Prepared by: Buzz IT Company Limited

Author: Tommy Leung

Date: 25/05/2017

Version: 0.1

Maxim’s POS Polling Enterprise Service Bus Implementation Service

**EL-FY16-902**

**Sales/Master/Pricing Data Exchange**

Performance Test Reports

**Table of Content**

[1 Document Control 3](#_Toc483476654)

[1.1 Document History 3](#_Toc483476655)

[1.2 Document/Design Owner 3](#_Toc483476656)

[1.3 Key Comments 3](#_Toc483476657)

[2 Background 3](#_Toc483476658)

[2.1 Document Purpose 3](#_Toc483476659)

[2.2 Document Audience 3](#_Toc483476660)

[2.3 Testing Objective 4](#_Toc483476661)

[2.4 Terminology Definition 4](#_Toc483476662)

[3 Performance Test Environment 5](#_Toc483476663)

[3.1 Network Diagram 5](#_Toc483476664)

[3.2 Hardware and Software Configuration 5](#_Toc483476665)

[3.3 Testing Tool 5](#_Toc483476666)

[4 Test Approach and Test Case 6](#_Toc483476667)

[5 Testing Results 7](#_Toc483476668)

[5.1 Case 1 – 10 Concurrent Process 7](#_Toc483476669)

[5.2 Case 2 – 50 Concurrent Process 8](#_Toc483476670)

[5.3 Case 3 – 100 Concurrent Process 9](#_Toc483476670)

[6 Conclusion 10](#_Toc483476671)

[7 Appendix 11](#_Toc483476671)

[7.1 Record Count on Each Polling Tables 11](#_Toc483476672)

# Document Control

## Document History

| Version | Date | Author | Revision Remark |
| --- | --- | --- | --- |
| 0.1 | 25/05/2017 | Tommy Leung | 1st Draft |

## Document/Design Owner

| Name | Title |
| --- | --- |
| Tommy Leung | System Analyst |

## Key Comments

| Name/Title |  | |
| --- | --- | --- |
| # | Comments |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Background

## Document Purpose

The purpose of the Performance Test report is to report the Performance Test result of Maxim’s ESB project.

## Document Audience

The audiences of this Performance Test report are the project owners and steering committee members to review Performance Test result. It is assumed that the project owners will refer the Performance Test result to determine whether to proceed the live of Maxim’s ESB system for sales data polling.

## Testing Objective

The performance testing approach will focus on the polling functions supported by the revamped ESB sales data polling implementation. Within the context of the performance testing engagement, we will:

• Focus on mitigating the performance risks for this revamped sales data polling implementation.

• Make basic working assumptions on which parts of the implementation need to be performance-tested.

• Reach consensus on these working assumptions and determine the appropriate level of performance and Performance Testing that shall be completed within this compressed time schedule.

## Terminology Definition

|  |  |
| --- | --- |
| **Terminology** | **Description** |
| EDW | Oracle Enterprise Data Warehouse |
| Staging | Staging Database to keep polling data |

# Performance Test Environment

## Network Diagram

N/A

## Hardware and Software Configuration

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Server** | **No. of CPU** | **RAM** | **Hard Disk** | **IP** | **OS** |
| 1 | **HKGESBAPPP01** - Production Oracle ESB Server 1 (Clustering) | ? | 48 GB | ? | 10.10.33.28 | ? |
| 2 | **HKGESBAPPP02** - Production Oracle ESB Server 2 (Clustering) | ? | 48 GB | ? | 10.10.33.29 | ? |
| 3 | **HKGDBAAGS01** - Performance Testing Database | 2 | 6GB | C: 100 DB  D: 400 DB  E: 200 DB  F: 40 DB  G: 600 DB | 10.10.31.85 | OS: Windows 2012 R2 / Ent.  DB: SQL Server 2016 Ent. SP1 |
| 4 |  |  |  |  |  | OS: Windows XP |

## Testing Tool

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Tool Name** | **Function and Description** | **Vendor** | **Version** |
| 1 | VisualVM | Monitor Tools to Monitor Web Logic Activities | Freeware Sponsored by Oracle | 1.39 |

# Test Approach and Test Case

As ESB polling data from remote sites and it take some time to generate bulk amount of data. In other words, it is not a determine factor in server loading. Besides, it is observed that data store in DBF and CSV is not significant compared to the data store in SQL Server. Therefore, in this performance test, we created large amount of records to a dummy SQL Server database installed in Windows XP. We count on the time required to pull all data from remote POS client to staging database and EDW respectively.

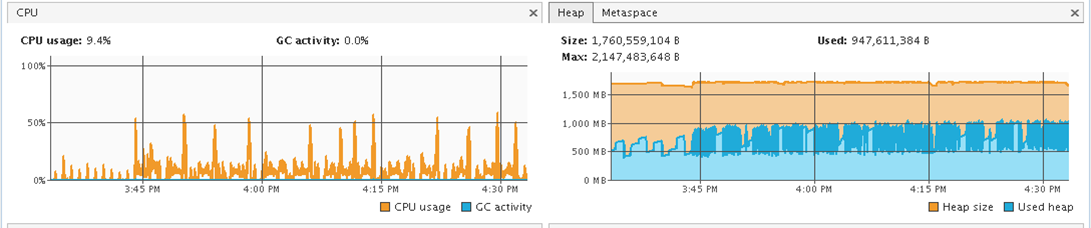
Concurrent process count, on the other hand, is another criterial factor to test in this performance test. Therefore, the project team also adjust the number of concurrent processes to run in order to get the optimized performance.

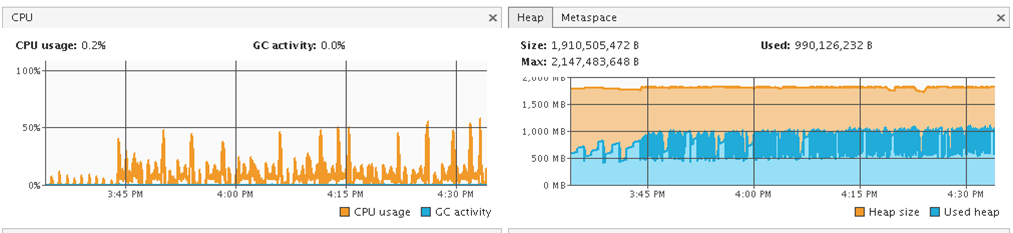
|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Number of**  **Client** | **Parallel**  **Process** | **Remarks** |
| 1 | 300 SQL Branch  40 TXT Branch  25 DBF Branch | 10 x 2 | 2 machines running in parallel and for each machine configure to run 10 concurrent processes. |
| 2 | 300 SQL Branch  40 TXT Branch  25 DBF Branch | 50 x 2 | 2 machines running in parallel and for each machine configure to run 50 concurrent processes. |
| 3 | 300 SQL Branch  40 TXT Branch  25 DBF Branch | 100 x 2 | 2 machines running in parallel and for each machine configure to run 100 concurrent processes. |

# Testing Results

## Case 1 – 10 X 2 Concurrent Process

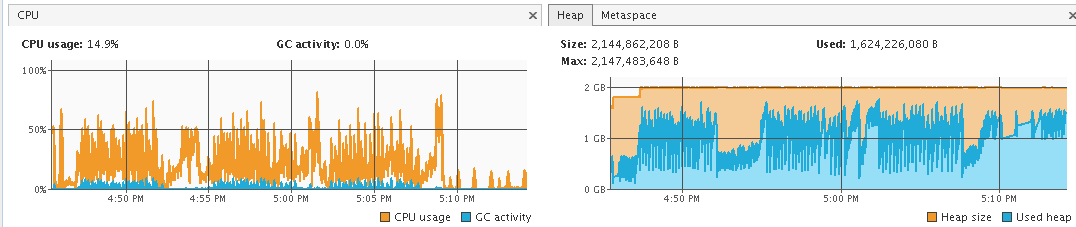
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Number of**  **Client** | **Parallel**  **Process** | **Upload to Staging**  **(second)** | **Upload to EDW**  **(second)** | **Total**  **(second)** | **All Client**  **(second)** | **All Clients**  **(min)** |
| 300 | 10 x 2 | 105 | 20 | 125 | 3000 | 50 |

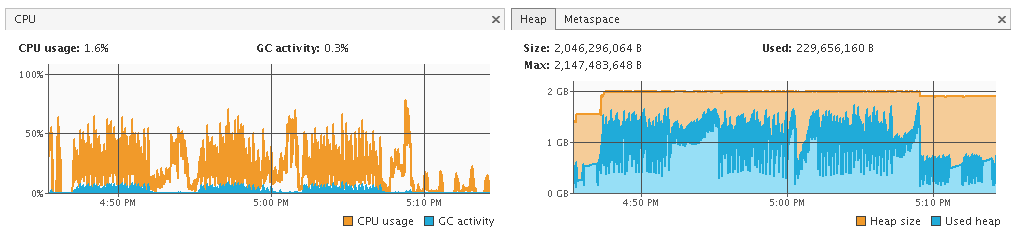




## Case 2 – 50 X 2 Concurrent Process

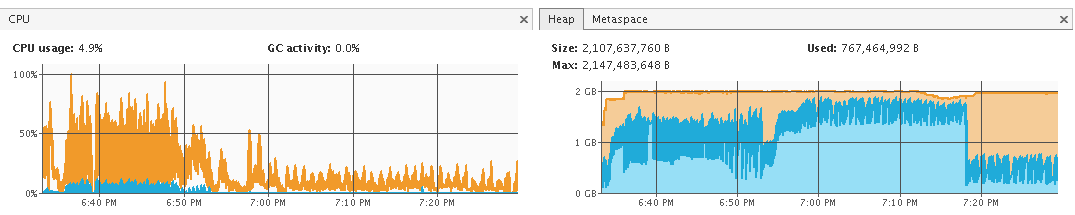
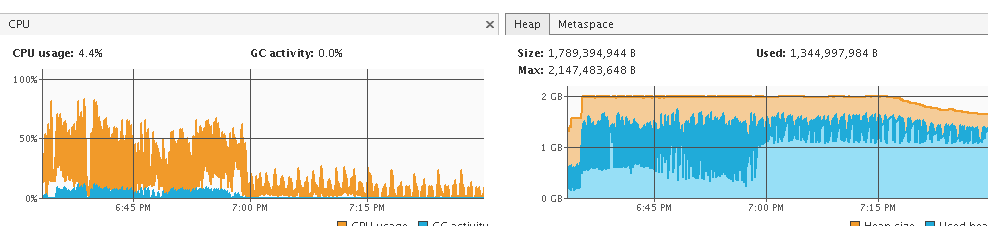
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Number of**  **Client** | **Parallel**  **Process** | **Upload to Staging**  **(second)** | **Upload to EDW**  **(second)** | **Total**  **(second)** | **All Client**  **(second)** | **All Clients**  **(min)** |
| 300 | 50 x 2 | 300 | 60 | 360 | 1380 | 23 |





## Case 3 – 100 X 2 Concurrent Process

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Number of**  **Client** | **Parallel**  **Process** | **Upload to Staging**  **(second)** | **Upload to EDW**  **(second)** | **Total**  **(second)** | **All Client**  **(second)** | **All Clients**  **(min)** |
| 300 | 100 x 2 | 600 | 90 | 690 | 1500 | 25 |



# Conclusion

* The system can support up to 100 parallel process in each machine and run healthily
* The process time between 50 and 100 parallel process does not have a significant different.
  + The bottleneck should be on the SQL Database staging server.
* Suggest to have initial setup to 50 parallel process in each machine.
* Suggest to set the JVM to (min 1024, max 2048) in each application server.
* The system can finish the follow data with 30 minutes
  + 300 branches and each branch have overs 100000 records (Total over 30,000,000 records)
  + Each Branch have over 30000 record in hist\_tran (Total over 9,000,000 records)
* By Projection, we can support existing data volume and future 5-year data volume
* The Stress Test Setup for POS is setup in GIGA-Lan, in real case it may take a longer time to proceed.
* The Stress Test is done in Stress Test SQL Server and production will have a more power machine for SQL Server
* The Stress Test is done in same Weblogic Environment as Production.

# Appendix

## Record Count on Each Polling Tables

| **Table Name** | **No. of Record** |
| --- | --- |
| hist\_check\_logs | 11596 |
| hist\_coupon\_sales | 0 |
| hist\_item | 11267 |
| hist\_itemstock | 11267 |
| hist\_orders | 14797 |
| hist\_orders\_extra | 4 |
| hist\_orders\_pay | 15219 |
| hist\_orders\_pay\_progress | 9 |
| hist\_payfig | 23 |
| hist\_paysum | 238 |
| hist\_possystem | 11 |
| hist\_redeemed\_coupon | 32 |
| hist\_stock\_movement | 201 |
| hist\_supp | 11310 |
| hist\_trans | 32158 |
| hist\_trans\_ecard | 678 |
| hist\_trans\_modifier | 0 |
| Total: | 108810 |

- End -