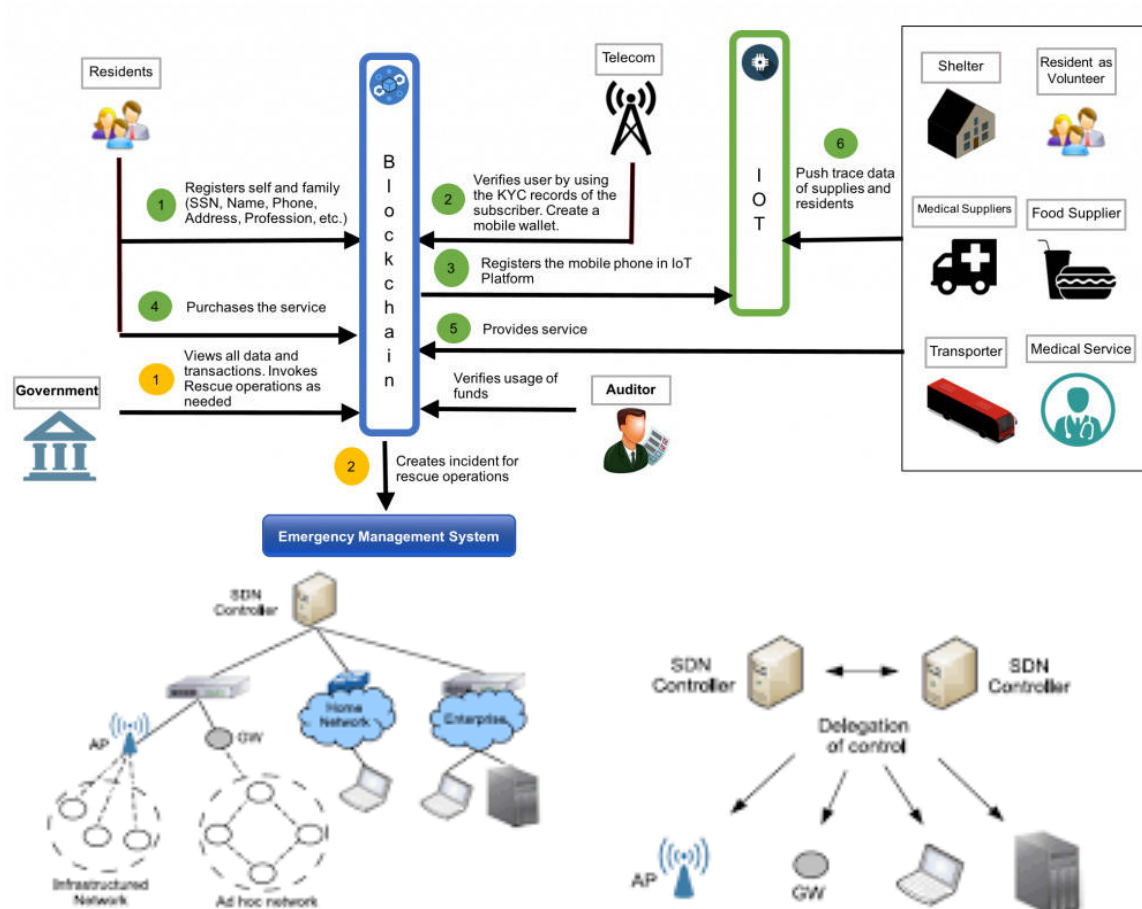


Figura 3 Investigación y desarrollo para sistemas tolerantes a fallas

Herramientas a utilizar

Vagrant, VirtualBox, Lenguaje de programación Python (u otro), OpenFlow, entre otros,

Requerimientos

Este proyecto consiste en la implementación de tres requerimientos principales:

1. Profundizar en la investigación de tecnologías avanzadas de servicios de telecomunicaciones y generar informe resultado de la investigación
2. Implementar prototipo de prueba de los conceptos investigados

Pruebas Mínimas Esperadas

1. Pruebas de funcionamiento del prototipo funcional implementado para la tecnología seleccionada

Proyectos

- **Proyecto 10. Implementar una red peer to peer basada en blockchain** que facilite la gestión de operaciones durante un desastre natural. Se trata de proveer la infraestructura necesaria para facilitar la comunicación entre diferentes partes (gobierno, organismos de socorro, ciudadanos), permitiendo el flujo de información y ayudas de manera eficiente y transparente. Ejemplo: <https://developer.ibm.com/tutorials/automate-deployment-ibm-blockchain-platform-network/>
- **Proyecto 11. SDN (Software Defined Networking)** para atención de desastres. Se requiere implementar una red que permita la reacción efectiva frente a desastres que afecten su disponibilidad. Para esto se hará uso de conceptos de SDN (Software Defined Networking), permitiendo la adaptación, flexibilidad de la red para reconfigurarse y su rápida recuperación ante desastres. Enlace: Referencia: <https://hal.inria.fr/hal-01400746/file/sdn-public-safety%286%29.pdf>
- **Proyecto 12. Estudio de mecanismos de tolerancia a fallos en servicios de nube.** Se requiere investigar los mecanismos para la comunicación de servicios tolerantes a fallos en servicios de nube. Entre los objetivos están: investigar sobre los diferentes patrones que soportan la tolerancia a fallos en servicios de nube; estudiar las tecnologías de fuente abierta que permiten la implementación de dichos patrones; Implementar un mecanismo de tolerancia a fallos seleccionado. Referencia: <https://resilience4j.readme.io/docs>.
- **Proyecto 13. Exploración de la Programación Paralela Portable en HPC-Cloud.** Se requiere investigar los mecanismos para la implementación de soluciones que hagan uso de programación paralela portable. Entre los objetivos están: estudiar las tecnologías de fuente abierta para la implementación de soluciones que hagan uso de programación paralela portable; Realizar pruebas de portabilidad y desempeño en diferentes ambientes computacionales. Referencias: <https://github.com/kokkos/kokkos>

7. OTRO PROYECTOS 4: *proyecto de interés personal de los estudiantes, el cual debe ser aprobado por el profesor.*

8. Entregables y Evaluación

El trabajo final se realizará en grupos de 3 (max. 4) estudiantes. **No se aceptarán grupos de menos de 3 personas.**

Todos los entregables se deben subir a Google Classroom. Se deben entregar los siguientes productos en la semana 16 de clase:

ENTREGABLES

- **Fecha de entrega: martes 8 de noviembre de 2022**
- Informe de la solución
 - **Entregable a subir:** Informe en formato IEEE, máx. 7 páginas.
- Presentación de la solución obtenida y configuración final
 - **Entregable a subir:** Diapositivas en repositorio GitHub del proyecto
- Sustentación de la solución implementada
 - **Entregable a subir:** Repositorio GitHub del proyecto con archivos de configuración, aprovisionamiento, Readme con paso a paso, etc.

NOTA: Tiempo máximo presentación + sustentación: 20 minutos por grupo.

EVALUACIÓN

Se usarán los siguientes criterios de evaluación

SO-2: Capacidad de aplicar diseño en ingeniería para producir soluciones que satisfagan las necesidades de las personas, teniendo en cuenta la salud pública, seguridad y bienestar, así como factores globales, culturales, sociales y económicos.			
Profesor:		Programa:	
Nombre del curso:		Semestre:	

Indicador de Desempeño:		PI-2.1: Aplica un proceso estructurado de diseño en ingeniería para producir alternativas de solución que satisfagan las necesidades de las personas y consideren los atributos de diseño.	Medio de Evaluación:	Trabajo final del curso	
Criterios de Evaluación		Niveles			
		Deficiente (0,0 - 2,9)	Aceptable (3,0 - 3,7)	Bueno (3,8 - 4,4)	Sobresaliente (4,5 - 5,0)
20%	1. Descripción del problema y sus necesidades y definición de requerimientos y restricciones	<p>Presenta una lista deficiente e incoherente de necesidades para caracterizar el problema y el usuario</p> <p>Relaciona erradamente los requerimientos y restricciones con el listado de las necesidades del problema</p>	Presenta una lista adecuada de necesidades para caracterizar el problema, pero no define correctamente los requerimientos y restricciones, ni los relaciona con las necesidades	Presenta una lista adecuada de necesidades para caracterizar el problema, define correctamente los requerimientos y restricciones, pero no hay una relación de requerimientos, y restricciones con las necesidades	Presenta una lista adecuada de necesidades para caracterizar el problema, define correctamente los requerimientos y restricciones, y hay una relación de requerimientos, y restricciones con las necesidades

25%	2. Generación y selección de alternativas de solución.	Presenta un diseño deficiente e incoherente en las alternativas de solución. Establece criterios deficientes y poco relevantes para la correcta evaluación de las alternativas de solución.	Presenta un diseño donde evidencia algunas incoherencias con los requerimientos planteados. Establece criterios adecuados, pero poco relevantes para la correcta evaluación de las alternativas de solución.	Presenta un diseño donde evidencia correctamente la relación con los requerimientos planteados. Establece criterios adecuados, pero poco relevantes para la correcta evaluación de las alternativas de solución	Presenta un diseño donde evidencia correctamente la relación con los requerimientos planteados. Establece criterios adecuados y relevantes para la correcta evaluación de las alternativas de solución
30%	3. Desarrollo del prototipo funcional	Falla al construir un prototipo funcional	Construye un prototipo funcional con serias deficiencias en sus partes.	Construye un prototipo funcional que presenta fallas en algunas de sus partes.	Construye un prototipo funcional sin fallas en todas sus partes funcionales.
25%	4. Sustentación Individual	No realiza una Sustentación adecuada de los conceptos requeridos para el desarrollo de la asignación	Realiza una Sustentación aceptable de los conceptos requeridos para el desarrollo de la asignación	Realiza una Sustentación adecuada de los conceptos requeridos para el desarrollo de la asignación	Realiza una Sustentación sobresaliente de los conceptos requeridos para el desarrollo de la asignación