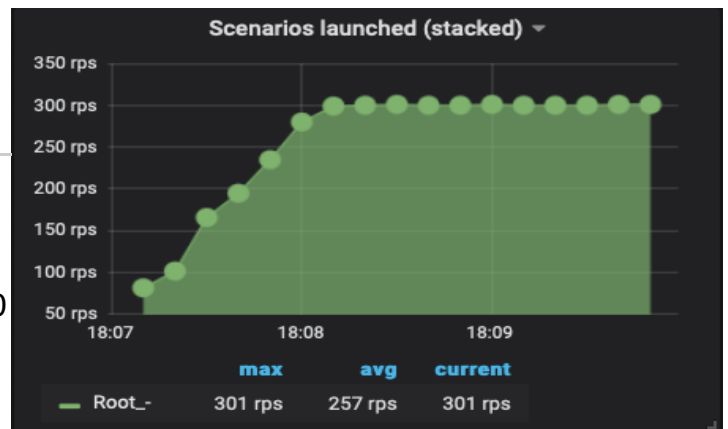


Informe de Perf Testing Arquitectura de Software

Comenzaremos este informe detallando las bases del proyecto llevado a cabo. En este trabajo practico nos planteamos el objetivo de analizar atributos de calidad de un sistema dummy el cual tiene unos pocos endpoints. A travez de herramientas como Artillery, Graphite, Grafana, y CAdvisor generaremos y recopilaremos informacion de forma tal que podamos llevar a cabo un analisis seguido de una reflexion final de las conclusiones obtenidas. Nuestro sistema dummy tendra dos servidores, uno en Node.js y otro en Python (el cual utilizara Flask y Gunicorn para funcionar como servidor).

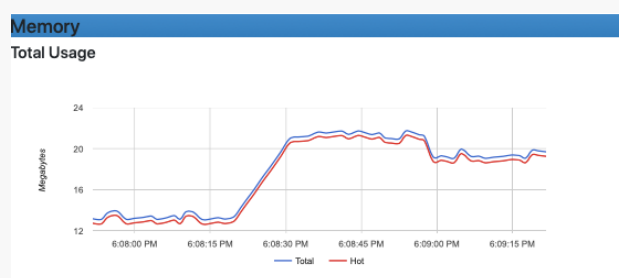
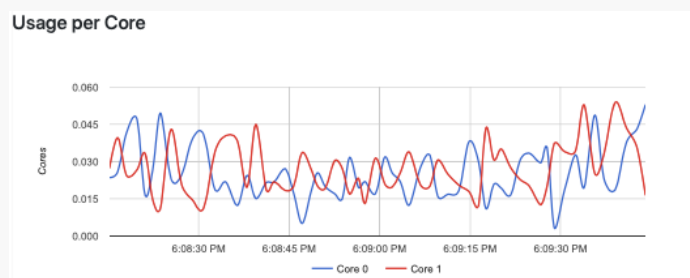
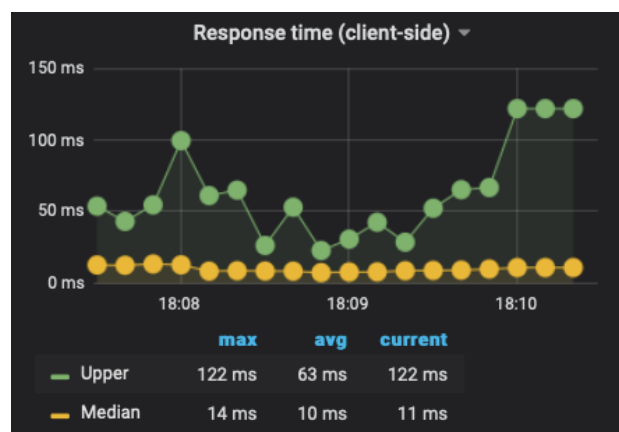
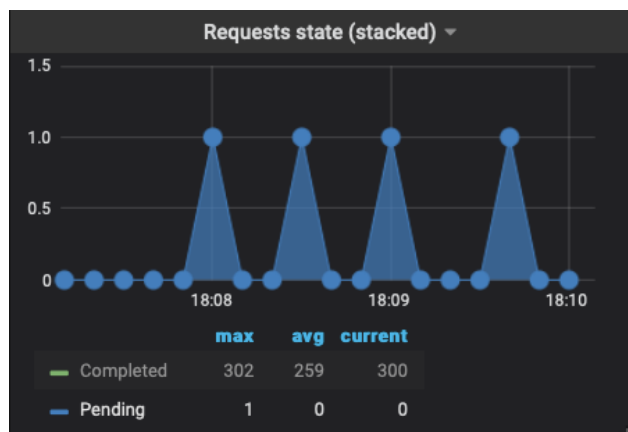
Load test

Para el caso del load test primero seguimos un esquema muy simple de "Phases". Primero habria una rampa de requests subiendo de 5 a 30 en 60 segundos para simular un comienzo estable y luego mantuvimos la cantidad de requests por segundo en 30 los siguientes dos minutos. Este esquema se repitio en todos los otros casos de load test



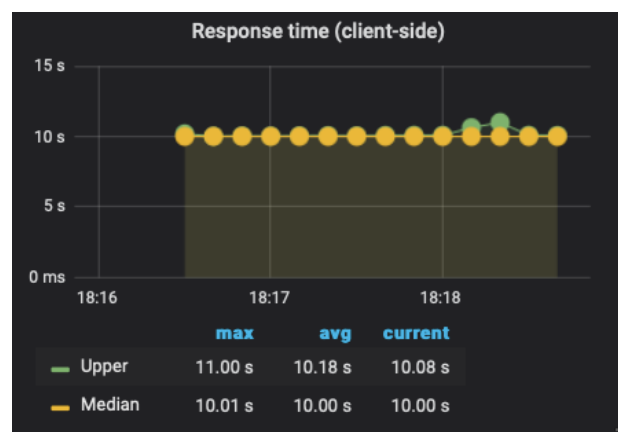
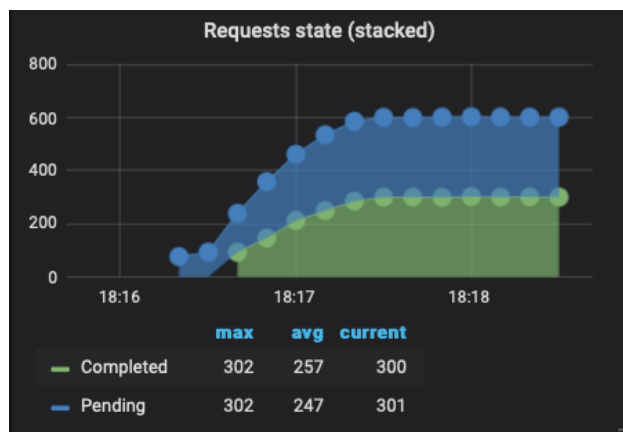
Caso Load Test Basico

En el caso del Ping hay dos particularidades importantes a notar, la cantidad de requests pendientes es bastante estable a travez de los periodos de 10 segundos. El otro hecho a notar muy importante es que el tiempo de respuesta tiene una varianza muy alta con respecto a los otros dos endpoints. En cuanto al uso de la memoria y el procesador no se ve algo muy extraño, el uso de los procesadores se mantiene bastante constante a travez de toda la experiencia y la memoria tambien

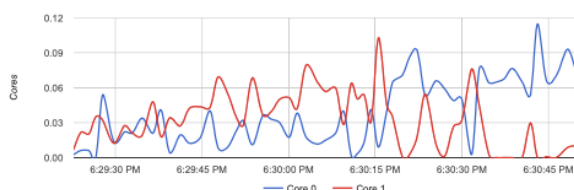


Caso Load Test Node Timeout

En el load test de Node timeout esta la particularidad muy importante a notar, esta es que el response time tiene una varianza muy pequeña, casi no hay diferencia entre los tiempos medios y los mas altos. La cantidad pending requests es constante, siempre se mantiene en 300 en la fase constante. En cuanto al uso de memoria se puede ver tambien como es mas gradual en este caso. Lo mas curioso de este endpoint es como el timeout parece ordenar las ejecuciones del servidor manteniendo un servidor con poca variabilidad. Aunque en el caso del ping la variabilidad no era extremadamente alta, quizas para un servicio que necesite respuestas en tiempo constante esta informacion es muy util

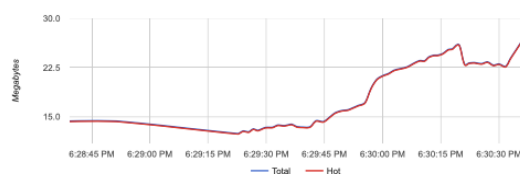


Usage per Core



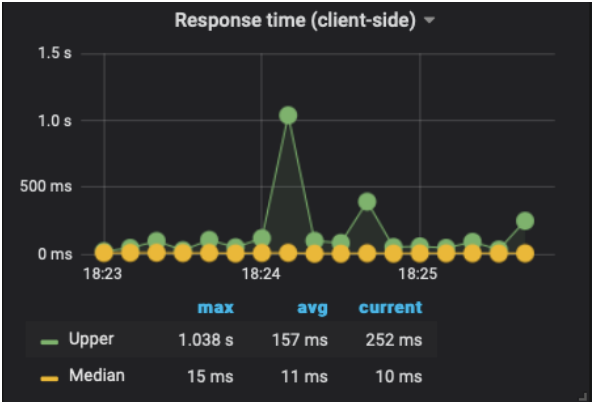
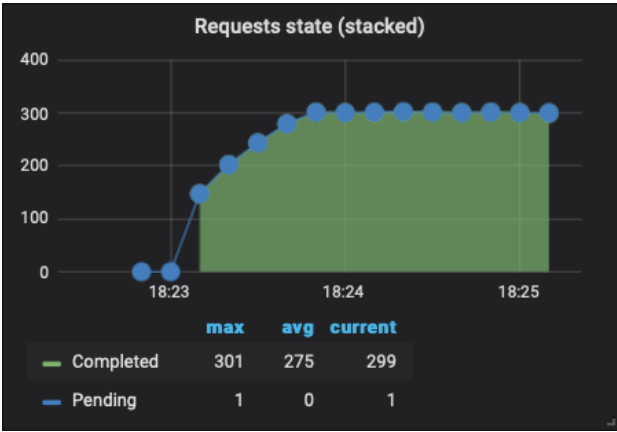
Memory

Total Usage

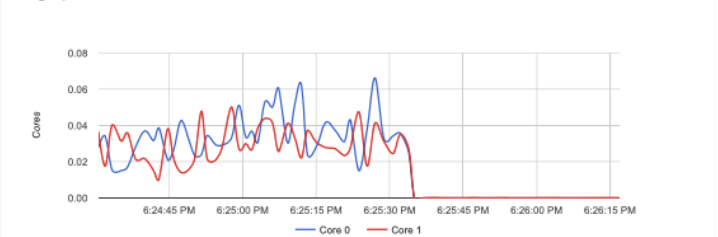


Caso Load Test Node Intensive

En el load test de Node intensive podemos ver que al igual que en el caso del Ping la cantidad de requests que estan pendientes siempre se mantiene bastante bajo, y en cuanto al response time podemos ver tambien una varianza mucho menor, aunque haya ocasionalmente algunos disparos hacia arriba por lo general se mantienen bastante juntas las dos curvas.



Usage per Core



Memory

Total Usage

