

Plan de Pruebas de Performance - Evolución Conector-

AGESIC

08/05/2020



Contenido

Resumen Ejecutivo	3
Introducción	4
Cronograma y Equipo	4
Objetivos	4
Sistema Bajo Pruebas	5
Configuración 1	5
Configuración 2	5
Configuración 3	5
Plan de Pruebas	6
Scripts de Prueba	6
Escenarios de Carga	6
Datos de Prueba	7
Criterios de Aceptación	7
Infraestructura	8
Monitorización	8
Plan de Ejecuciones	9
Anexo I: Configuraciones	11
Generación de Access Logs	11
Generación de logs del Garbage Collector	11
Configurar algoritmo del Garbage Collector	12

Resumen Ejecutivo

Los objetivos de las pruebas descritas en este plan se limitan al Conector y al componente de Ruteo.

En el caso del Conector se busca evaluar su evolución en relación a las pruebas que se realizaron para la versión anterior del mismo, así como también ayudar a definir una línea base de desempeño.

Para la solución de Ruteo se tiene dos grandes objetivos:

- Conocer su estado debido al aumento de actividad reciente.
- Hallar la configuración óptima para los criterios de degradación.

Durante el proceso de ejecución de las pruebas se medirá la performance de los componentes que se vean impactados por las mismas, mediante la recolección de indicadores de salud de cada nodo del sistema.

Para eso se ejecutarán pruebas utilizando dos servicios mock diferentes, uno con un delay de 300 ms para representar un tiempo aceptable de respuesta, y otro con un delay más alto (entre 10 a 30 segundos) para las pruebas sobre el servicio de degradación. Estos serán estresados con la herramienta Gatling (específica para simulación de carga sobre protocolo HTTP).

Las pruebas se ejecutarán en los siguientes escenarios:

- **Conector:**
 - **Baseline:** 1 UV iterando 30 veces (mensajes simples).
 - **Baseline:** 1 UV iterando 30 veces (mensajes con adjunto MTOM 5 Mb).
 - **Carga Objetivo:** 150 UV escalando en 10 minutos y sosteniendo la carga por 20 minutos (mensajes simples).
 - **Carga Objetivo:** 3 UV escalando en 10 minutos y sosteniendo la carga por 20 minutos (mensajes con adjunto MTOM 5 Mb).
 - **Stress:** 300 UV escalando en 15 minutos y sosteniendo la carga por 45 (mensajes simples).
 - **Stress:** Escalando hasta 6 UV en 15 minutos y manteniendo la carga durante 1 hora (mensajes con adjunto MTOM 5 Mb).
 - **Stress:** Escenario mixto escalando hasta 300 UV con mensajes simples y 6 UV con MTOM en 15 minutos y sosteniendo la carga por 45.
- **Ruteo:**
 - **Baseline:** 1 UV iterando 30 veces (mensajes simples).
 - **Carga Objetivo:** Escenario mixto escalando hasta 80 TPS con mensajes simples y 4 TPS con MTOM en 15 minutos y sosteniendo la carga por 45.

Introducción

En el presente documento se describe el Plan de Pruebas de Performance a realizar sobre la evolución de la versión 4.0 tag 6.2 Conector de la PDI desarrollada por Kreitech, así como también sobre la solución de Ruteo de la misma.

Se describen las pruebas, ambientes, datos de prueba, monitorización y plan de ejecución a seguir, para este proyecto.

Cronograma y Equipo

Las pruebas se ejecutarán fuera de horario de oficina, ya que algunos componentes de red son compartidos con el ambiente de producción. Por esto y por disponibilidad de los componentes preparados para probar, se planifica realizar las ejecuciones en las últimas semanas de mayo.

El equipo de Abstracta se encarga de la preparación de las pruebas (automatización de scripts, mock de pruebas, escenarios de carga, etc.), y el equipo de AGESIC se encarga de la monitorización y manejo de la infraestructura.

Objetivos

En este proyecto se plantean 3 objetivos:

- Evaluar la performance de la evolución del Conector.
- Evaluar la performance de la solución de Ruteo de la PDI, debido al aumento de actividad en la misma.
- Hallar la configuración óptima de los criterios de degradación para el Ruteo.

Durante el proceso de ejecución de las pruebas se medirá la performance de los componentes que se vean impactados por las mismas, mediante la recolección de indicadores de salud de cada nodo del sistema.

Para eso se ejecutarán pruebas utilizando un servicio mock de "*ConsultaProveedores*" en el caso de mensajes SOAP simples y para el caso de los mensajes con MTOM se configurará Data Power para responder el mismo adjunto MTOM que recibe como se hizo en las pruebas de Ruteo ejecutadas en 2017.

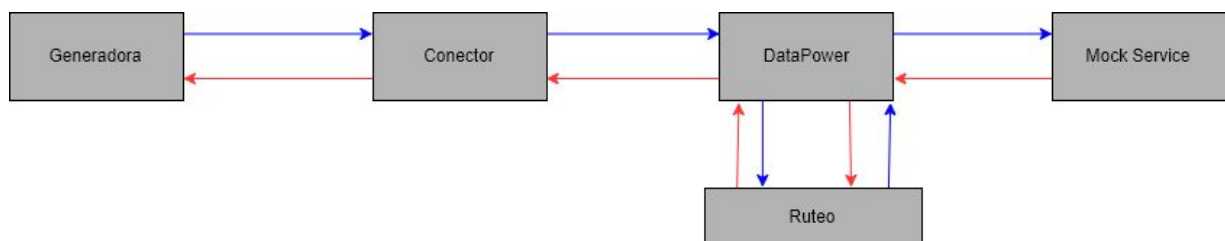
En ambos casos se utilizará la herramienta Gatling para generar tráfico (específica para simulación de carga sobre el protocolo HTTP).

Sistema Bajo Pruebas

Las pruebas se ejecutarán utilizando Data Power de producción, la solución de Ruteo de del ambiente de testing y servicios mock.

Dado los distintos objetivos del proyecto se deben considerar 3 configuraciones distintas para la ejecución de las pruebas.

Configuración 1



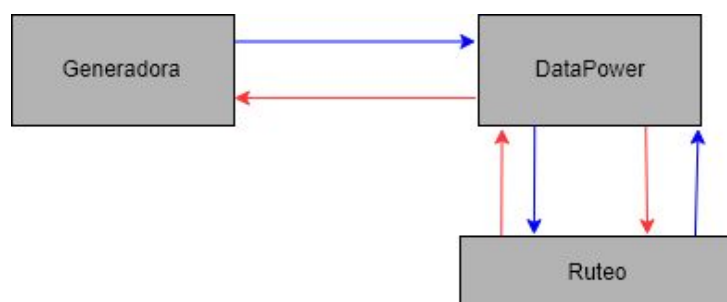
Con esta configuración se busca evaluar la performance del Conector en un escenario similar al real en interacción con Data Power y utilizando el Ruteo.

Configuración 2



En esta configuración se busca evaluar la performance únicamente de la solución de Ruteo para el envío de mensajes SOAP simples los cuales serán respondidos por el servicio Mock.

Configuración 3



Esta configuración tiene el mismo objetivo que la anterior para el caso de envío de mensajes SOAP con adjunto MTOM, los cuales serán respondidos por Data Power.

Plan de Pruebas

En esta sección se describe el plan de pruebas en el que se basará la ejecución de las pruebas de performance. En términos generales se utilizará la herramienta Gatling para ejecutar simulaciones de prueba a nivel de servicios SOAP sobre HTTP.

Scripts de Prueba

Para evaluar tanto la performance del Conector y del Ruteo en términos de los dos primeros objetivos planteados en dicha sección, se planifica la utilización de los dos scripts desarrollados para las pruebas del Ruteo en 2017:

- Un script que envía un mensaje simple, sin MTOM.
- Otro script que envía un mensaje con MTOM de 5 Mb.

Por otra parte, para cumplir con el tercer objetivo se planifica un script que consuma todos los servicios que utilizan la funcionalidad de degradación del Ruteo.

Escenarios de Carga

Para definir los escenarios de carga, se utilizaron las siguientes tres fuentes:

- El informe de las pruebas de performance realizadas por Abstracta en 2016 sobre la versión 2 del Conector.
- El informe de las pruebas de performance realizadas por Abstracta en 2017 sobre el Ruteo actual de la PDI.
- Métricas sobre el uso productivo del Ruteo de la PDI al día de hoy.

De esta manera se plantean los siguientes escenarios de carga sobre cada sistema objetivo:

- **Conector:**
 - **Baseline:** 1 UV iterando 30 veces (mensajes simples).
 - **Baseline:** 1 UV iterando 30 veces (mensajes con adjunto MTOM 5 Mb).
 - **Carga Objetivo:** 150 UV escalando en 10 minutos y sosteniendo la carga por 20 minutos (mensajes simples).
 - **Carga Objetivo:** 3 UV escalando en 10 minutos y sosteniendo la carga por 20 minutos (mensajes con adjunto MTOM 5 Mb).
 - **Stress:** 300 UV escalando en 15 minutos y sosteniendo la carga por 45 (mensajes simples).
 - **Stress:** Escalando hasta 6 UV en 15 minutos y manteniendo la carga durante 1 hora (mensajes con adjunto MTOM 5 Mb).
 - **Stress:** Escenario mixto escalando hasta 300 UV con mensajes simples y 6 UV con MTOM en 15 minutos y sosteniendo la carga por 45.

- **Ruteo:**

- **Baseline:** 1 UV iterando 30 veces (mensajes simples).
- **Carga Objetivo:** Escenario mixto escalando hasta 80 TPS con mensajes simples y 4 TPS con MTOM en 15 minutos y sosteniendo la carga por 45.

Sobre este último escenario se desea utilizar tanto el script de mensajes simples como el script que consume un servicio mock con delay que utilice degradación.

Para los escenarios con degradación se plantea crear un servicio mock con un delay en la respuesta y configurar el servicio de degradación teniendo en cuenta el P95 de los tiempos de respuesta del servicio real.

Los escenarios de carga antes descritos para evaluar el Conector, serán ejecutados con la solución hospedada en Tomcat como en Wildfly.

Datos de Prueba

Considerando que la performance del componente de ruteo no depende de los datos que viajan en los mensajes o parámetros, no se ve la necesidad de tener datos de prueba que deban variar, y de esta manera se decide simular la carga contra un mock de un servicio fijo o contra Data Power directamente en el caso de MTOM.

Para el caso de las pruebas con MTOM, se podrá utilizar distintos tamaños de adjuntos, dependiendo del resultado de las pruebas en la medida en la que sean ejecutadas.

Criterios de Aceptación

Se tienen los siguientes criterios de aceptación para el Conector:

- Identificar la máxima cantidad de TPS que soporta.
- Identificar el máximo tamaño de archivo adjunto en servicios con MTOM que soporta en relación a la cantidad de usuarios concurrentes.
- Conocer el tiempo extra (*overhead*) que agrega.
- Conocer su comportamiento condiciones de stress.

En cuanto al Ruteo se espera que sea capaz soportar la carga definida y que los criterios utilizados para definir la degradación de los servicios modele de mejor manera la realidad evitando cuellos de botella.

Infraestructura

Para desplegar el Conector se utilizará una VM con Centos 7, 1 CPU, 2 GB de memoria RAM y 20 GB de disco duro.

En cuanto al nodo de ruteo, se clonará uno de los de Producción y se utilizará de forma aislada al resto de la solución de PDI, con excepción de DataPower. Contará con 4 vCpu 10GB de memoria RAM y 55GB de HDD.

Se utilizará el Data Power de Producción que cuenta con 2 procesadores de 10 núcleos 2.80 GHz Intel Xeon E5-2680V2, 2 discos duros de 1200 GB configurados en RAID 1 y 192 GB de memoria RAM

Para las generadoras se utilizará VMs con CentOS 7, 4 CPU, 16GB de memoria RAM y 30GB de disco. Es importante verificar que exista la menor cantidad de saltos de red entre la generadora, el ruteo y el Conector, en este sentido es importante tener la misma consideración entre la VM donde se hospede el Conector y Data Power.

Monitorización

AGESIC cuenta con Zabbix, para una monitorización completa de los distintos componentes de la infraestructura.

A nivel del conector se monitorizará tanto su implementación en Tomcat como en Wildfly mediante JVisualVM (herramienta que provee la JDK), de manera de tener consumos de memoria y manejo del heap de la JVM

Se modificarán los xml de configuración de ambos Conectores para generar logs de acceso y así monitorizar el *time taken*. Ver anexo [Generación de Access Logs](#).

También se generarán logs del Garbage Collector de los Conectores con la opción `-verbose:gc`. Ver anexo [Generación de logs del Garbage Collector](#).

Se obtendrán los tiempos de respuesta del Data Power desde su log.

Con respecto a la/s VM/s donde corran el/los mocks de los servicios a utilizar y en la VM donde se despliegue el Conector, se contará con NMON para tener lecturas a nivel de sistema operativo.

Es importante la monitorización durante la ejecución de pruebas, no solo con el objetivo de documentar el consumo de recursos, sino que ya que se utilizará el Data Power de Producción y por tanto se debe controlar que existan degradaciones que puedan afectar a los usuarios de la PDI.

Plan de Ejecuciones

De los scripts y escenarios antes especificados, se plantea el siguiente plan de ejecución:

Conector			
Escenario	Script	Tipo de Mensaje	Servidor
Baseline	1 UV iterando 30 veces	Mensajes simples	Tomcat y Wildfly
Baseline con (MTOM)	1 UV iterando 30 veces	Mensajes con adjunto MTOM 5 Mb	Tomcat y Wildfly
Carga Objetivo	150 UV escalando en 10 minutos y sosteniendo la carga por 20 minutos	Mensajes simples	Tomcat y Wildfly
Carga Objetivo (MTOM)	3 UV escalando en 10 minutos y sosteniendo la carga por 20 minutos	Mensajes con adjunto MTOM 5 Mb	Tomcat y Wildfly
Stress	300 UV escalando en 15 minutos y sosteniendo la carga por 45	Mensajes simples	Tomcat y Wildfly
Stress (MTOM)	Escalando hasta 6 UV en 15 minutos y manteniendo la carga durante 1 hora	Mensajes con adjunto MTOM 5 Mb	Tomcat y Wildfly
Stress (Mixto)	Escalando hasta 300 UV con mensajes simples y 6 UV con MTOM en 15 minutos y sosteniendo la carga por 45.	Mensajes simples y mensajes con adjunto MTOM 5 Mb	Tomcat y Wildfly

Ruteo			
Escenario	Script	Tipo de Mensaje	Otras Columnas (configs,etc)
Baseline	1 UV iterando 30 veces	Mensajes simples	
Carga Objetivo	Escalando hasta 80 TPS con mensajes simples y 4 TPS con MTOM en 15 minutos y sosteniendo la carga por 45.	Mensajes simples y mensajes con adjunto MTOM 5 Mb	
Carga Objetivo	Escalando hasta 80 TPS con mensajes simples y 4 TPS con MTOM en 15 minutos y sosteniendo la carga por 45.	Mensajes simples y mensajes con adjunto MTOM 5 Mb	Utilizando degradación

Anexo I: Configuraciones

Generación de Access Logs

En el caso de Tomcat ya estaba configurado para generar logs de acceso, pero se debe modificar el patrón de logeo en el archivo **conf\server.xml** a

```
%a %A %h %t %u &quot;%r&quot;; %s %b %D
```

quedando de esta forma:

```
<!-- Access log processes all example.
Documentation at: /docs/config/valve.html
Note: The pattern used is equivalent to using pattern="common" -->
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" pattern="%a %A %h %t %u &quot;%r&quot;; %s %b %D" prefix="logs" />
</Host>
```

Para Wildfly se deben agregar en el archivo **standalone\configuration\standalone.xml** la siguiente configuración debajo del tag Location:

```
<access-log pattern="%a %A %h %t %u &quot;%r&quot;; %s %b %D"
directory="{jboss.server.log.dir}" prefix="access" suffix=".log"
worker="default"/>
```

```
<subsystem xmlns="urn:jboss:domain:undertow:4.0">
  <buffer-cache name="default"/>
  <server name="default-server">
    <http-listener name="default" socket-binding="http-testing" max-post-size="30000000" redirect-socket="https-testing" record-request-start-time="true"/>
    <http-listener name="production" socket-binding="http-prod" max-post-size="30000000" redirect-socket="https-prod" record-request-start-time="true"/>
    <https-listener name="default-ssl" socket-binding="https-testing" max-post-size="30000000" security-realm="SslRealm" record-request-start-time="true"/>
    <https-listener name="production-ssl" socket-binding="https-prod" max-post-size="30000000" security-realm="SslRealm" record-request-start-time="true"/>
    <host name="default-host" alias="localhost">
      <location name="/" handler="welcome-content"/>
      <access-log pattern="%a %A %h %t %u &quot;%r&quot;; %s %b %D" directory="{jboss.server.log.dir}" prefix="access" suffix=".log" worker="default" />
      <filter-ref name="server-header"/>
      <filter-ref name="x-powered-by-header"/>
      <http-invoker security-realm="ApplicationRealm"/>
    </host>
  </server>
</subsystem>
```

Generación de logs del Garbage Collector

Para Tomcat se debe modificar el archivo **bin\setenv.bat**

Agregando las siguientes opciones:

```
-Xlog:gc:%CATALINA_HOME%\logs\gclog.log -verbose:gc
```

a las que ya estan en **CATALINA_OPTS**

Para Wildfly se modificaron dos archivos, primero en **bin\standalone.conf.bat** se descomenta la línea **GC_LOG="true"**

Luego en **bin\standalone.bat** se deben cambiar las siguientes opciones

```
if not "%PRESERVE_JAVA_OPTS%" == "true" (
  if "%GC_LOG%" == "true" (
    rem Add rotating GC logs, if supported, and not already defined
    echo "%JAVA_OPTS%" | findstr /I "\-verbose:gc" > nul
    if errorlevel == 1 (
      rem Back up any prior logs
      move /y "%JBOSS_LOG_DIR%\gc.log.1" "%JBOSS_LOG_DIR%\backupgc.log.1" > nul 2>&1
      move /y "%JBOSS_LOG_DIR%\gc.log.0" "%JBOSS_LOG_DIR%\backupgc.log.0" > nul 2>&1
      move /y "%JBOSS_LOG_DIR%\gc.log.2" "%JBOSS_LOG_DIR%\backupgc.log.2" > nul 2>&1
      move /y "%JBOSS_LOG_DIR%\gc.log.3" "%JBOSS_LOG_DIR%\backupgc.log.3" > nul 2>&1
      move /y "%JBOSS_LOG_DIR%\gc.log.4" "%JBOSS_LOG_DIR%\backupgc.log.4" > nul 2>&1
      move /y "%JBOSS_LOG_DIR%\gc.log.*.current" "%JBOSS_LOG_DIR%\backupgc.log.current" > nul 2>&1

      "%JAVA%" -verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=5 -XX:GCLogFileSize=3M -Xloggc:"%JBOSS_LOG_DIR%\gc.log"
      if not errorlevel == 1 (
        if not exist "%JBOSS_LOG_DIR%" > nul 2>&1 (
          mkdir "%JBOSS_LOG_DIR%"
        )
      )
      set JAVA_OPTS=%JAVA_OPTS% -verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=5 -Xloggc:"%JBOSS_LOG_DIR%\gc.log"
    )
  )
)
```

por `-verbose:gc -Xlog:gc:"%JBOSS_LOG_DIR%\gc.log"` quedando así:

```
if not "%PRESERVE_JAVA_OPTS%" == "true" (
  if "%GC_LOG%" == "true" (
    rem Add rotating GC logs, if supported, and not already defined
    echo "%JAVA_OPTS%" | findstr /I "\-verbose:gc" > nul
    if errorlevel == 1 (
      rem Back up any prior logs
      move /y "%JBOSS_LOG_DIR%\gc.log.1" "%JBOSS_LOG_DIR%\backupgc.log.1" > nul 2>&1
      move /y "%JBOSS_LOG_DIR%\gc.log.0" "%JBOSS_LOG_DIR%\backupgc.log.0" > nul 2>&1
      move /y "%JBOSS_LOG_DIR%\gc.log.2" "%JBOSS_LOG_DIR%\backupgc.log.2" > nul 2>&1
      move /y "%JBOSS_LOG_DIR%\gc.log.3" "%JBOSS_LOG_DIR%\backupgc.log.3" > nul 2>&1
      move /y "%JBOSS_LOG_DIR%\gc.log.4" "%JBOSS_LOG_DIR%\backupgc.log.4" > nul 2>&1
      move /y "%JBOSS_LOG_DIR%\gc.log.*.current" "%JBOSS_LOG_DIR%\backupgc.log.current" > nul 2>&1

      "%JAVA%" -verbose:gc -Xlog:gc:"%JBOSS_LOG_DIR%\gc.log" -version > nul 2>&1
      if not errorlevel == 1 (
        if not exist "%JBOSS_LOG_DIR%" > nul 2>&1 (
          mkdir "%JBOSS_LOG_DIR%"
        )
      )
      set JAVA_OPTS=%JAVA_OPTS% -verbose:gc -Xlog:gc:"%JBOSS_LOG_DIR%\gc.log"
    )
  )
)
```

Configurar algoritmo del Garbage Collector

Se utilizará la configuración `-XX:+UseParallelGC` para solucionar un bug de la JVisualVM que muestra la memoria máxima al doble de lo que en realidad es (<https://github.com/oracle/visualvm/issues/127>).

En el caso de Tomcat se agregó en `CATALINA_OPTS` en el archivo **bin\setenv.bat**, mientras que para Wildfly se agregó en `JAVA_OPTS` en el archivo **bin\standalone.conf.bat**