

Performance Testing Oracle SOA Platform and Services

By Kathiravan Udayakumar

Performance Testing is key exercise that will help to measure the performance of the SOA Platform for a SOA Implementation to meet the non functional requirements set for the platform. Objective of the Performance Testing is to measure, test and ensure the below list of larger objectives are met.

1. Higher Throughput
2. Higher Reliability
3. Comply to Service Level Agreements
4. Better Response Time
5. Data Resiliency
6. Greater Stability

Oracle SOA Performance Testing can be broadly classified into two types, Oracle SOA Platform Performance Testing and other Oracle SOA Service Performance Testing. Methodology, Approach, Tools and techniques used for Platform Testing and Service Testing are different.

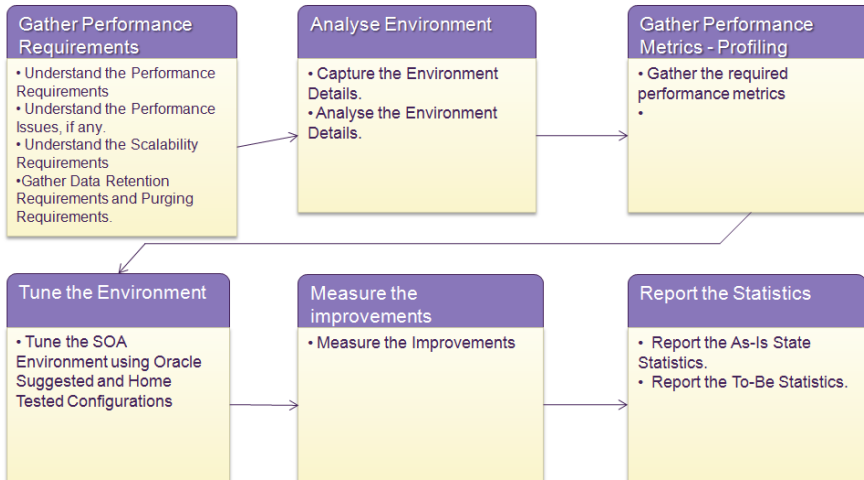
This article will discuss in detail about the below listed items

1. Oracle SOA Platform Performance Testing
 - a. Data Burst Utility
2. Oracle SOA Service Performance Testing

Oracle SOA Platform Performance Testing

The below stated approach should be followed in Testing the platform performance which includes,

1. Gathering Performance Requirements
2. Analysing the Environment Condition
3. Gathering the Performance Metrics (through Profiling)
4. Tune the Environment
5. Measure the Improvements
6. Report the Statistics



Gathering Performance Requirements:

Gathering Performance Requirements includes the below listed activities.

1. Understanding the Performance Requirements
2. Understand the Performance issues, if any
3. Understand the Scalability Requirements, if any
4. Understanding the Service Line Agreements
5. Understanding the audit retention and purging needs.

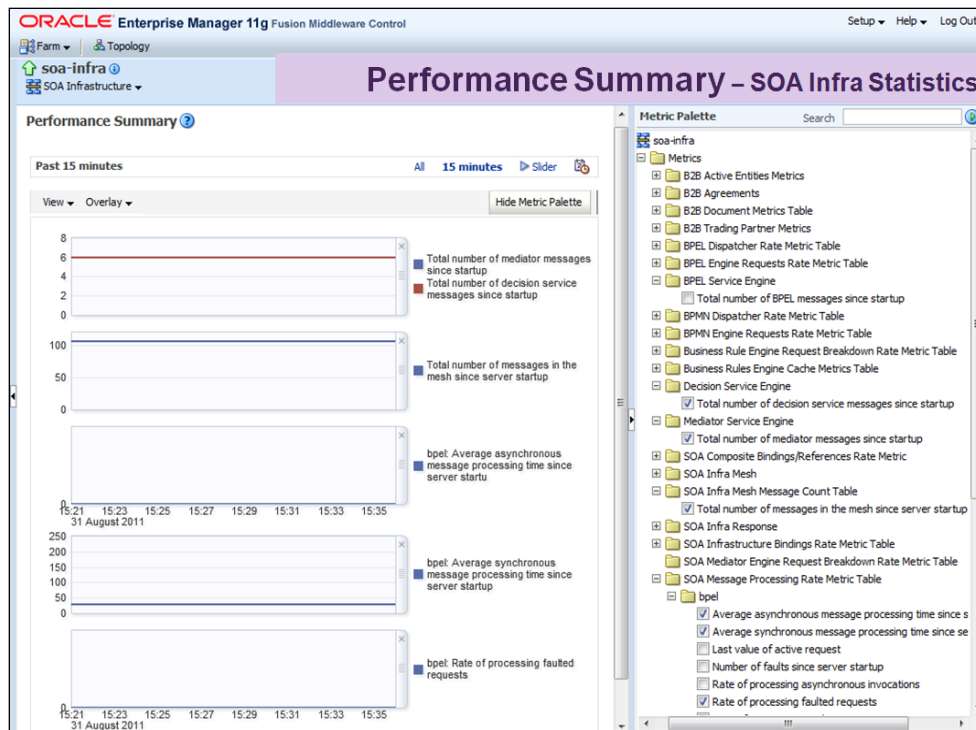
Analyse the Environment:

Analyse the Environment is a second step in Performance Testing Exercise where the details of the SOA Environments are captured and analysed. Typically this step would capture and gather below list of information.

1. SOA Installation Mode
2. OS and Hardware Configuration
3. Weblogic and SOA Cluster configuration
4. Database and Network Configuration.

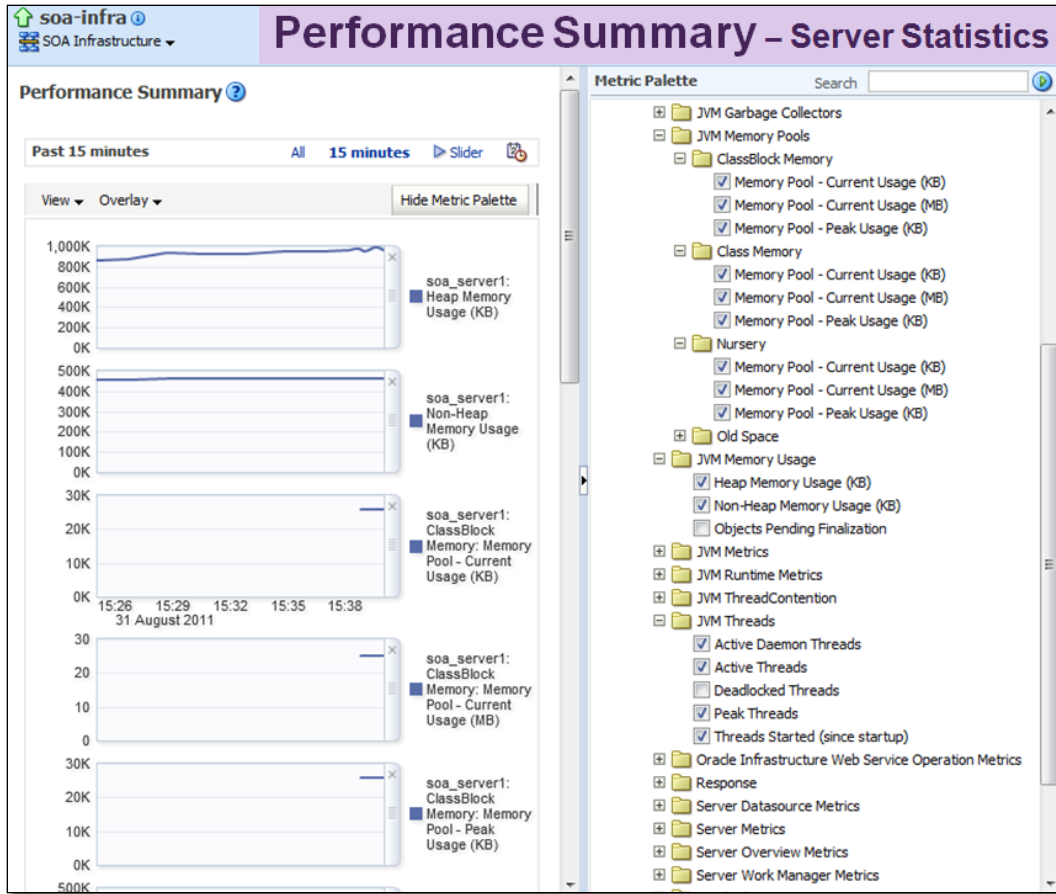
Gathering Performance Metrics:

Analyse the Environment is a second step in Performance Testing Exercise where the details of the SOA Environments are captured and analysed. Typically this step would capture and gather below list of information. Oracle Fusion Middleware console provides an option to collect various metrics of the SOA Environment.

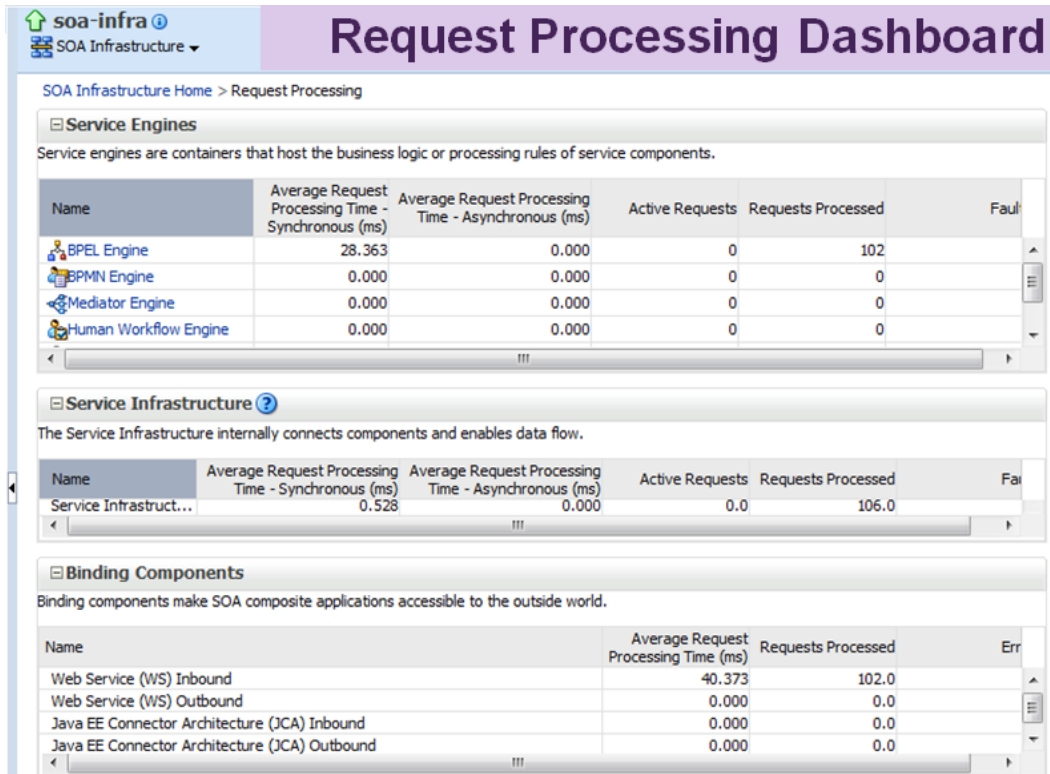


Performance of Hardware, Network, OS, JVM and Weblogic can be obtained using various tools and technique described in Chapter 9.

Oracle Fusion Middleware Console also provides an option to collect the JVM Performance using the Server Metrics Statistics collection agents as shown below.



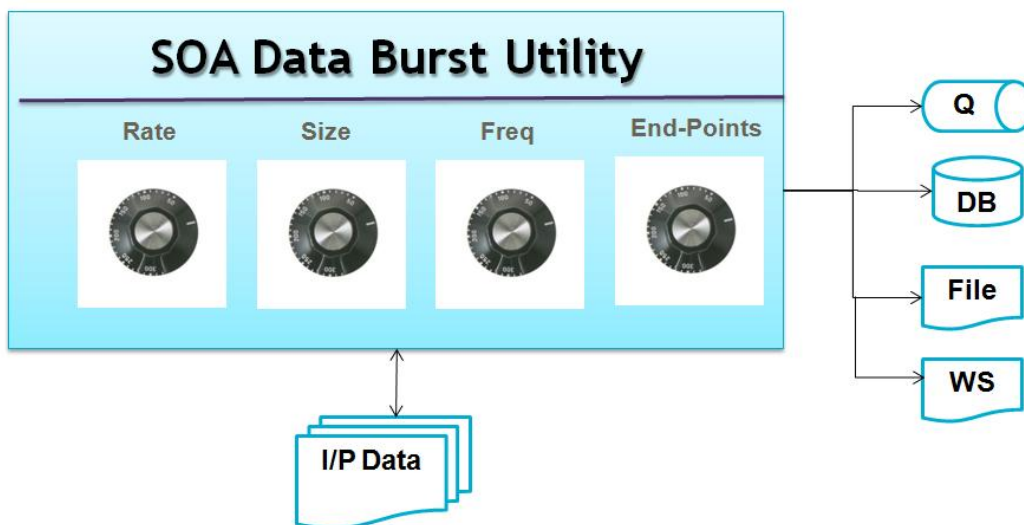
Various Engine Processing capabilities can be obtained using the Request Processing Dashboard Provided in SOA Administration -> Request Processing



Before we can proceed more on the Performance measurement or Performance Tuning it is essential to understand the technique to test the platform for larger objectives. When new Oracle SOA Environment are installed it may not be possible to understand/test the scalability capabilities of the platform using the tools or service provided in the Fusion Middleware console; it is essential to mimic the actual load of the platform. The following section of the article will discuss in detail about the SOA Data Burst Utility which will aid in generating the SOA Data Burst across the platform and help in generating or mimic the actual load of the platform.

SOA Data Burst Utility:

Objective of this tool is to generate the schedulable data burst at required Rate, Data Size to required Configurable End Points. This will help to mimic the real world performance load



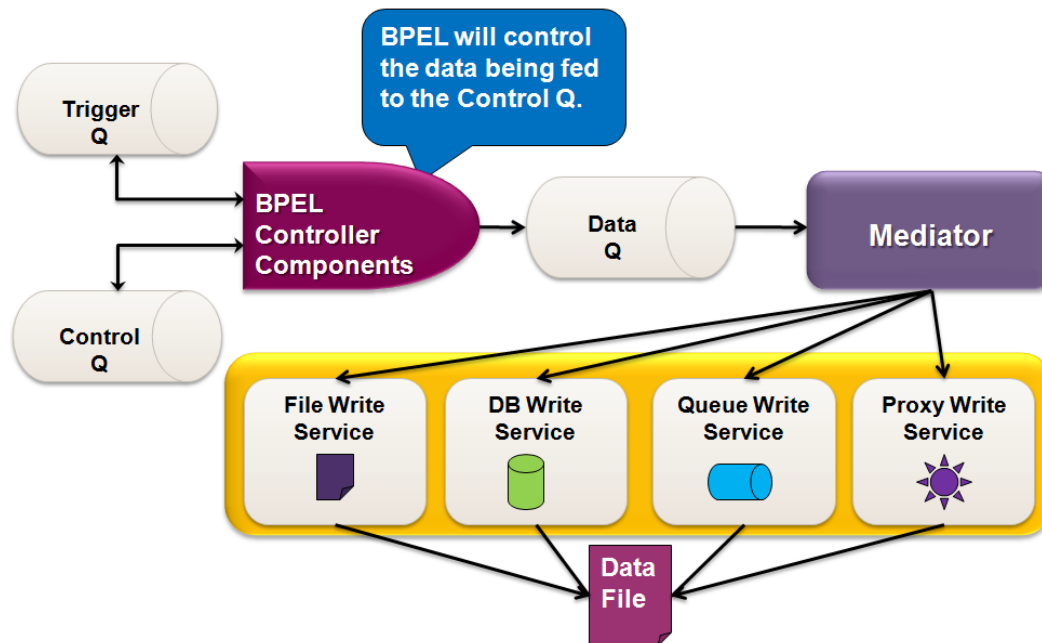
Features:

- SOA Data Burst Utility will be a SCA Composite Process that will read the data from the file and burst the data to the target systems/ end-points.
- 500 KB file will be read by the Data Burst Utility and Data will be pushed to required endpoints. Size of Data could be changed dynamically or during the trigger of the data burst component.
- Based on the data size set for the Channel; the data will be aggregated and it will be fed to the system.
- Utility will feed the required channels based on the channel settings which includes rate, frequency and end-points for the data feed.

Components & Implementation Details

Data Burst Utility should contain three major controller components.

1. Trigger Component
2. Controller Component
3. Data Burst Component
4. Mediator to route the data to required End-Points (Dynamic Service Call Framework implementation)



Trigger Component:

Trigger Component will read the Trigger Q and call the Controller Component with Required Details

Component Type: BPEL

Process Type: One-Way ASync

Input: Read from Trigger Q. *Input Message Structure is shown below.*

```

<TriggerBurst>
< BurstDetails >
    <Rate>5s</Rate>
    <SizeLevel>2</SizeLevel>
    <Period>1h</Period>
</ BurstDetails >
<EndPoints>
    <EndPoint @type="file">WSDLLocationOftheFilewriteService</EndPoint>
    <EndPoint @type="db"> WSDLLocationOftheDBwriteService </EndPoint>
    <EndPoint @type="q"> WSDLLocationOftheQwriteService</EndPoint>
    <EndPoint @type="ws"> WSDLLocationOftheProxyService</EndPoint>
</EndPoints>
</TriggerBurst>

```

Processing:

1. Call the Controller Component
2. Place the ControlMsg in the Control Q
3. End-the-Trigger.
4. Output: <None>

Controller Component:

Controller Component will be called from the Trigger Component. Controller component will keep placing the Control Message in Control Q until the Burst Period is over. Rate will determine the Rate in which the control message will be placed in Control Q.

Component Type: BPEL

Process Type: One-Way ASync

Input:

- Read from Control Q

```

<ControlMsg>
    <CorrelationID>SOA-ECID</CorrelationID>
<EndPoints>
    <EndPoint @type="file">WSDLLocationOftheFilewriteService</EndPoint>
    <EndPoint @type="db"> WSDLLocationOftheDBwriteService </EndPoint>
    <EndPoint @type="q"> WSDLLocationOftheQwriteService</EndPoint>
    <EndPoint @type="ws"> WSDLLocationOftheProxyService</EndPoint>
</EndPoints>
</ ControlMsg >

```

Processing:

- Place the ControlMsg in the Control Q
- Place the Data in Data Q. Data Q Schema is shown below.

```

< EndPointData>
<CorrelationID>SOA-ECID</CorrelationID>

```

```

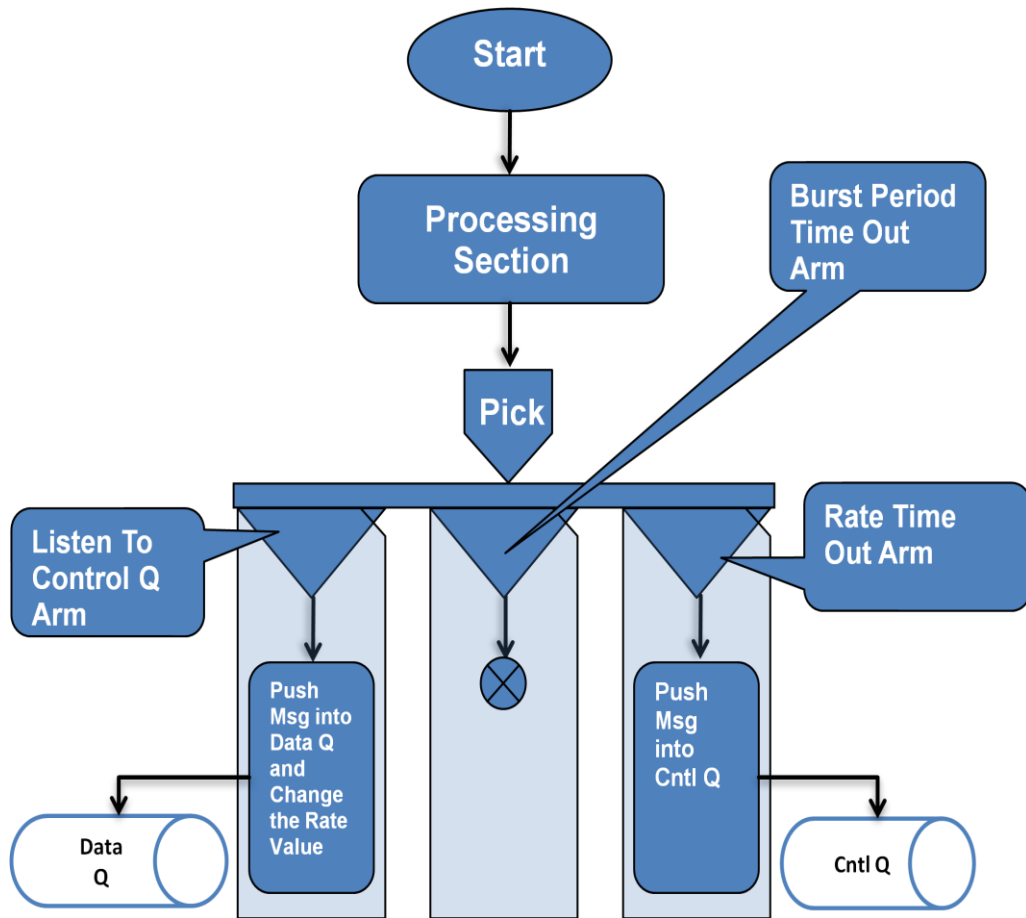
<EndPoint>
  <EndPoint @type="file">WSDLLocationOftheFilewriteService</EndPoint>
  <EndPoint @type="db"> WSDLLocationOftheDBwriteService </EndPoint>
  <EndPoint @type="Q"> WSDLLocationOftheQwriteService</EndPoint>
  <EndPoint @type="WS"> WSDLLocationOftheProxyService</EndPoint>
</EndPoint>
</ EndPointData>

```

Output:

- Metrics Collected in the Run.

Controller Component Implementation Template: Below Template can be utilized for the implementing the data burst utility.



Oracle SOA Service Performance Testing:

Oracle SOA Services can be tested from console and Oracle SOA Test Framework provide this option to test the performance of the service by using the Stress Test Functionality provided by the Oracle Fusion Middleware console.

Service syncbpel_client_ep

Port SyncBPEL_pt

Operation process

Endpoint URL http://localhost:7001/soa-infra/services/default/BPELExample1/syncbpel_client_ep

Request

Response

Security

Quality of Service

HTTP Transport Options

Additional Test Options

Enable Stress Test

Concurrent Threads 5

Loops per Thread 10

Delay in Milliseconds 1000

Input Arguments

Tree View

Name	Type	Value
* payload	payload	
* input	string	StressTest