Proyecto Final

(LISTA 1)
Servidor Web con Balanceo de Carga

Load Balancing

- Aplicación web sin balanceo de carga

Load Balancing de Nivel 4

Proxy directo vs. Proxy Inverso

Fuente: https://www.dokry.com/15688

Load Balancing de Nivel 7

Algoritmos de Balanceo de Carga

- Round Robin (por defecto): Servidores seleccionados por turnos

- Leastconn: Servidores con el menor numero de conexiones son seleccionados

- Source: Selecciona el servidor basándose en la dirección IP de origen

Alta Disponibilidad y Health Checks

1. Active/Passive Cluster is healthy
2. Primary node fails
3. Floating IP is assigned to Secondary node

Balanceadores de Carga

- Modulo de Apache (mod_proxy_balancer)
- Linux Virtual Servers (LVS): Balanceador simple de nivel 4 incluido en varias distribuciones de Linux
- Nginx: Servidor web que incluye capacidades de proxy y load-balancing
- En la nube
  - Orquestación de contenedores: Docker Swarm, Kubernetes
  - IaaS: OpenNebula, OpenStack

Pruebas de Carga

- Para las pruebas de carga se sugiere Artillery (https://www.artillery.io/)

- Aplicación de fuente abierta para pruebas de desempeño en servidores web.

- Con estas pruebas podrá verificar el numero de usuarios concurrentes que la aplicación soportará y que carga hará que la aplicación deje de funcionar.

Requerimientos

- Ver en Classroom

Requerimientos

- Este conjunto de proyectos consiste en la implementación de dos requerimientos principales:

  - Clúster de servidores web con balanceo de carga
  - Configuración de ambiente para pruebas de carga

Proyectos

- **Proyecto 1:** Balanceador de carga usando el modulo de apache mod_proxy_balancer + Pruebas de carga con Artillery.

- **Proyecto 2:** Balanceador de carga usando NGINX + Pruebas de Carga con Artillery

Proyecto Final

(LISTA 2)
Balanceo de Carga de Bases de datos

Balanceo de Carga en Bases de Datos

Requerimiento

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 Esperadas

- Se deben realizar al menos las siguientes pruebas:
- Comprobar el funcionamiento con queries de lectura
- Comprobar el funcionamiento básico con queries de escritura
- Realizar pruebas de carga usando herramientas como Sysbench o similar.

Proyectos

- **Proyecto 3**: Balanceo de Carga de bases de datos con 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

Proyecto Final

(LISTA 3)
Monitoreo de Infraestructura

Monitoreo de Infraestructura

Requerimientos

- Este conjunto de proyectos consiste en la implementación de tres requerimientos principales:

  - Montar una red de equipos (ej. Máquinas virtuales)
  - Usar una herramienta de nube para el monitoreo de infraestructura de red montada
  - Mostrar los resultados de monitoreo haciendo uso de los dashboards que provee la herramienta usada.

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 y realicen su monitoreo. Enlace Herramienta: https://www.nagios.org

- **Proyecto 9.** Monitoreo de Cluster HAProxy con DataDog. Se requiere realizar monitoreo completo de un cluster HAProxy usando la plataforma DataDog. Enlace Herramienta: https://www.datadoghq.com/

Proyecto Final

(LISTA 4)
Investigacion y Desarrollo

Tecnologias como soporte a sistemas tolerantes a fallos (caso de estudio: atencion de desastres)

Requerimientos

- Profundizar en la investigación de tecnologías avanzadas de servicios de telecomunicaciones y generar informe resultado de la investigación

- Implementar prototipo de prueba de los conceptos investigados

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.

Proyectos

- **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

Proyectos

- **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.

Proyectos

- **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://www.intel.com/content/www/us/en/developer/tools/oneapi/overview.html#gs.udlzaw

  https://github.com/kokkos/kokkos

Referencias

- An introduction to HAProxy and Load Balancing Concepts: https://www.digitalocean.com/community/tutorials/an-introduction-to-haproxy-and-load-balancing-concepts
- Vagrant Provisioning. https://www.vagrantup.com/docs/provisioning/
- Nagios. https://www.nagios.org
- DataDog. https://www.datadoghq.com/
- SDN https://hal.inria.fr/hal-01400746/file/sdn-public-safety%286%29.pdf
- One API: https://www.intel.com/content/www/us/en/developer/tools/oneapi/overview.html#gs.udlzaw
- Kokkos: https://github.com/kokkos/kokkos