HP Business Analytics

For the Linux ® **Operating System**

Software Version: 10.10

Performance Benchmark Document

Document Release Date: Feb 2016

Software Release Date: Feb 2016

Table of Contents

1. Introduction

1.1 Purpose and Scope

This document outlines the performance benchmark for HP IT Business Analytics (ITBA) v10.10, and highlights the performance improvements compared with the previous release v9.50

Generally, ITBA v10.10 has the same performance as v10.00 for legacy Content Packs, KPIs and Metrics, and UI. Additional benchmark tests were created for the new Content Packs (SA, AM, and more) that are supported by ITBA v10.10.

The scenarios used in this document follow the critical paths that every ITBA customer goes through: DCS, ETL, KPI Calculation, and UI operations.

Note that all the tests are conducted at HP Software R&D labs.

1.2 Environment Setup

The following tables list the properties of the machines in use:

Configuration 1: Single Node Vertica Setup[1]						
Server	**VM**	**CPU**	**Memory**	**Disk**	**Network**	**OS**
ITBA	1	8 cores, 2.13 GHz	16 GB	100 GB	1 GB	Red Hat Linux 6.5
Vertica	1	8 cores, 2.13 GHz	16 GB	100 GB	1 GB	Red Hat Linux 6.5

Configuration 2: 3-Nodes Vertica Setup (ksafe1)[2]						
Server	**VM**	**CPU**	**Memory**	**Disk**	**Network**	**OS**
ITBA	1	8 cores, 2.13 GHz	16 GB	100 GB	1 GB	Red Hat Linux 6.5
Vertica	3	8 cores, 2.13 GHz	16 GB	100 GB	1 GB	Red Hat Linux 6.5

[1] Configuration 1 is referred as [single setup] in this document.
[2] Configuration 2 is referred as [cluster setup] in this document

For the cluster setup, Native Connection Load Balancing is enabled. It is a feature built into the Vertica server and client libraries. A host in the Vertica cluster can redirect a client attempt to another currently-active host in the mix, and this redirection is based on ROUNDROBIN policy. According to the Vertica Administrator Guide, in most situations, it is the right choice, as it is easy to set up, less at risk for host failures, uses less memory and CPU, and is supported by HPE.

2. Scenarios

2.1 DCS & ETL

In this scenario, we perform end-to-end DCS & ETL operations for the Project and Portfolio Management (PPM), Application Lifecycle Management (ALM), Cloud Service Automation (CSA), and Service Manager (SM) data sources, Server Automation (SA), and Asset Management (AM). All the sources are restored from customer dumps. It's acknowledged that data distribution often has a certain impact on the performance, so by using customer dumps we ensure that the test results reflect real world situations.

We have replaced the heavy weight BODS in v9.50 with a lightweight SQL-based workflow in v10.10/v10.00. Generally speaking, for DCS, the load is on the ITBA server, that is, multiple threads are leveraged to extract the source data from the database or the web server, while, for ETL, the load is on the Vertica nodes(s) as multiple entities are processed in parallel.

PPM

In comparison with v9.50, the size of the flat-file is smaller, the requirement of the database storage is lower, and the total time cost is reduced significantly for both DCS and ETL. With these two PPM sources: Small and Large, it is proved that the resource consumption of the operating system as well as the performance grow linearly in sync with the size of the content pack.

<table>
<tr><th>PPM: Small</th><th colspan="2">v10.10/v10.00</th><th colspan="2">v9.50</th></tr>
<tr><td rowspan="10">Initial load, ~1.21 m rows in target tables</td><td colspan="4">Flat-File Size</td></tr>
<tr><td colspan="2">0.57 GB</td><td colspan="2">1.68 GB</td></tr>
<tr><td colspan="4">Database Storage Requirement Per Node</td></tr>
<tr><td colspan="2">2+ GB</td><td colspan="2">~15 GB</td></tr>
<tr><td>DCS Time Cost</td><td>ETL Time Cost</td><td>DCS Time Cost</td><td>ETL Time Cost</td></tr>
<tr><td colspan="4">Single Setup</td></tr>
<tr><td>1+ mins</td><td>6+ mins</td><td>4 mins</td><td>102 mins</td></tr>
<tr><td colspan="4">Cluster Setup</td></tr>
<tr><td>1+ mins</td><td>11+ mins</td><td>n/a</td><td>n/a</td></tr>
</table>

PPM: Large	v10.10/v10.00		v9.50	
Initial load, ~3.91 m rows in target tables	**Flat-File Size**			
	2.89 GB		7.79 GB	
	Database Storage Requirement Per Node			
	6+ GB		~75 GB	
	DCS Time Cost	**ETL Time Cost**	**DCS Time Cost**	**ETL Time Cost**
	Single Setup			
	5+ mins	21+ mins	13 mins	242 mins
	Cluster Setup			
	5+ mins	41+ mins	n/a	n/a

ETL in a cluster setup is also much slower than in a single setup. Clusters take care of the high availability of the underlying databases, which essentially means that in a 3-nodes cluster even if one node goes down, the load can be forwarded to the other nodes that are still up and running. This is critical in production environments to avoid system downtime. However, this advantage comes at a cost, that is, each load of the data to the database has to be duplicated **ksafe** time(s) (in our cluster setup, it is set to 1) to populate its main and buddy projections. If our cluster setup is configured with no backup, ETL time cost is close to that of a single setup, but be careful as this type of usage is NOT recommended according to Vertica Support.

ALM

ALM shows similar performance improvement for both DCS and ETL. However, source extraction is still the bottleneck of the scenario, and it's most likely caused by the network transmission speed. We sampled 20 packets, 2 k in size each, against our remote ALM web server, the average round trip took ~241.103 ms; while for a local web server or database, it took ~4 ms. Therefore, if the customer uses a local ALM web server, DCS should be much faster.

Note: The data in the table below is based on 5 ALM projects.

ALM	v10.10/v10.00		v9.50	
Initial load, ~0.36 m rows in target tables	**Flat-File Size**			
	0.41 GB		n/a	
	Database Storage Requirement Per Node			
	2+ GB		n/a	
	DCS Time Cost	**ETL Time Cost**	**DCS Time Cost**	**ETL Time Cost**
	Single Setup			
	40+ mins	4+ mins	81 mins	29 mins
	Cluster Setup			
	41+ mins	8+ mins	n/a	n/a

CSA

CSA shows similar performance improvement for ETL. However, based on the DCS time cost, the source extraction is shown as the bottleneck. The round trip of 20 2 m packets between the ITBA server and the local CSA web server took ~4 ms. Further analysis of the CSA extractor indicates that the execution of the CSA REST request is extremely time-consuming. Therefore, unless CSA enhances the API, this bottleneck can hardly be avoided.

<table>
<tr><th>CSA</th><th colspan="2">v10.10/v10.00</th><th colspan="2">v9.50</th></tr>
<tr><td rowspan="10">Initial load, ~3.73 m rows in target tables</td><td colspan="4">Flat-File Size</td></tr>
<tr><td colspan="2">0.15 GB</td><td colspan="2">n/a</td></tr>
<tr><td colspan="4">Database Storage Requirement Per Node</td></tr>
<tr><td colspan="2">1+ GB</td><td colspan="2">n/a</td></tr>
<tr><td>DCS Time Cost</td><td>ETL Time Cost</td><td>DCS Time Cost</td><td>ETL Time Cost</td></tr>
<tr><td colspan="4">Single Setup</td></tr>
<tr><td>283+ mins</td><td>9+ mins</td><td>330 mins</td><td>30 mins</td></tr>
<tr><td colspan="4">Cluster Setup</td></tr>
<tr><td>286+ mins</td><td>15+ mins</td><td>n/a</td><td>n/a</td></tr>
</table>

SM

SM shows similar disk usages reduction and performance improvement for both initial load and 1% delta load. Surprisingly, unlike other content packs, its cluster setup is almost as fast as its single setup.

<table>
<tr><th>SM</th><th colspan="2">v10.10/v10.00</th><th colspan="2">v9.50</th></tr>
<tr><td rowspan="9">Initial load, ~13.95 m rows in target tables</td><td colspan="4">Flat-File Size</td></tr>
<tr><td colspan="2">1.00 GB</td><td colspan="2">3.49 GB</td></tr>
<tr><td colspan="4">Database Storage Requirement Per Node</td></tr>
<tr><td colspan="2">9+ GB</td><td colspan="2">~75 GB</td></tr>
<tr><td>DCS Time Cost</td><td>ETL Time Cost</td><td>DCS Time Cost</td><td>ETL Time Cost</td></tr>
<tr><td colspan="4">Single Setup</td></tr>
<tr><td>10+ mins</td><td>26+ mins</td><td>20 mins</td><td>453 mins</td></tr>
<tr><td colspan="4">Cluster Setup</td></tr>
<tr><td>10+ mins</td><td>29+ mins</td><td>n/a</td><td>n/a</td></tr>
</table>

<table>
<tr><th>SM</th><th colspan="2">v10.10/v10.00</th><th colspan="2">v9.50</th></tr>
<tr><td rowspan="8">1% Delta load, ~14.05 m rows in target tables</td><td colspan="4">Flat-File Size</td></tr>
<tr><td colspan="2">14.1 MB</td><td colspan="2">n/a</td></tr>
<tr><td colspan="4">Database Storage Requirement Per Node</td></tr>
<tr><td colspan="2">< 1GB</td><td colspan="2">n/a</td></tr>
<tr><td>DCS Time Cost</td><td>ETL Time Cost</td><td>DCS Time Cost</td><td>ETL Time Cost</td></tr>
<tr><td colspan="4">Single Setup</td></tr>
<tr><td>10+ mins</td><td>3+ mins</td><td>19 mins</td><td>56 mins</td></tr>
<tr><td colspan="4">Cluster Setup</td></tr>
</table>

	10+ mins	3+ mins	n/a	n/a

AM

The below table shows the DCS and ETL performance of a particular AM source.

AM	v10.10/v10.0		v9.50	
Initial load, ~5.83 m rows in target tables	Flat-File Size			
	424 MB		n/a	
	Database Storage Requirement Per Node			
	3GB		n/a	
	DCS Time Cost	ETL Time Cost	DCS Time Cost	ETL Time Cost
	Single Setup			
	1 mins	10+ mins	n/a	n/a
	Cluster Setup			
	1 mins	13+ mins	n/a	n/a

SA

In v10.10, ITBA extracts the SA source data directly from the database instead of using the API as done before v9.50. However, DCS still takes more than 1 hour with high volume data because it takes time to parse xml data into DCS steps. In delta load, the DCS and the ETL have excellent performance.

SA	v10.10		v9.50	
Initial load, ~82.04 m rows in target tables	Flat-File Size			
	2.57GB		n/a	
	Database Storage Requirement Per Node			
	15GB		n/a	
	DCS Time Cost	ETL Time Cost	DCS Time Cost	ETL Time Cost
	Single Setup			
	80+ mins	62+ mins	n/a	n/a
	Cluster Setup			
	82+ mins	95+ mins	n/a	n/a

SA	v10.10		v9.50	
Delta load, ~82.82 m rows in target tables	Flat-File Size			
	22 MB		n/a	
	Database Storage Requirement Per Node			
	< 1GB		n/a	
	DCS Time Cost	ETL Time Cost	DCS Time Cost	ETL Time Cost
	Single Setup			
	<1 min	7+ mins	n/a	n/a
	Cluster Setup			
	<1 mins	12+ mins	n/a	n/a

2.2 KPI Calculation

In this scenario, we performed a series of KPI Engine calculations using a simple formula with various breakdowns, calculation types, and calculation periods. As a result, we observed that:

1> *Comparing with v9.50, we have achieved a certain level of performance improvement*.

2> The total time cost grows linearly in sync with the increase of calculation units. Generally speaking more breakdowns and calculation periods lead to more calculation units.

3> With a large number of distinct values in breakdown(s) there is no consecutive load on the Vertica database, the stress is all on the Postgres database. With a smaller number of distinct values, the load goes back to the Vertica database, as there are less historical results to be processed.

KPI Definition	Breakdown Details
Source: PPM, Small customer dump Context: ProjectPortfolioManagement Formula: COUNT(ProjectTask , *)	Breakdown Case #1 ➔ ProjectTask: OverdueTaskIndicator ➔ Project: Class ➔ Person: Name (x125)
	Breakdown Case #2 ➔ ProjectTask: OverdueTaskIndicator ➔ Project: Class ➔ Person: Name (x25)
	Breakdown Case #3 ➔ ProjectTask: OverdueTaskIndicator ➔ Project: Class ➔ Person: Name (x5)
	Breakdown Case #4 ➔ ProjectTask: OverdueTaskIndicator ➔ Project: Class
	Breakdown Case #5 ➔ ProjectTask: OverdueTaskIndicator

Breakdown Case #	Calculation Type	Calculation Period	Calculation Units		Total Time Cost (s)	
			v10.10/v10.00 Single Setup	v9.50	v10.10/v10.00 Single Setup	v9.50
1	Recalculate, 1year	Yearly	1518	1578	35	49
1	Recalculate, 1 year	Quarterly	3795	3945	92	133

1	Recalculate, 1 year	Monthly	9867	10257	249	365
1	Recalculate, 1 year	Weekly	40227	43558	1064	1506
1	Recalculate, 1 year	Daily	277794	287985	8034	9840
1	Recalculate, 2 years	Daily	554829	575240	16173	19665
5	Recalculate, 2 years	Daily	2924	2920	647	1011
4	Recalculate, 2 years	Daily	10965	10950	928	1547
3	Recalculate, 2 years	Daily	39474	39420	1612	3050
2	Recalculate, 2 years	Daily	127925	133590	3038	5487
1	Calculate Now	Daily	759	790	17	22

For a cluster setup, it shows very similar numbers.

Breakdown #	**Calculation Type**	**Calculation Period**	**Calculation Units**		**Total Time Cost (s)**	
			V10.10/v10.00 Cluster Setup	**v9.50**	**V10.10/v10.00 Cluster Setup**	**v9.50**
1	Recalculate, 1year	Yearly	1518	1578	41	49
1	Recalculate, 1 year	Quarterly	3795	3945	97	133
1	Recalculate, 1 year	Monthly	9867	10257	261	365
1	Recalculate, 1 year	Weekly	40227	43558	1068	1506
1	Recalculate, 1 year	Daily	277794	287985	7770	9840
1	Recalculate, 2 years	Daily	554829	575240	15348	19665
5	Recalculate, 2 years	Daily	2924	2920	716	1011
4	Recalculate, 2 years	Daily	10965	10950	1057	1547
3	Recalculate, 2 years	Daily	39474	39420	1847	3050
2	Recalculate, 2 years	Daily	127925	133590	3133	5487
1	Calculate Now	Daily	759	790	16	22

2.3 UI Operations

In this scenario, we simulated a large number of concurrent users viewing pages and exploring the web portal using Load Runner [v12.50]. Note that we do not use the data from any of those customer dumps, as in v9.50. This test is designed using a demo CAP, so in order to draw a fair comparison with the previous release, we stick to the CAP data.

Viewing a BOE report was excluded in this test, as its behavior depends on the performance of a 3rd party library, its weight was assigned to other UI operations accordingly.

The following table presents the workflow transactions, their weights in percentage, and corresponding simulated steps:

Transaction	Weight, %	Simulated Steps
TX_BA_open_VPOps_Main_double	10%	Open a page with four components: 1) Scorecard 2) KPI View (8 KPIs) 3) Historical View (3 KPIs) 4) KPI Rolodex (15KPIs)
TX_BA_open_VPOps_8_components	10%	Open a page with eight components: 1) Scorecard 2) KPI View (4 KPIs) 3) Historical View (3 KPIs) 4) KPI Rolodex (8KPIs) 5) KPI List (7 KPIs) 6) Historical Metric View (1 Metric) 7) Pie Chart View (2 KPIs) 8) KPI View (7 KPIs)
TX_BA_open_VPOps_Main	10%	Open a page with four components: 1) Scorecard 2) KPI View (5 KPIs) 3) Historical View (1 KPIs) 4) KPI Rolodex (6KPIs)
TX_BA_refresh_VPOps_Main	7%	Click the Refresh button on the page TX_BA_open_VPOps_Main
TX_BA_view_Reduce_Risk	10%	Simulate the user action that displays the information while setting the cursor on the shortcut of a KPI or Objective
TX_BA_explore_Reduce_Cost	10%	Click the shortcut link of a KPI in a page to open Explorer
TX_BA_view_Met_SLAs	10%	Simulate the user action that displays the information while setting the cursor on the shortcut of a KPI or Objective

TX_BA_explore_Affected	10%	Click the shortcut link of a KPI in a page to open Explorer
TX_BA_change_to_current	7%	Change the Period to current in Explorer
TX_BA_dataset	1%	Click the Data Set button in Explorer
TX_BA_goalmap	5%	Click the Goal Map button in Explorer
TX_BA_forecast	5%	Click the Forecast button in Explorer
TX_BA_annotation	5%	Add and Delete annotation while viewing the KPI details in the Dashboard Page

When comparing with v9.50, we can support more concurrent users with larger TTPS. Note that the test results exclude the ramp-up and ramp-down periods to reflect only the peak load.

	Total Transactions Per Second, Pass	**Total Transactions Per Second, Fail**	**Concurrent Users[3]**	**Think Time, Second**
v10.10/v10.00, Single Setup	5.15	0.001	300	60 (90% - 120%)
v10.10/v10.00, Cluster Setup	5.10	0.003	300	60 (90% - 120%)
v9.50	1.00	< 0.01	100	60 (90% - 120%)

The following tables list the details for both single setup and cluster setup. Although both can support up to 300 concurrent users and 5+ TTPS, cluster setup is slightly slower than single setup in terms of Avg. and 90% response time, that is, some cross-node queries should be optimized in future releases.

v10.10/v10.00, Single Setup	**Min.**	**Avg.**	**Max.**	**Std. Dev.**	**90%**	**Pass**	**Fail**	**Stop**
TX_annotation	0.0	0.1	2.9	0.1	0.2	926	4	0
TX_change_to_current	0.1	0.2	3.3	0.1	0.3	1301	0	0
TX_dataset	0.5	0.6	1	0.1	0.8	157	0	0
TX_explore_Affected	0.3	0.4	2.9	0.1	0.4	1886	0	0
TX_explore_Reduce_Cost	0.6	0.9	4.9	0.2	1.0	1828	0	0
TX_forecast	0.2	0.5	1.4	0.2	0.7	935	0	0
TX_goalmap	0.0	0.1	3.1	0.1	0.1	924	0	0
TX_open_VPOps_Main	0.7	1.5	7.7	0.6	2.4	1924	0	0

[3] Concurrent Users represents the number of users currently logged in the system. It may be inactive due to the think time configuration. All these users are not logged in and out of the system simultaneously, they are instead controlled by user ramp-up settings: one every 00:00:05 (HH:MM:SS); and user ramp-down settings: one every 00:00:05 (HH:MM:SS).

TX_refresh_VPOps_Main	0.1	0.3	1.9	0.1	0.5	1307	0	0
TX_view_Met_SLAs	0.1	0.2	3.9	0.1	0.2	1804	0	0
TX_view_Reduce_Risk	0.1	0.1	5.4	0.2	0.1	1785	0	0
TX_open_VPOps_8_components	1.4	3.0	12.4	1.0	4.9	1825	0	0
TX_open_VPOps_Main_double	0.6	1.7	5.1	0.6	2.6	1874	0	0

v10.10/v10.00, Cluster Setup	**Min.**	**Avg.**	**Max.**	**Std. Dev.**	**90%**	**Pass**	**Fail**	**Stop**
TX_annotation	0.0	0.2	0.3	0.0	0.2	895	10	0
TX_change_to_current	0.1	0.2	0.7	0.1	0.3	1236	0	0
TX_dataset	0.9	1.2	2.0	0.2	1.5	181	0	0
TX_explore_Affected	0.3	0.4	1.3	0.1	0.5	1926	0	0
TX_explore_Reduce_Cost	0.7	1.0	1.6	0.1	1.1	1750	0	0
TX_forecast	0.2	0.7	2.2	0.2	0.9	859	0	0
TX_goalmap	0.0	0.1	0.2	0.0	0.1	927	0	0
TX_open_VPOps_Main	0.8	2.0	7.7	1.0	3.4	1808	0	0
TX_refresh_VPOps_Main	0.1	0.4	1.8	0.2	0.5	1323	0	0
TX_view_Met_SLAs	0.1	0.2	0.4	0.0	0.2	1872	0	0
TX_view_Reduce_Risk	0.1	0.1	0.2	0.0	0.2	1828	0	0
TX_open_VPOps_8_components	1.5	4.0	14.4	1.7	6.7	1919	0	0
TX_open_VPOps_Main_double	0.7	2.1	8.8	1.0	3.4	1877	0	0

3. Conclusion

Performance was one of the high priority projects in the v10.10/v10.00 release. We expended considerable efforts optimizing the performance, as well as making a significant move by shifting to the Linux operating system and leveraging the world-famous real-time analytics platform, Vertica. This document proves that we took the right decision: faster with less hardware requirements.

4. Appendix I – v9.50 Environment Setup

Configuration: Distributed Setup						
Server	**Model**	**CPU**	**Memory**	**Disk**	**Network**	**OS**
ITBA	VM	8 cores, 2.67 GHz	16 GB	100 GB	1 GB	Windows 2008 R2
BOE	VM	8 cores, 2.67 GHz	16 GB	100 GB	1 GB	Windows 2008 R2
DWH	VM	8 cores, 2.67 GHz	16 GB	100 GB	1 GB	Windows 2008 R2

SQLServer	ProLiant DL580	24 Cores 2.93GHz	64 GB	1 TB	1 GB	Windows 2008 R2

5. Appendix II – Configuration References

In the v10.10/v10.00 Support Matrix, there are two set of configurations, minimum and recommended. In this performance benchmark we adopted the recommended one with 16 GB physical memory. However, there are differences in terms of application performance among these configurations based on the results of additional tests. This section can be used as a guideline for capacity planning.

Configuration	CPU (Cores)		Memory (GB)	
	ITBA	**Vertica**	**ITBA**	**Vertica**
Minimum	4	4	12	12
Recommended, 1	8	8	16	16
Recommended, 2	8	8	16	24

DCS and ETL

On an ITBA server, no extractor can utilize more than 50% of the processing power of the 8-cores system, so as expected, a 4-cores system will not affect its performance. A smaller number of cores for the Vertica server means lower concurrency, and more entities queued up for processing, which in turn slows down the ETL process. For example, the initial load of PPM-Large takes 21+ minutes with an 8-cores configuration; while it increases to 27+ minutes with a 4-cores configuration. Similarly for the initial load of SM: with an 8-cores configuration it spends 30+ minutes; while with a 4-cores configuration it goes up to 35+ minutes. Also note that for small content packs or content pack with a very skewed data distribution, that is, a few large entities, and the rest of the entities being much smaller, the impact is trivial.

We support running multiple content packs concurrently. In general, the 12 GB physical memory for the Vertica server is sufficient for the initial load of 1 to 2 content packs; the 16 GB physical memory for the Vertica server is sufficient for the initial load of 2 to 3 content packs. However, if additional content packs are required, and there are entities with millions of rows, we suggest to upgrade the physical memory to at least 24 GB.

KPI Calculation

It is expected that there are no significant differences among these configurations.

UI Operations

A smaller number of cores for the ITBA server results in a smaller number of concurrent users, 150; lower TTPS, 2.6. If there are more concurrent users logging on and browsing the portal, both the avg. and 90% response time increase dramatically, which is often the indicator of a bad end user experience.