ORACLE®

UTILITIES

Oracle Utilities Meter Data Management 2.0.1 Demonstrates Extreme Performance on Oracle Exadata/Exalogic

ORACLE®

Introduction

New utilities technologies are bringing with them huge increases in data volumes. Residential smart meters alone will increase the consumption data processed for billing purposes from 12 meter reads per year to 35,000 or more. As a consequence, utility business managers and executives are asking:

- Does hardware and software technology currently exist that is capable of handling the large quantity of data from smart meters?

- Can today's technology process the data and produce the results needed to drive the meter-to-cash process in a timely manner?

The tests described in this white paper answer these two questions affirmatively. They demonstrate that Oracle Utilities Meter Data Management (MDM) 2.0.1, running on a full rack Oracle Exadata Database Machine X2-2 and Exalogic X2-2, can easily handle smart metering data for any size of utility. In just one hour, for instance, this hardware/software combination can:

- Process more than one billion meter reads.[1,2] This approximates the volume of data produced in an hour by 250 million meters recording four interval consumption measurements per hour.[3]

---

[1] A "meter read" may be thought of as an interval or scalar data point—that is, a number reported by a meter representing consumption at the end of a given period of time.

[2] For this and all results reported throughout this paper, please note that actual results may vary, based on a broad range of implementation-specific factors, such as transaction mix, hardware platform, network parameters, and database size. Oracle does not warrant or guarantee that customers will obtain the same or similar results, even if they use the same or similar equipment and/or software applications. Oracle does not warrant, endorse, or guarantee any performance of any products, any results desired or achieved, or any statements made within this document.

- Calculate four bill determinants for each of 19.3 million bills—a total of more than 77 million bill determinants. A bill determinant is defined as being the results of a calculation that produces a customer's consumption for a defined period of time.

- These numbers far exceed today's typical utility requirements.

---

[3] It is unlikely that utilities would establish an AMI system that relied on interval data recorded on only one channel. Far more common is a scenario in which each meter reports intervals on multiple channels while also providing a check on consumption using a third channel. The scenario tested in this paper is, in fact, one of these more realistic ones; it uses three channels, two reporting 15-minute interval data and one providing a consumption check, for a total of 193 meter reads per day.

Methodology

For this demonstration, MDM 2.0.1, running on a full rack Exadata and a full rack Exalogic, performed two separate tests:

- It processed meter data and subjected it to validation and correction via configurable validation, editing, and estimation (VEE) rules, storing both the raw and validated/corrected data in the database, where it is available for use and review by business users.

- It used the validated and corrected meter read data to calculate the bill determinants that a billing / CIS system would use to create a customer bill among many other possible uses.

The tests covered two batch use cases, both based on meter reads from 5.5 million interval meters. Each meter had reads on three channels: one interval kWh channel, one interval kVarh channel, and one scalar kWh check channel. Both interval channels measured consumption every 15 minutes. The scalar check channel measured consumption once a day. Thus the total number of reads per meter was 193 per day or 5,790 per 30-day billing period.

The purpose of these tests is to validate and document the performance of the MDM VEE and bill determinant processes.

The creation of smart meter payloads and insertion into corresponding MDM staging table is not included in these test results. Subsequent tests (including tests of Oracle Utilities Smart Grid Gateway) will include conversion of the head-end-specific raw smart meter payload into the standard MDM format and the loading of data into MDM.

The historic data as well as the meter readings were randomly generated using an internal data generation tool to closely emulate production scenarios and realistic distribution of the data. This was necessary to ensure that the database cache hit rate stayed in a realistic range during the benchmark.

Test 1: Meter Read Processing. The raw meter data are read, edited, estimated, and stored as final measurements for the subsequent process. This process invokes the relevant set of VEE rules included in MDM, as detailed in Appendix 1.

Test 2: Bill Determinant Processing. Validated and corrected meter read data for the 30-day billing period are summarized as billable usage. Measurements from the interval kWh channel (2,880 reads per meter) are mapped to time of use (TOU) quantities based on the TOU map defined in the usage rule—in this case, three "buckets" representing consumption during on-peak, shoulder, and off-peak periods. A fourth bill determinant, obtained by summing the previous three bill determinants, represents total consumption.

For more details on the methodology, see Appendix 1.

Technical Environment

The tests were executed on an Oracle Exadata Database Machine X2-2 and Oracle Exalogic X2-2. This constitutes a full rack Exalogic server (30 application server nodes) in the application tier and a full rack Exadata server (8 database server nodes) in the database tier. Exalogic machines consist of Sun Fire X4170 M2 Servers as compute nodes, Sun ZFS Storage 7320 appliances, as well as required InfiniBand and Ethernet networking components. Each of the application servers in the application tier is connected to one of the database servers in the database tier using a round-robin algorithm.

For more details on the technical environment, see Appendix 2.

Technical Findings

- MDM running on Exadata and Exalogic exhibits strong horizontal scaling. Both meter read processing and bill determinant processing showed near linear horizontal scalability. In other words, throughput goes up proportionately as nodes (i.e. new servers on an application tier) are added to a system. (For details on horizontal scaling, see Appendix 3.).

- A compression ratio[4] of 1.8x was achieved for the database structures that store meter read data using hybrid columnar compression (HCC/Query High).

- The database partitioning strategy used demonstrates that rapid accumulation of large amounts of historical data does not have to affect Oracle application scalability. This is a key benefit of Exadata, which features highly tuned and balanced elements of computing, storage and networking that adapts and scale predictably.

Business Findings

These tests demonstrate that Oracle Utilities Meter Data Management 2.0.1, running on a full rack Exadata X2-2 and a full rack Exalogic X2-2 server:

- Validated, edited, estimated, and stored initial measurement data for smart meters at a rate of 1 billion meter reads per hour.

- Responded to 19.3 million requests for bill determinants with a total of 77.2 million bill determinants calculated in an hour.

---

[4] Subsequent to these tests, in a separate study, compression ratio as high as 10x been achieved for the table structure that stores meter read data.

- Furthermore, as Appendix 3 illustrates in detail, the measured throughput of 1 billion meter reads per hour was achieved using 24 compute nodes on Exalogic. Had all 30 compute nodes been used, MDM and the full rack Exalogic would have achieved much higher throughput, conservatively estimated at 1.4 billion per hour.

- Furthermore, as Appendix 3 illustrates in detail, the maximum measured bill determinant processing throughput of 19.3 million usage transactions per hour was limited by CPU resources on the Exadata hardware. Had there been more Exadata racks available, the full rack Exalogic could have conservatively achieved a throughput of 40 million usage transactions per hour.

Appendix 1: Additional Details on Methodology

Historic Data

The tests were conducted using 3 months of historical measurements for the 5.5 million meters. The historic data was randomly generated using a data generation tool to closely emulate production scenarios and realistic distributions of the data. This ensured that the database cache hit rate stayed in a realistic range during the tests.

Testing

For the measurement data processing test (i.e. applying the VEE rules), each test load consisted of slightly more than one billion meter reads.[5]

For the bill determinant calculation test, the billing period was set at one month (30 days). Each request for bill determinants resulted in the aggregation of consumption measured in the kWh interval channels. The resulting 2,880 meter reads for each meter's kWh interval channel were processed into four bill determinants: the first three representing total consumption during the month for each of three time-of-use billing periods and a fourth –a sum of the first three—represented total.

Note that for processing purposes, the tests grouped raw data into units of "Initial Measurement Data" (IMD). Each IMD consists of all the meter reads in a single channel for a single day. Each meter in these tests produced three IMDs per day. Note that IMDs are not identical in terms of the number of meter reads contained in each. For the tests described in this paper, the IMDs for each of the two interval channels contain 96 meter reads each, while the IMDs for the kWh scalar check channel contain only one meter read each.

There are a total of 193 (96 kWh interval + 96 kVarh interval + 1 kWh scalar) reads per smart device (meter).

---

[5] 5.5 million meters x 193 daily meter reads per meter = 1.0165 billion meter reads.

The following table represents the data volumes used to determine the workload profile for the tests.

ENTITY TYPE	NUMBER	COMMENTS
Meters	5,500,000	
Service Point	5,500,000	
Channels	16,500,000	Three channels per meter – 1 kWh interval channel (15 minute), 1 kVarh interval channel (15 minute) and 1 kWh scalar check channel.
Initial Measurement Data	1,485,000,000	Total number of channels x 30 days in the month x 3 months
Final Measurements	95,535,000,000	Row counts in the database at the end of the tests. Repeated test runs may add data.

VEE Rules

VEE RULES FOR INTERVAL INITIAL MEASUREMENT DATA PROCESSING

RULE	DESCRIPTION
UOM Check	This VEE rule checks the unit of measure (UOM) passed in with the Initial Measurement against the primary unit of measure configured on the measuring component's type
Interval Size Validation	This VEE rule algorithm validates that the seconds per interval (SPI) supplied with the Initial Measurement is equal to the interval size defined on measuring component's type.
Spike Check	This VEE rule algorithm looks for spikes by taking the highest interval and the third-highest interval, and determining the percent difference between the two.
Interval Interpolation 1	This VEE rule attempts to interpolate gaps in Initial Measurement Data sets using prior and subsequent intervals as starting points for linear interpolation.
Interval Interpolation 2	This VEE rule attempts to interpolate gaps in Initial Measurement Data sets using prior and subsequent intervals as starting points for linear interpolation.
Interval Interpolation 3	This VEE rule attempts to interpolate gaps in Initial Measurement Data sets using prior and subsequent intervals as starting points for linear interpolation.
Interval Averaging	This VEE rule performs estimation of missing consumption by aggregating MC's consumption history - the average consumption of which becomes the estimated amount.
Replacement Rule	This VEE rule algorithm checks if the intervals in an Interval Initial Measurement will replace existing final measurements.
Sum Check	This rule evaluates whether consumption for the current Initial Measurement Data is within a tolerance of the sum of the consumption during the same period for any measuring components related to the current one.
Negative Consumption Check	A VEE rule using this algorithm logs a VEE exception using the exception type and severity configured on the rule if the total consumption (as calculated by summing all measurements from Initial Measurement Data) is less than zero.
Hi Low Check	This VEE rule performs validation of measurements using the historical measurement data

**VEE RULES FOR SCALAR INITIAL MEASUREMENT DATA (CONSUMPTION CHECK) PROCESSING**

RULE	DESCRIPTION
UOM Check	This VEE rule checks the unit of measure (UOM) passed in with the Initial Measurement against the primary unit of measure configured on the measuring component's type
Multiplier Check	This VEE rule validates that the register multiplier supplied with the Initial Measurement is equal to the multiplier stored on the measuring component.
Replacement Rule	This VEE rule checks if there exists a final measurement with a date/time equal to the date/time in the current initial measurement being validated.
Negative Consumption	A VEE rule using this algorithm logs a VEE exception using the exception type and severity configured on the rule if the total consumption (as calculated by summing all measurements from Initial Measurement Data) is less than zero.
Hi Low Check	This VEE rule performs validation of measurements using the historical measurement data

USAGE RULES FOR BILL DETERMINANT CALCULATION

RULE	DESCRIPTION
Get TOU Mapped Usage	This usage rule is used to get time of use quantities from interval measuring components installed in the service points linked to the usage subscription for the specified 'Interval' usage period. Only measuring components that match the UOM/SQI defined in the usage rule instance are processed. Measurements within the period are mapped to time of use quantities based on the TOU map defined in the usage rule.
Service Quantity Math	This rule is for deriving the total consumption by summing each TOU period.
Validate Against Tolerance	This will validate the total consumption during "ON Peak" to be less than 500,000

Appendix 2: Technical Environment

Below is a high-level architecture topology diagram.
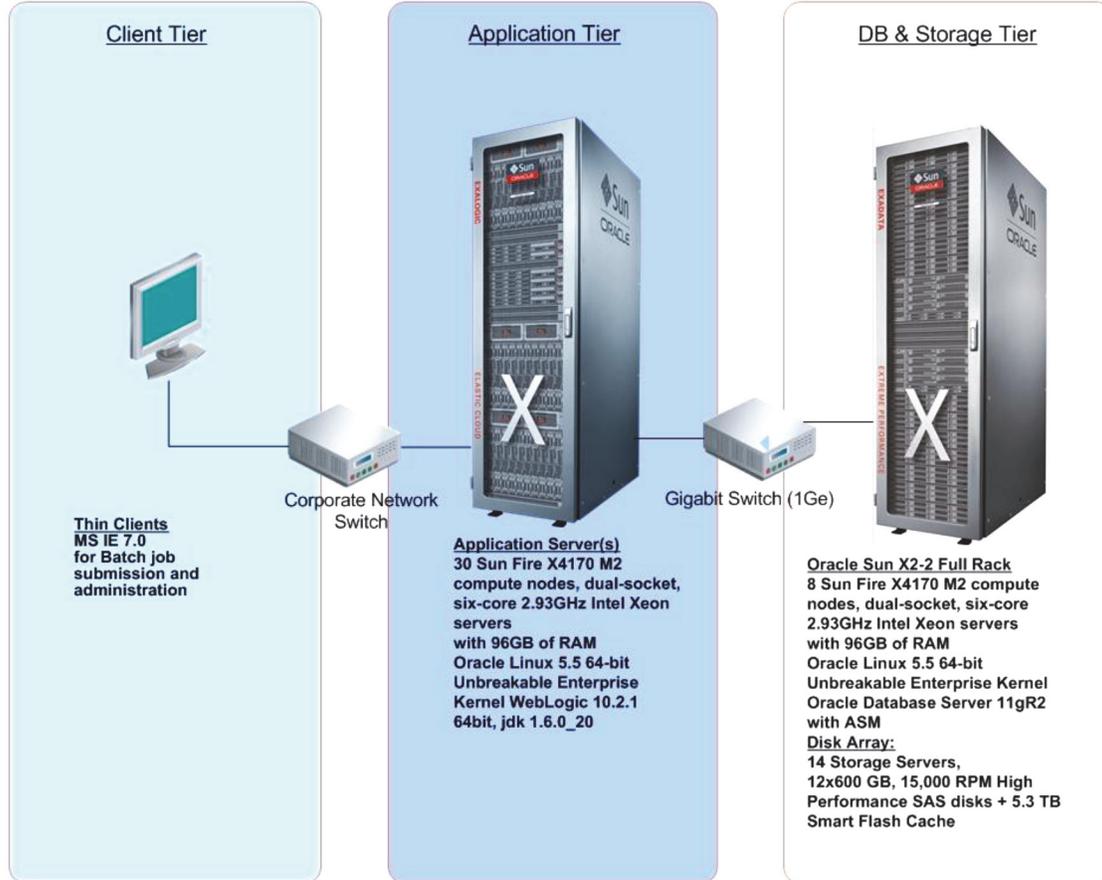
Exadata

For the Exadata configuration, the main database table for storing meter readings and usage was partitioned, and the data for that table was stored using Oracle Secure Files (an Oracle Database Enterprise Edition Advanced Compression feature) with the compression level set to medium. The following table further defines the specifications of the database and the host servers.

DATABASE SERVERS/STORAGE SYSTEM	
Platform	Exadata X2-2 Full Rack, Sunfire X4170M2
Operating System	Oracle Enterprise Linux 5 – 64 bit
O/S Version and Release	2.6.18-164.el5

Database Software / Version	Oracle Database Server 11gR2 with Automatic Storage Management (ASM)
Number of CPUs	8 RAC nodes, each with 2 Sockets (12 Cores)
Processor /CPU Speed	Intel Xeon X5670 / 2.93GHz
Memory	96 GB
Storage System	14 Exadata Storage Servers, each with 12 High Performance SAS disks and 384 GB Smart Flash Cache
Raid Level	ASM 2-way normal mirror.

Note that Exadata Storage Servers provide a high-bandwidth, massively parallel solution delivering up to 75 gigabytes per second of raw I/O bandwidth and up to 1,500,000 I/O operations per second (IOPS). These significant performance increases derive from the use of Exadata Smart Flash Cache in each Exadata Storage Server and hierarchical optimization of the Oracle Database.

Exalogic

For the Exalogic configuration, each Exalogic machine was connected to the database server via a ten-gigabit Ethernet interface. The following table outlines the application operating system and hardware specifications. One application server was used for the vertical scalability testing; two application servers were used for the horizontal scalability testing. (See Appendix for results of these tests.)

APPLICATION/BATCH MIDDLE TIER	
Platform	Exalogic X2-2 full Rack, Sunfire X4170 M2
Operating System	Oracle Enterprise Linux 5 – 64 bit
O/S Version and Release	2.6.18-164.el5
Software / Version	Oracle Utilities Meter Data Management version 2.0.1 Weblogic 10.0 MP2 – 64 bit
# CPU	30 compute nodes, each node has 2 Sockets and 12 Cores (Total 360 cores).
Processor / CPU / Speed	2.93GHz
Memory	96 GB on each compute node

Partitioning Strategy

The database storage data structures that keep high volume raw and finalized meter data reads were partitioned by the date-time of the readings. Furthermore, these structures were sub-partitioned by their primary access key. All associated index structures were created as local indexes to the partitions

and sub-partitions while making sure that the critical use cases contained the partitioning key during query and DML operations against these structures. This partitioning strategy not only helps by making query and DML operational performance on these data structures independent of the duration of the historical transactions to be kept for online access, it also helps by enabling archiving of this large volume of data in timely and efficient manner.

Appendix 3: Tests of Horizontal Scalability

Horizontal scalability (i.e. scale out) is defined as the ability of an application to demonstrate higher throughput when more nodes are added to a system, such as adding a new server on the database tier.

For these tests, horizontal scalability was measured by starting with 8 compute nodes and increasing the number (in increments of 8) up to a total of 24 compute nodes, while capturing a performance benchmark data point on each increment. Horizontal scalability was tested until the point at which the CPU on the application server or database tier was saturated.

In the case of the usage transaction / bill determinant calculations, the scalability was limited by database CPU.

The charts below depict the scalability characteristics of Oracle Utilities Meter Data Management v2.0.1, as measured during the tests. For the horizontal scalability evaluation tests, the execution threads were uniformly distributed across all the application servers.
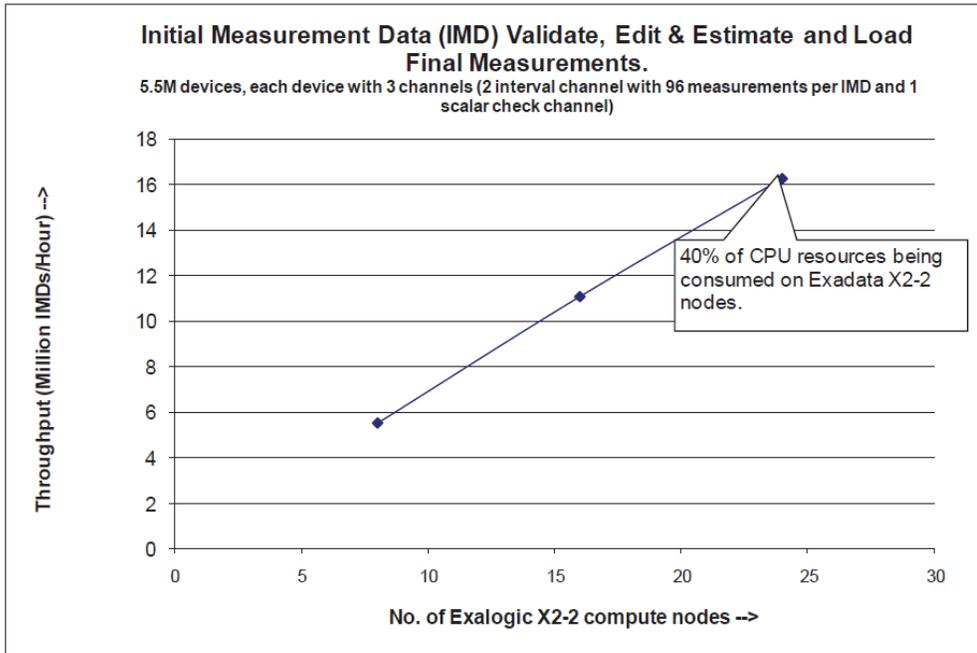
Horizontal scalability chart for interval read validate & load for Exalogic X2-2 compute node

Throughput shown above is "Interval Read Validate & Load" per hour. CPU utilization shown here is the average CPU utilization per node.
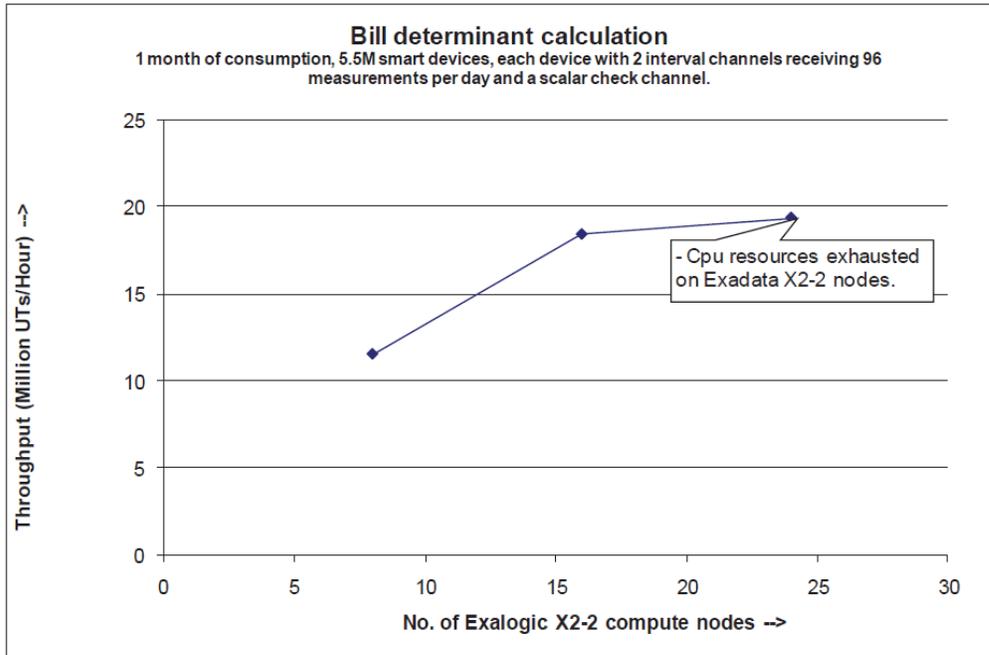
Horizontal scalability chart for billing for Exalogic X2-2 compute notes

Throughput shown above is expressed as "UTs/Hour" (i.e. Usage Transaction requests per hour), where each usage transaction requested results in the production of four bill determinants. The extrapolated throughput for full rack Exalogic X2-2 with sufficient Exadata is estimated to be approximately 40 million usage transactions per hour.

Both data processing and bill determinant processing, showed near linear scalability (horizontal scalability).

Note: Horizontal scalability on the database tiers was not measured for the following reasons:

- Exadata is a database machine made of multiple RAC nodes that communicate with storage servers. Each RAC node communicates with all of the storage servers. Thus, even when only one of the RAC nodes is used, it still uses all the available storage servers.

Oracle is committed to developing practices and products that help protect the environment

Hardware and Software, Engineered to Work Together