

Análisis y monitorización de aplicaciones a través de plugins con Naemon

Sofía Fernández Moreno

Universidad de Granada

Septiembre de 2019

Estado del arte

Realización del despliegue de Naemon

Pruebas de carga

Pruebas de carga en un sistema

Modelado de las pruebas

Conclusiones y futuros trabajos

¿Qué es la monitorización?

Comparativa herramientas de monitorización

Naemon

Estado del arte

¿Qué es la monitorización?

La **monitorización** es aquella en la que se toman las medidas preventivas y consecuentes con la información que se obtienen de todos los dispositivos que se encuentran conectados a una red, para evitar posibles eventos que hacen que se interrumpan el correcto funcionamiento de alguno de ellos

Dentro de este concepto es importante aplicar el uso del protocolo **SNMP** (Simple Network Management Protocol).

***SNMP** permite el intercambio de información amplia entre los diferentes dispositivos de red mediante consultas de forma remota (polling) y mediante mensajes basándose en eventos (traps).*

Comparativa herramientas de monitorización

Incluir imagen de todas

Estado del arte

Realización del despliegue de Naemon

Pruebas de carga

Pruebas de carga en un sistema

Modelado de las pruebas

Conclusiones y futuros trabajos

¿Qué es la monitorización?

Comparativa herramientas de monitorización

Naemon

Funcionamiento

Archivo de configuración principal

añadir imagen

Estado del arte

Realización del despliegue de Naemon

Pruebas de carga

Pruebas de carga en un sistema

Modelado de las pruebas

Conclusiones y futuros trabajos

¿Qué es la monitorización?

Comparativa herramientas de monitorización

Naemon

Tipos de objetos

Estado del arte

Realización del despliegue de Naemon

Pruebas de carga

Pruebas de carga en un sistema

Modelado de las pruebas

Conclusiones y futuros trabajos

¿Qué es la monitorización?

Comparativa herramientas de monitorización

Naemon

Interfaz GUI: Thruk

Realización del despliegue de Naemon

Estado del arte

Realización del despliegue de Naemon

Pruebas de carga

Pruebas de carga en un sistema

Modelado de las pruebas

Conclusiones y futuros trabajos

Entorno de desarrollo

Desarrollo de despliegue

Docker Engine

Estado del arte
Realización del despliegue de Naemon
Pruebas de carga
Pruebas de carga en un sistema
Modelado de las pruebas
Conclusiones y futuros trabajos

Entorno de desarrollo
Desarrollo de despliegue

Dockerfile con imagen de Naemon

Orquestación estática de aplicaciones

La **orquestación estática** es aquella que el sistema requiere una configuración más manual de los recursos y no permite el escalado de forma muy eficiente.

Estado del arte
Realización del despliegue de Naemon
Pruebas de carga
Pruebas de carga en un sistema
Modelado de las pruebas
Conclusiones y futuros trabajos

Entorno de desarrollo
Desarrollo de despliegue

Docker-Compose

Pruebas de carga

Estado del arte
Realización del despliegue de Naemon
Pruebas de carga
Pruebas de carga en un sistema
Modelado de las pruebas
Conclusiones y futuros trabajos

Pruebas de rendimiento
Comparativa de herramientas
Locust

Tipos de pruebas

Estado del arte
Realización del despliegue de Naemon
Pruebas de carga
Pruebas de carga en un sistema
Modelado de las pruebas
Conclusiones y futuros trabajos

Pruebas de rendimiento
Comparativa de herramientas
Locust

Elección a realizar

Estado del arte
Realización del despliegue de Naemon
Pruebas de carga
Pruebas de carga en un sistema
Modelado de las pruebas
Conclusiones y futuros trabajos

Pruebas de rendimiento
Comparativa de herramientas
Locust

Comparativa de herramientas

Estado del arte
Realización del despliegue de Naemon
Pruebas de carga
Pruebas de carga en un sistema
Modelado de las pruebas
Conclusiones y futuros trabajos

Pruebas de rendimiento
Comparativa de herramientas
Locust

Funcionamiento de Locust

Estado del arte
Realización del despliegue de Naemon
Pruebas de carga
Pruebas de carga en un sistema
Modelado de las pruebas
Conclusiones y futuros trabajos

Pruebas de rendimiento
Comparativa de herramientas
Locust

Locustfile

Pruebas de carga en un sistema

Estado del arte

Realización del despliegue de Naemon

Pruebas de carga

Pruebas de carga en un sistema

Modelado de las pruebas

Conclusiones y futuros trabajos

Locust en Docker-Compose

Naemon en Docker-Compose

Enlazado de Locust

Estado del arte

Realización del despliegue de Naemon

Pruebas de carga

Pruebas de carga en un sistema

Modelado de las pruebas

Conclusiones y futuros trabajos

Locust en Docker-Compose

Naemon en Docker-Compose

Resultados

Estado del arte

Realización del despliegue de Naemon

Pruebas de carga

Pruebas de carga en un sistema

Modelado de las pruebas

Conclusiones y futuros trabajos

Locust en Docker-Compose

Naemon en Docker-Compose

Creación de host y servicios

Creación de plugin como ejemplo

Estado del arte

Realización del despliegue de Naemon

Pruebas de carga

Pruebas de carga en un sistema

Modelado de las pruebas

Conclusiones y futuros trabajos

Locust en Docker-Compose

Naemon en Docker-Compose

Resultados

Modelado de las pruebas

¿Qué es la carga de trabajo?

La **carga de trabajo** es el conjunto de todas las peticiones que el sistema recibe de su entorno durante un periodo de tiempo dado.

El análisis de la carga es un papel fundamental en cualquier estudio en los que hay que determinar **índices de rendimiento**, estos se encuentran directamente relacionados con la carga y no se pueden expresar de forma independiente a ésta. Además el índice de rendimiento siempre debe ir determinado de la información de la carga bajo la que fue determinado.

Estado del arte
Realización del despliegue de Naemon
Pruebas de carga
Pruebas de carga en un sistema
Modelado de las pruebas
Conclusiones y futuros trabajos

Carga de trabajo
PNP4Nagios

PNP4Nagios en Dockerfile

Captura de resultados

Añadir demo.mp4

Conclusiones y futuros trabajos

Conclusiones

En cuanto a los resultados obtenidos en dicho análisis se aprecia como el sistema responde de forma positiva durante los treinta minutos de comprobación, ya que no pierde paquetes a la hora de realizar el PING, aplicando tiempos RTA bastante reducidos, además a la hora de mandar peticiones HTTP, éste responde de forma favorable puesto que los tiempos de respuesta son lo suficientemente pequeños para que no haya problemas de pérdida de conexión, además el tamaño de los paquetes generados son siempre los mismos por lo que no tendremos ninguna desfragmentación generada.

Trabajos futuros

En cuanto a los trabajos futuros a partir del actual, el principal sería la realización de forma automatizada del despliegue pudiendo apoyar- nos de la herramienta **Ansible**. Otra idea futura sería la adaptación de la pila **ELK(Elasticsearch, Logstash y Kibana)** de **Elastic** para recoger todos los registros generados durante la monitorización, haciendo que la búsqueda, análisis y visualización de los datos aparezcan con mayor facilidad en los dashboard, además de poder manejarse gran cantidad de datos de forma eficiente.

Gracias por su atención



sofiafernandezmoreno/TFG