Oracle Multitenant on Fujitsu M10 Server

A Study of Database Consolidation and Expansion using Fujitsu M10 Building Blocks and Pluggable Databases

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

Table of Contents

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

1. Introduction

The business environment has been changing fast and dramatically. Information and Communications Technology (ICT) not only responds to the changing business climate but, in many cases, drives the change. There is an urgent need to lower the capital and operating costs of ICT. At the same time, there is a growing need to invest in cloud-type ICT platforms that are capable of quickly providing new generations of services directly connected to the business.

In this era of Big Data, businesses are being defined with the data they acquire and use. Databases are at the core of business operations. The flexibility and scalability to provide database services with ease and speed when starting a new business project or service are crucial. As new services proliferate across the enterprise, so do the databases. For business to flourish the sprawl of databases and servers needs to be tamed through consolidation for a reduction in capital and operating costs.

Organizations have approached database consolidation at different levels of their ICT architecture: with virtualized environments, with database instances, or with database schemas. Each of these methods has its merits, benefits, as well as compromises. The introduction of Oracle Database 12*c*, introduces a new approach to database management and consolidation with new Pluggable Databases and Oracle Multitenant capabilities.

Consolidation with Pluggable Databases using Oracle Multitenant allows organizations to reach higher levels of consolidation ratios and scalability with powerful manageability capabilities. Database consolidation not only becomes significantly easier with Oracle Database 12*c* compared to conventional methods, it also becomes possible to deploy new databases extremely rapidly and to increase flexibility to support new business services and applications.

It can be quite challenging to provide a server infrastructure to support these promising new database management and consolidation capabilities in fast-changing business environments while lowering risks and costs. One option is to deploy a large capacity server provisioned for the predicted future workload requirements. As hard as it would be to predict the future, paying for the future use in advance would not be financially attractive. The other option is to deploy new servers to replace existing ones as the business grows. This too would be a costly option with implications on business continuity at the crucial time of rapid business growth.

To address this dilemma and avoid excessive risks and costs, Fujitsu developed the Fujitsu M10 server that can flexibly support business growth with drastically lower upfront investment and without impacting business continuity. Fujitsu M10 server infrastructure can be built modularly with the Building Block architecture. Fujitsu M10 can grow in increments of one single CPU core with the "CPU Core Activation" feature. The Physical Partition Dynamic Reconfiguration capability of Fujitsu M10 brings Building Blocks together into a single system and enables dynamic growth; one Building Block at a time and one CPU core at a time.

Individually, Oracle Database 12*c* and Fujitsu M10 each reduces the customer's capital and operating costs and provides flexible scalability and growth. When deployed together, Oracle Database 12*c* running on the Fujitsu M10 platform, they can deliver the most efficient and flexible database server environment.

This document shows how Pluggable Databases and hardware resources can be added and removed, without stopping servers or services, by using the Oracle Multitenant features of Oracle Database 12*c* and the CPU core activation and Dynamic Reconfiguration features of Fujitsu M10-4S. This document also provides proof that, in the process of expanding the database server, the transaction processing capacity increases linearly while the response time remains constant.

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

2. Oracle Database 12*c* Multitenant

2.1 Product Overview

The Oracle Multitenant[1] option for Oracle Database 12*c* is an innovative, in-database virtualization solution for consolidation of Oracle 12*c* databases that reduces capital and operational IT costs. As depicted in **Figure 1** below, the innovation is driven by efficient use of the container database (CDB) memory and background processes to provide measurable efficiencies and greater consolidation density of distinct pluggable databases (PDBs). The combination of hosted PDB tenants in the CDB Root is the Multitenant Container Database.

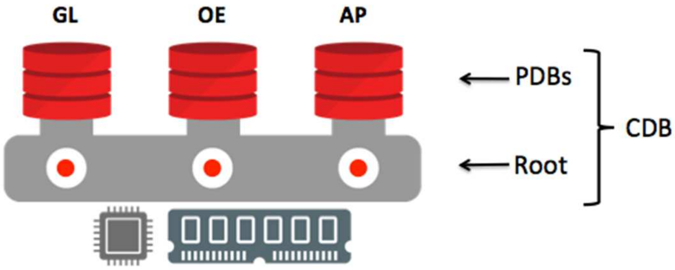

Figure 1. Oracle Multitenant Container Database

Each PDB is a portable, self-contained database with its own schemas, schema objects, and non-schema objects stored in the individual PDB SYSTEM, SYSAUX, and USER tables. Each PDB has its own set of data files distinct from the CDB, which allows for easy unplug/plug database mobility and rapid PDB provisioning through cloning. The root container database (CDB$ROOT) provides shared database dictionary structures, SGA memory and background processes.

Oracle Multitenant fully complements other Oracle Database options including Real Application Clusters (RAC), Oracle Data Guard and Oracle GoldenGate. As described in this document, the combination of Oracle Multitenant, RAC and the dynamic Capacity On Demand (COD) capabilities of the Fujitsu M10 server illustrates the ease of rapid, online database provisioning of PDBs through online core activation and dynamic CPU resource changes to the CDB initialization parameter. A simple SQL statement provisions additional PDBs. The dynamic building block approach found in Fujitsu M10-4S is an efficient, cost effective deployment model, which complements the high-density consolidation benefits of Oracle Multitenant.

Additional practical benefits of this architecture, as a result of efficient database consolidation are:

- **Simplified database management through Manage Many-as-One**
  The Oracle Multitenant architecture offers coarse grained database management as a *common user* executing across all or a subset of hosted PDBs. Simplified management tasks, for example, would be to patch the CDB and all hosted PDBs in a single patching operation or to back up the CDB and all hosted PDBs as a single RMAN backup. When required, management isolation can be restricted to individual PDBs through *local users*.

- **Agile data mobility through database unplug/plug**
  Oracle Multitenant in-database virtualization allows for distinct PDB tablespaces and data files. Through simple SQL commands, PDBs can relocate from one hosting CDB to another.

- **Fast thin provisioning at the SQL command line or the PDB Self-Service Application**
  Oracle Multitenant supports thin provisioning of PDBs, as well as full provisioning. File system storage that supports copy-on-write such as ASM Cloud File System (ACFS) enables users to snapshot clone PDBs from a simple SQL statement. Additional PDB cloning

[1] For a complete description of Oracle Multitenant, please see http://oracle.com/goto/Multitenant

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

functionality includes subset cloning of specific schemas, meta-data cloning and remote cloning from a source database.

- **Existing databases can be easily adopted as a PDB with no application changes**
  From an application perspective, Oracle Multitenant looks and feels like a traditional non-Container database (non-CDB). No application changes are required.

Oracle Multitenant on the Fujitsu M10 reduces capital expense through efficient consolidation and capacity on demand provisioning enabling a greater density of applications per server when you need it. Oracle Multitenant reduces operational costs through standardization; rapid database provisioning through simple SQL or the PDB self-service provisioning application; and the inherent Manage Many-as-One CDB/PDB framework for patching, upgrades and backups.

2.2 Multitenant Use Cases on the Fujitsu M10

The principal use case for Oracle Multitenant is consolidation. IT departments are reducing energy, maintenance and licensing costs by consolidating servers and software. Coupled with consolidation comes standardization. Further reductions in IT costs are achieved in efficient, low risk, automated operational provisioning and maintenance practices. Oracle Multitenant satisfies both of these requirements. These are the first, necessary steps when transitioning to a service based architecture and a cloud deployment model. Oracle Multitenant, by design, is the cornerstone of Oracle **Database as a Service** (DBaaS). And by implementing multitenancy in the database and not in the application, Oracle Multitenant is perfect for **Software as a Service** (SaaS). Cloud Database life cycle management through orchestration becomes easier with Oracle Multitenant for both on-premise private cloud as well as private cloud - public cloud interoperability.

DBaaS

With Oracle Multitenant, customers are defining their DBaaS, electing to provision PDBs for each level of service. For example, as depicted in **Figure 2** below, customers may define a DBaaS catalog based on tiered availability service levels. PDBs can be provisioned at each tier and, leveraging the unplug/plug mobility of PDBs, can easily navigate between the service levels as application availability requirements change.

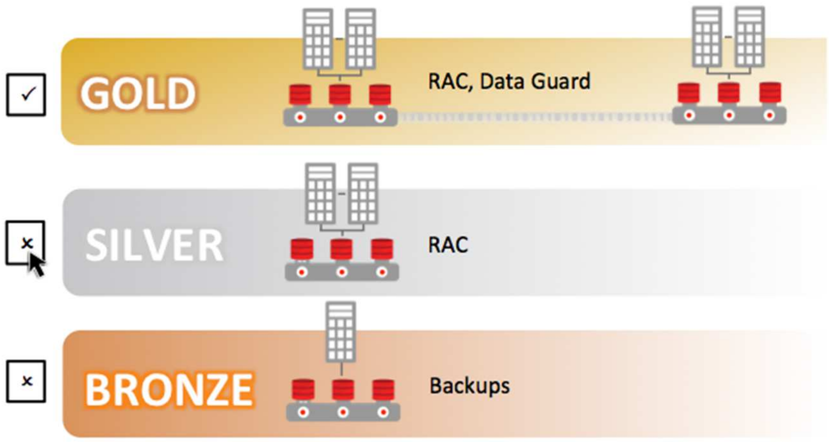

Figure 2. Oracle Multitenant for Database as a Service

An added efficiency of Oracle Multitenant and PDBs is that when a CDB is configured for either RAC or Data Guard, plugging in a PDB from a single instance CDB in to a CDB configured for RAC or Data Guard Standby, it automatically becomes a multi-instance RAC PDB or, the case of Data Guard, redo for the PDB is automatically replicated to the standby with no configuration changes necessary. Because the APIs for unplug/plug service level migration are simple SQL commands, such operations integrate seamlessly with any service orchestration framework.

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

SaaS

Implementing multitenancy at the application level can be difficult where data isolation, database security and a common reporting tool set are required. Deploying SaaS applications on Oracle Multitenant satisfies each of these requirements. Each pluggable database is physically isolated from other tenants. If further isolation is required Oracle security products such as Database Vault and Label Security can be configured with Oracle Multitenant.

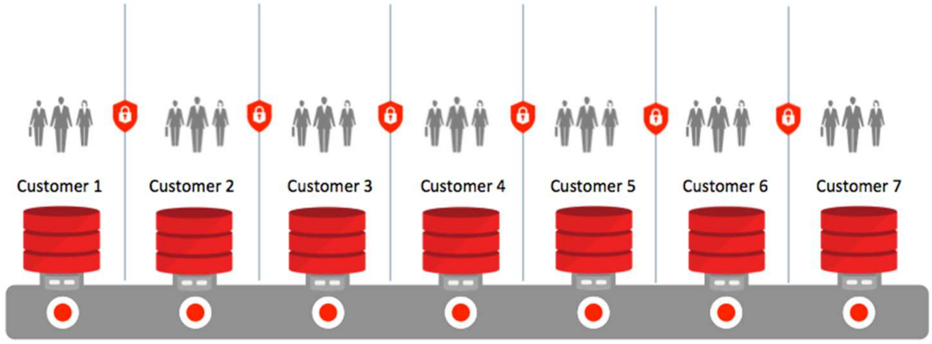

Figure 3. Oracle Multitenant for Software as a Service

With SaaS applications it is likely the applications share a common data model. Oracle Multitenant provides a fast query operation for reporting purposes across all PDBs or a subset of PDBs by executing select statements with the CONTAINERS() clause. This combination of distinct, isolated pluggable database with the ability to execute queries across all or a subset of pluggable database at scale is a unique feature of Oracle Multitenant.

Cloud Scale Computing

Oracle Multitenant and the building block architecture and dynamic core activation of the Fujitsu M10 are a powerful combination for private or public cloud deployments where DBaaS and SaaS capacity requirements expand and contract on demand and where scale-out database deployments may grow to 100s if not 1000s of instances. This study illustrates the agility of these two solutions to meet online capacity demands while achieving extreme consolidation, increasing aggregate transaction volume without sacrificing application performance.

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

3. Fujitsu M10 Server

3.1 Product Overview

Fujitsu M10 is a SPARC server that provides world-record performance and high reliability for mission-critical systems such as databases, and offers unique scalability and flexibility with innovative cutting-edge technical capabilities adopted from mainframes, supercomputers, and the previous generation SPARC Enterprise M-Series servers.

The Fujitsu M10 series lineup consists of three models:

- Fujitsu M10-1: the optimum entry-level server for any application that requires high performance, high reliability, and virtualization in a compact 1U form factor
- Fujitsu M10-4: the ideal mid-range server for data center consolidation
- Fujitsu M10-4S: a scalable high-end server with the unique ability to grow modularly by connecting "Building Blocks" with Fujitsu's innovative high-speed interconnect.

All Fujitsu M10 models support CPU core activation for step-by-step expansion of CPU resources in single core increments. Fujitsu M10-4S, which can grow step-by-step from small to very large (up to 16 Building Blocks), provides dynamic flexibility through its Physical Partition Dynamic Reconfiguration capability.

With three model types (**Figure 4**) and features that deliver flexible growth of resources, Fujitsu M10 enables customers to make investments that match their business growth now, while also providing room to grow without costly downtime.

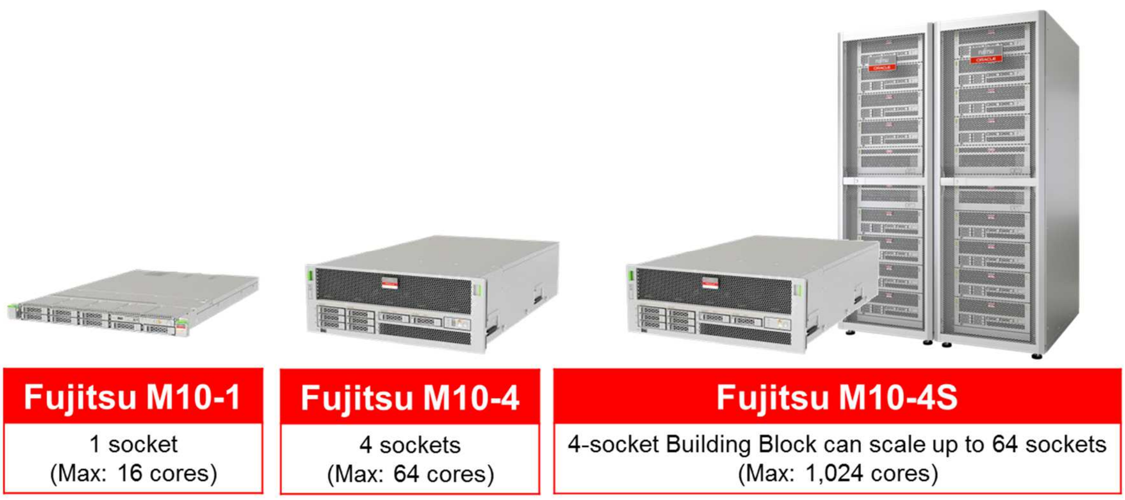

Figure 4. Fujitsu M10 server family lineup

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

3.2 CPU Core Activation

All Fujitsu M10 models support CPU core activation, which allows you to increase CPU performance gradually, in response to increased business volume, without stopping the system.

Fujitsu M10 provides flexibility for handling sudden load increases, and scalability that can be fine-tuned to suit any business need. Initial investment costs are kept low, and the customer's Total Cost of Ownership (TCO) is optimized dynamically as compute requirements increase. The fine grained granularity of CPU core activation provides TCO benefits not only at initial purchase. The core activation features also provides power and support savings.

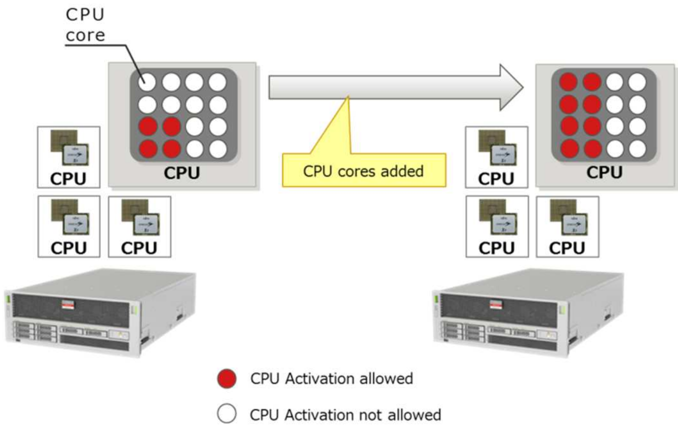

Figure 5. Adding CPU cores through CPU core activation

3.3 Building Block Architecture

Fujitsu M10-4S benefits from the unique Building Block architecture to grow chassis-by-chassis. From as little as two or four CPU sockets in a single M10-4S chassis, the system can grow all the way to 64 sockets in an interconnected 16 Building Block configuration. The high scalability of Fujitsu M10-4S is achieved by connecting Building Blocks with Fujitsu’s proprietary high-speed interconnect and allows you to flexibly create hardware partitions of various sizes. Each physical partition can be used as a physically isolated, highly-reliable system environment.

Fujitsu M10-4S Physical Partitions (PPARs), adds an additional level of flexibility, isolation and reliability. All Fujitsu M10 models support firmware level virtualization with Oracle VM Server for SPARC and software level virtualization with Oracle Solaris Zones. With these choices of virtualization technologies, Fujitsu M10 resources can be efficiently utilized across a wide range of workloads. In addition, consolidation of a large number of diverse systems on to Fujitsu M10 becomes possible and leads directly to lower TCO.

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

3.4 Physical Partition Dynamic Reconfiguration

Physical Partition Dynamic Reconfiguration (PPAR DR) is employed as a standard feature of the Fujitsu M10-4S scalable system at no additional cost. Through Fujitsu M10's system monitoring facility (XSCF), PPAR DR enables addition and removal of hardware resources such as processors, memory, and I/O devices in Physical Partitions without stopping the production system. (**Figure 6**).

PPAR DR makes it possible to change configuration, add capacity (active expansion), and actively perform hardware maintenance on mission-critical systems in a timely manner to meet the business demands for scalability and growth.

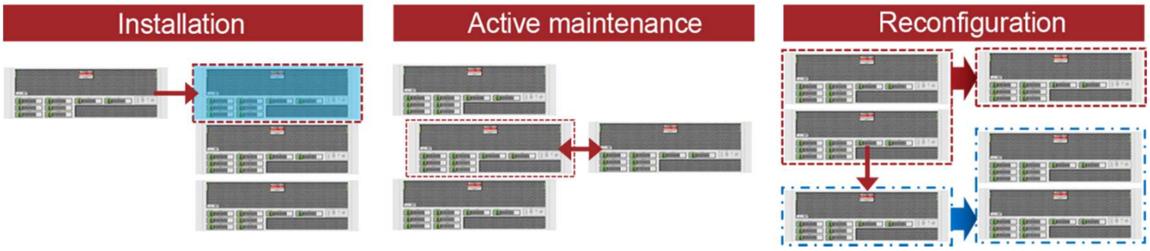

Figure 6. Various use case scenarios for Dynamic Reconfiguration

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

4. Verification

As described above, Oracle Database 12*c* Pluggable Databases and Fujitsu M10 hardware resources can be added and removed dynamically using the Oracle Multitenant features of Oracle Database 12*c* and the CPU core activation and Dynamic Reconfiguration features of Fujitsu M10-4S. Fujitsu offers the following detailed study as verification that, in the process of expanding the database server, the transaction processing capacity increases linearly while the response time remains constant.

4.1 Verification Environment

In this verification test, a Fujitsu M10-4S server (Oracle Solaris 11.1) with Oracle Database 12*c* is used as the database server, a Fujitsu ETERNUS DX410 S2 storage system is used for disk storage, and a Fujitsu PRIMERGY RX200 S7 server with Oracle Load Testing Accelerator for Oracle Database is used as the load generator.

4.1.1 Hardware and Software Used

Database Server

Fujitsu M10-4S (8 Building Blocks)

Processor: 3.7GHz SPARC 64X+, 4 sockets (16 cores per socket)

Memory: 1TB

Operating System: Oracle Solaris 11.1.17.5.0

Storage System

ETERNUS DX410 S2 (4 units)

HDD: 300GB (15,000rpm)

I/O: 8Gbps FC

Oracle Database

Oracle 12.1.0.1.4

Network

Cluster interconnect Network switch: CISCO Catalyst 4900M

Local Area Network switch: Fujitsu SR-X324T1

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

4.1.2 System Configuration

The database server consists of two Oracle Real Applications Cluster nodes. (**Figure 7**)

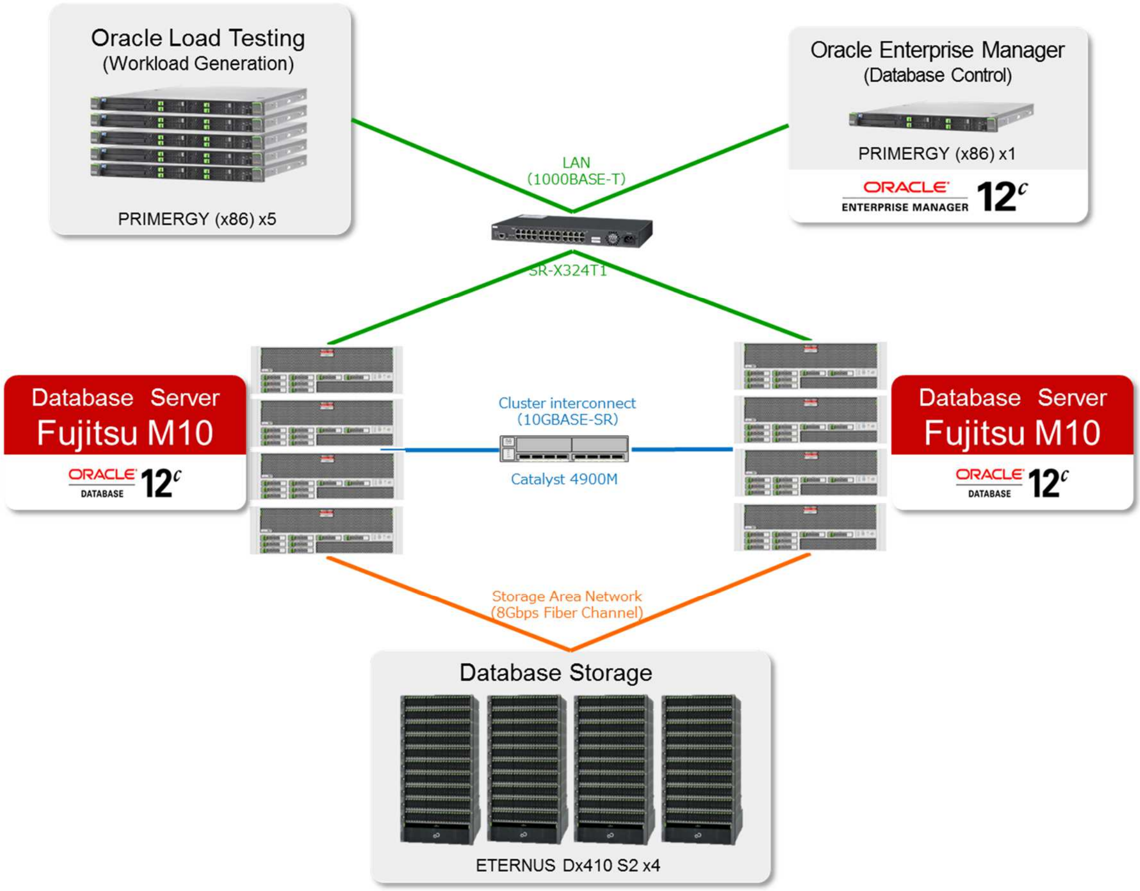

Figure 7 System configuration diagram

4.1.3 Storage Settings

Oracle Automatic Storage Management and Oracle 12*c* Pluggable Database are constructed as described below.

Storage volume:	62TB (total) = 1.3TB/LUN x 48
ASM Disk Group structure:	S.A.M.E (Stripe And Mirror Everything)
	All of the disks are arranged in one DISK Group (**Figure 8**)
Stored data:	Database files, REDO Log

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

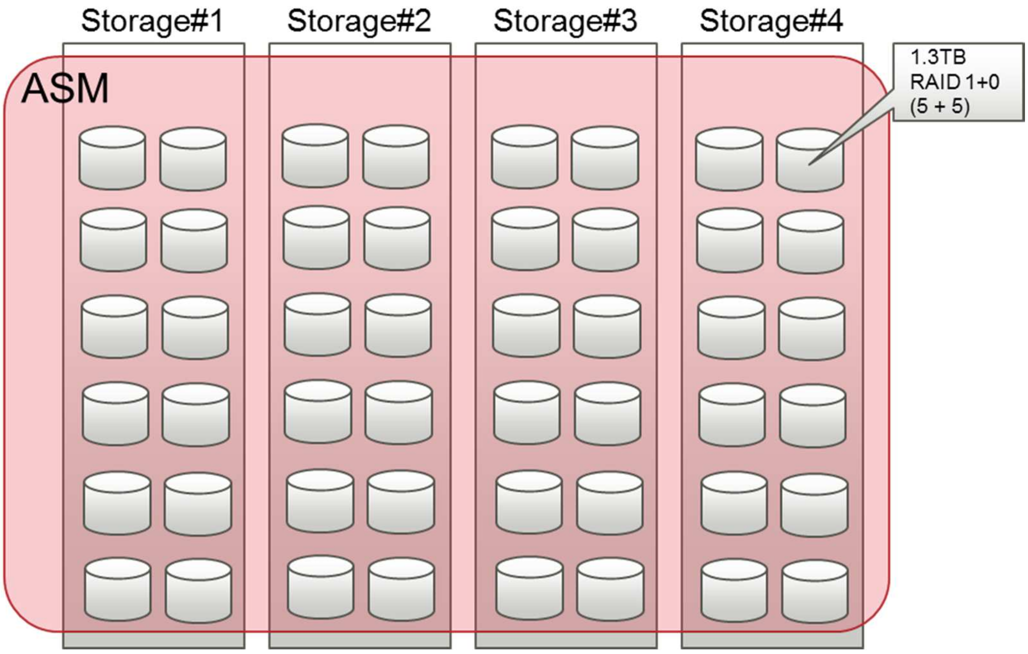

Figure 8. RAID and ASM disk group configuration diagram

For the pluggable databases, 252 disks are created in the ASM, and 10GB is allocated per PDB. (**Figure 9**) 4 CPU cores are assigned per PDB, and 10 users connect to each PDB.

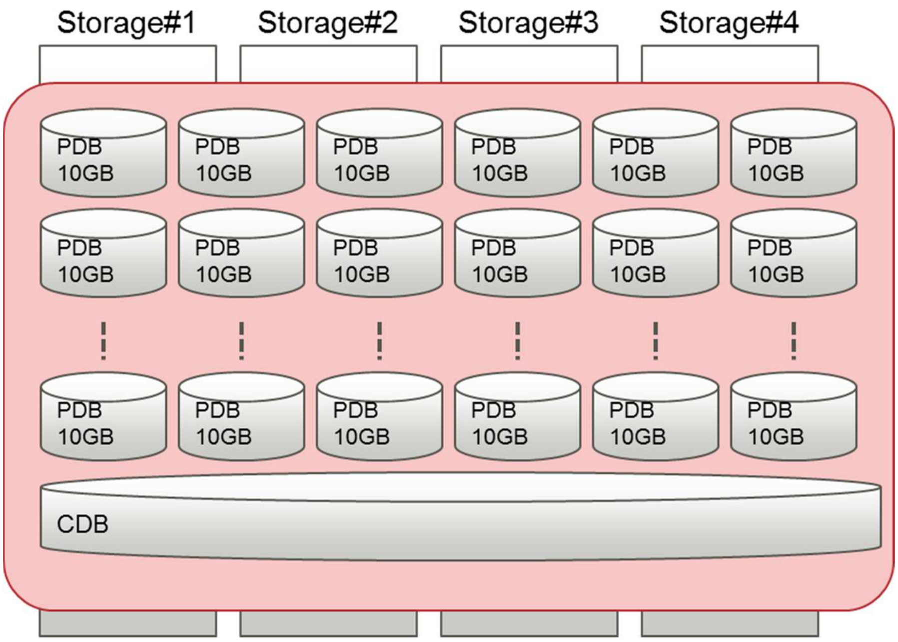

Figure 9. PDB configuration diagram

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

4.1.4 Testing Tools

Oracle Load Testing Accelerator (OLT) for Oracle Database is used. An OLTP model (Online Shopping – JpetStore -) is used for the work load.

No Web server or application server are used.

To increase load, a LOOP process and SELECT process are executed every time a trigger is fired and INSERT is executed.

4.2 Verification Details

The following sections describe the methodologies used to expand both the Fujitsu M10 computer resources and the Oracle Database 12*c* Pluggable Database environments during the verification testing.

4.2.1 Adding Resources through CPU Core Activation

In step with database expansion, hardware resources (CPU cores) are added utilizing the Fujitsu M10 CPU core activation feature.

Adding CPU cores on a running Fujitsu M10-4S is accomplished as shown below.

Confirm the number of CPU cores that are active from the Fujitsu M10-4S service processor (eXtended System Control Facility: hereafter XSCF).

```
XSCF> showcodusage
Resource In Use Installed CoD Permitted Status
-------- ------ --------- ------------- ------
PROC         16        64            16 OK: 0 cores available

PPAR-ID/Resource In Use Installed Assigned
---------------- ------ --------- --------------
0 – PROC             16        64       16 cores
Unused – PROC         0         0        0 cores
```

In the example above, there is one physical partition in one Fujitsu M10-4S Building Block, and 16 CPU cores have been assigned. The remaining 48 cores have not been activated.

In this verification test, hardware resources will be added to match the expansion of the database, so a CPU core activation keys will be added to activate additional CPU cores.

```
XSCF>addcodactivation "Product: SPARC M10-4S
SequenceNumber: 116
Cpu noExpiration 48
Text-Signature-SHA256-RSA2048:
SBxYBSmB32E1ctOidgWV09nGFnWKNtCJ5N3WSlowbRUYlVVySvjncfOrDNteFLzo
:
:
1TSgrjnee9FyEYITT+ddJQ=="
Above Key will be added, Continue?[y|n]:y
```

In the example above, 48 additional CPU cores were activated.

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

Confirm the CPU cores were activated.

```
XSCF> showcodusage
Resource In Use Installed CoD Permitted Status
-------- ------ --------- ------------ ------
PROC         16        64            64  OK: 48 cores available

PPAR-ID/Resource In Use Installed Assigned
---------------- ------ --------- -------------
0 – PROC             16        64       16 cores
Unused – PROC         0         0       48 cores
```

Assign CPU cores to physical partitions.

```
XSCF> setcod -p 0 -s cpu 64
```

Confirm that they have been assigned.

```
XSCF> showcodusage
Resource In Use Installed CoD Permitted Status
-------- ------ --------- ------------ ------
PROC         64        64            64 OK: 0 cores available

PPAR-ID/Resource In Use Installed Assigned
---------------- ------ --------- -------------
0 – PROC             64        64       64 cores
Unused – PROC         0         0       0 cores
```

Next, add CPU cores to the logical domain in which the Oracle database is running.

Execute the ldm list-devices command on the logical domain in which the Oracle database is running. Confirm the status of the added hardware resources, then execute the ldm add-core command and add CPU cores to the logical domain in which the Oracle database is running.

```
# ldm list-devices
CORE
   ID          %FREE          CPUSET
   128         100            (256, 257)
   132         100            (264, 265)
   136         100            (272, 273)
   140         100            (280, 281)
....
MEMORY
   PA                                         SIZE                  BOUND
   0x700000000000                             32G
   0x720000000000                             32G
   0x740000000000                             32G
   0x760050000000                             31488M
....

# ldm add-core 48 OracleDB-domain
```

The fact that CPU cores have been assigned to a physical partition without suspending system operation has been confirmed.

Repeat the same procedure when adding CPU cores, as needed.

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

4.2.2 Adding Resources through PPAR DR

If further hardware resources are required after activating all CPU cores inside an M10-4S Building Block, additional Building Blocks are added through PPAR DR. (**Figure 10**)

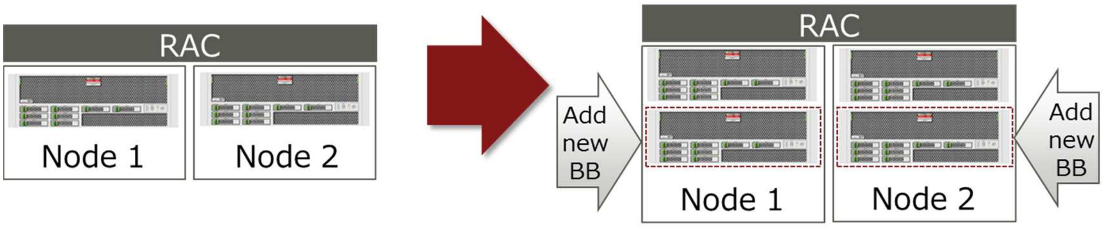

Figure 10. Adding Fujitsu M10-4S and PDB through PPAR DR

The steps for adding Fujitsu M10-4S Building Blocks through PPAR DR are shown below.

Confirm the current physical partition configuration information from the Fujitsu M10-4S XSCF.

```
XSCF> showpcl -p 0
PPAR-ID    LSB    PSB     Status
00                        Running
           00     00-0
```

Register the to-be-added Fujitsu M10-4S Building Block to the physical partition configuration information.

```
XSCF> setpcl -p 0 -a 1=01-0
```

Confirm the physical partition configuration information.

```
XSCF> showpcl -p 0
PPAR-ID    LSB    PSB     Status
00                        Running
           00     00-0
           01     01-0
```

Check the status of the Fujitsu M10-4S.

As shown below, the newly added Fujitsu M10-4S Building Block is displayed as "01-0" in the system board pool status.

```
XSCF> showboards -p 0
PSB  PPAR-ID(LSB)  Assignment  Pwr  Conn Conf Test    Fault
---- ------------ ----------- ---- ---- ---- ------- --------
00-0 00(00)        Assigned    y    y    y    Passed  Normal
01-0 SP            Available   n    n    n    Passed  Normal
```


Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

Add the newly registered Fujitsu M10-4S Building Block (01-0) to the physical partition that is running.

```
XSCF> addboard -c configure -p 0 01-0
PSB#01-0 will be configured into PPAR-ID 0. Continue?[y|n] :y
Start connecting PSB to PPAR. [3600sec]
0..... 30..... 60..... 90.. ...120.....150.....180.....210.....240.....
270.....300.....330.....360.....390.....420.....450.....480.....510.....
540.....570.....600.....630.....660.....690.....720.....750.....780.....
810.....840.....870.....900.....930.....960.....end
Connected PSB to PPAR.
Start configuring PSB to Logical Domains (LDoms) Manager. [1800sec]
0.....end
Configured PSB to Logical Domains (LDoms) Manager.
Operation has completed.
```

Check the status of the newly added Fujitsu M10-4S Building Block.

```
XSCF> showboards -p 0
PSB  PPAR-ID(LSB) Assignment  Pwr  Conn Conf Test    Fault
---- ------------ ----------- ---- ---- ---- ------- --------
00-0 00(00)       Assigned    y    y    y    Passed  Normal
01-0 00(01)       Assigned    y    y    y    Passed  Normal
```

At this point, if the CPU cores for the newly registered Fujitsu M10-4S Building Block have not been activated, follow the steps in "4.2.1 Adding Resources through CPU Core Activation" to activate the CPU cores on the newly registered Fujitsu M10-4S Building Block.

Also, the new Building Block’s memory can be added by using the ldm add-memory command to add memory to the logical domain on which the Oracle database is running.

```
# ldm set-memory 64G OracleDB-domain 1
```

4.2.3 Oracle Database Operations

Check the CPU information recognized by Solaris OS.

```
# psrinfo -vp
The physical processor has 16 cores and 32 virtual processors (0-31)
The core has 2 virtual processors (0 1)
The core has 2 virtual processors (2 3)
The core has 2 virtual processors (4 5)
The core has 2 virtual processors (6 7)
…
SPARC64-X+ (chipid 0, clock 3700 MHz)
The physical processor has 16 cores and 32 virtual processors (32-63)
…
```

Check the memory size recognized by the Solaris OS in the same way.

```
# prtconf | grep Memory
Memory size:  8388608 Megabytes
```


Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

As described above, by adding resources using the Fujitsu M10-4S CPU core activation feature and PPAR-DR, it is possible to dynamically expand the CPU and memory recognized by Solaris OS.

To continue, check the CPU/memory resources that are recognized by Oracle 12*c*.

The CPU resources recognized by Solaris are dynamically detected by the Oracle database, which dynamically adjusts the cpu_count (threads) initialization parameter.

```
$ sqlplus / as sysdba
…
Connected to:
Oracle  Database  12c  Enterprise  Edition  Release  12.1.0.1.0  -  64bit
Production
With  the  Partitioning,  Automatic  Storage  Management,  OLAP,  Advanced
Analytics
and Real Application Testing options

SQL> show parameter cpu_count

NAME                           TYPE        VALUE
```

In the example above, Oracle 12*c* has dynamically recognized the number of CPU threads when using Fujitsu M10-4S (SPARC64 X+ CPU chip) with 16 CPUs/256 cores (512 threads) assigned to the Solaris OS.

Next, set and confirm the memory size used by Oracle 12*c*. Since the memory size used by Oracle 12*c* does not change automatically, adjust it with the memory_target initialization parameter.

Set the maximum memory size that can possibly be used in the database by dynamic expansion with the memory_max_target parameter in advance. Oracle 12*c* can dynamically expand the memory size up to the value of memory_max_target.

```
SQL> show parameter memory_

NAME                           TYPE        VALUE
------------------------------ ----------- ------------------------------
memory_max_target              big integer 6144G
memory_target                  big integer 3072G

SQL> alter system set memory_target = 6144G;

System altered.

SQL> show parameter memory_

NAME                           TYPE        VALUE
------------------------------ ----------- ------------------------------
memory_max_target              big integer 6144G
memory_target                  big integer 6144G
```

As described above, with Fujitsu M10-4S, it is possible to dynamically expand CPU/memory hardware resources, and the running Solaris OS and Oracle Database 12*c* properly recognize the added resources.

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

Since the CPU/memory resources that can be used by a Container Database (CDB) are dynamically expanded, additional Pluggable Database (PDB) that runs on the CDB can be added and started dynamically.

```
SQL> show pdbs

    CON_ID  CON_NAME                        OPEN MODE  RESTRICTED
----------  ------------------------------  ---------- ----------
         2  PDB$SEED                        READ ONLY  NO
         3  PDB001                          READ WRITE NO
         4  PDB002                          READ WRITE NO
```

```
SQL> create pluggable database pdb003 admin user syspdb identified by xxx
…
SQL> alter pluggable database pdb003 open;

Pluggable database altered.

SQL> show pdbs

    CON_ID  CON_NAME                        OPEN MODE  RESTRICTED
----------  ------------------------------  ---------- ----------
         2  PDB$SEED                        READ ONLY  NO
         3  PDB001                          READ WRITE NO
         4  PDB002                          READ WRITE NO
         5  PDB003                          READ WRITE NO
```

In the example above, PDB003 was added and started, and its OPEN MODE status has become "READ WRITE".

In this verification test, operations described above are executed to gradually add resources in the following way, and the increase in transactions and responses are measured.

- With one Fujitsu M10-4S Building Block, add CPU core resources (from 16 cores to 64 cores)
- Starting from a 2-node RAC configuration with one Fujitsu M10-4S Building Block for each node, add one Fujitsu M10-4S Building Block at a time to both nodes, to expand the RAC environment up to a 4 Building Block per node configuration. (**Figure 11**)

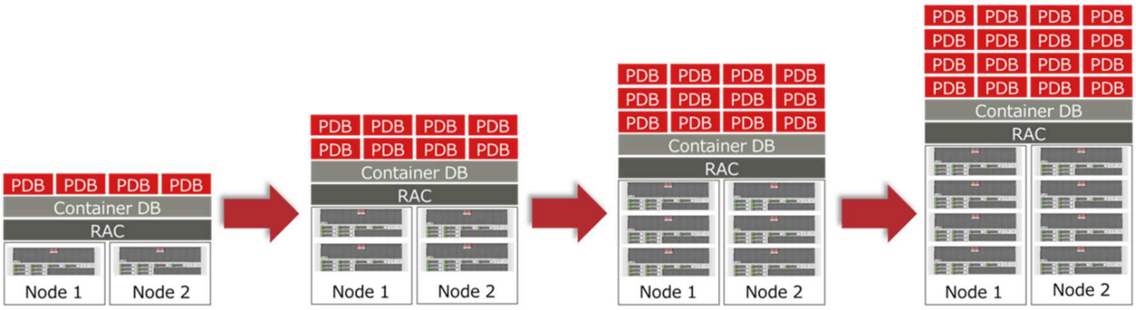

Figure 11. Gradually adding Fujitsu M10-4S Building Blocks and PDBs through PPAR DR

Note- The commands, setting items, and display examples described in this document are only examples.
The necessary settings and displays may differ depending on the configurations, etc. of the actual system.
Always refer to the appropriate product manuals.

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

4.3 Verification Results

The transaction processing volume (TPS) and average response time is measured when adding resources (from 16 cores to 64 cores) using CPU core activation and adding resources (from 2 Building Blocks to 8 Building Blocks) using PPAR DR. (**Table 1**) The transaction processing volume increases as resources are added and a constant average response time is maintained.

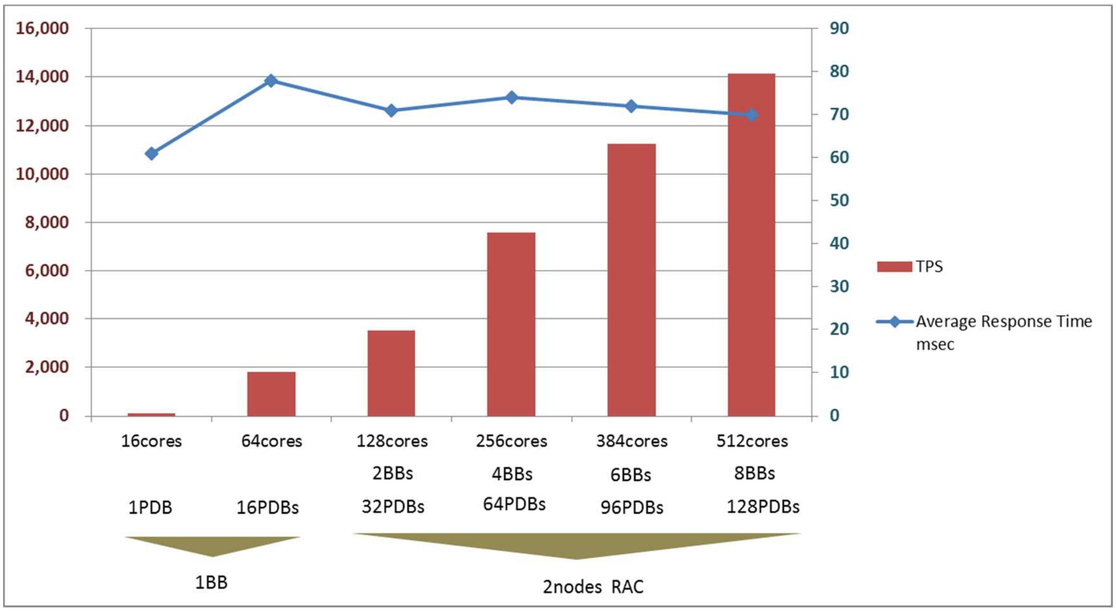

Table 1. Gradually adding Fujitsu M10-4S Building Blocks and PDBs through PPAR DR

However, suspending the adding of resources and continuing to increase the load, resulted in worse average response time, and no increase in the transaction processing volume. (**Table 2**)

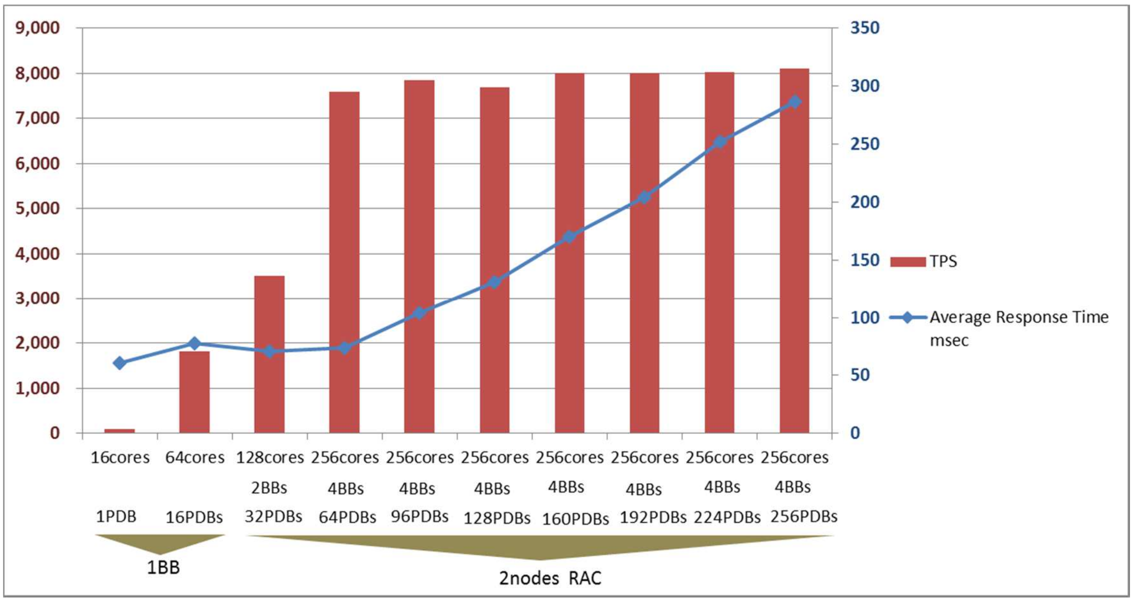

Table 2. Adding up to 4BB resources to Fujitsu M10-4S and suspending

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

With these results, the effectiveness of expanding server resources using the Oracle Multitenant feature of Oracle Database 12*c* and the CPU core activation and PPAR DR features of Fujitsu M10-4S is demonstrated. This flexible expansion without downtime makes Oracle Database 12*c* Pluggable Databases and Fujitsu M10 servers the best platform for extreme database consolidation.

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

5. Review

From this verification test, we were able to confirm that it is possible to expand a running server (Oracle database service) by using the Oracle Multitenant feature of Oracle Database 12*c* to add PDBs, and using the CPU core activation and PPAR DR features of Fujitsu M10-4S to add resources, without stopping the system.

Database consolidation with the Oracle Multitenant feature of Oracle Database 12*c* makes it possible to control investment costs and reduce running costs through efficient use of CPU and memory usage, reduction of storage I/O, and by using features for controlling the necessary hardware resources and managing them effectively.

In addition, Fujitsu M10-4S offers a high-performance CPU design, large memory capacity, and expansion up to 16 Building Blocks to accommodate further business growth and high levels of server consolidation. Fujitsu M10 makes it possible to control initial implementation costs and pay as your business grows.

Through the synergistic effect of Oracle Database 12*c* and Fujitsu M10-4S, customers can make more appropriate investments, not only in terms of software and hardware costs, but also including system integration costs, running costs, and maintenance costs.

From the point of view of efficiency, flexibility, investment cost control, and ICT cost reduction, combining Oracle Database 12*c* and Fujitsu M10-4S in a database server can be seen as not just a database consolidation platform, but also as the optimal database cloud platform for rapid database deployment.

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.

Fujitsu Limited
Shiodome City Center
1-5-2 Higashi-Shimbashi
Minato-ku, Tokyo 105-7123, Japan

The content of this document may change due to revisions at any time and without notice.
Fujitsu Limited makes no guarantees in regard to the content of this document.
In addition, it does not accept any responsibility for any damages incurred due to the content of this document.
UNIX is a registered trademark of The Open Group in the United States and other countries.
SPARC64, the SPARC64 logo, and all SPARC trademarks are trademarks or registered trademarks of SPARC International, Inc. (U.S.) in the United States and other countries and are used under license from SPARC International Inc.
All other product names and brand names may be trademarks or registered trademarks of the respective companies.
The system names, product names, etc. given in this document are not necessarily accompanied by a trademark notice ((R), TM).

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

blogs.oracle.com/oracle
facebook.com/oracle
twitter.com/oracle
oracle.com

Hardware and Software, Engineered to Work Together

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.0115

Copyright © 2015 FUJITSU LIMITED, All Rights Reserved.
Copyright © 2015 Oracle Corporation. All Rights Reserved.