The REAL ‘Big Data’
Actually using smart sensors and other time sensitive data

Tom Rieger
IBM
tomrieger@us.ibm.com
952-221-6514

© 2011 IBM Corporation
A CASE STUDY: Smart Sensor data: The next BIG DATA challenge with always-on data generators

GPS devices on moving vehicles (buses, trains, trucks, cars), smart meters for utilities and all kinds of remote sensing equipment.

This is not just an incremental increase; this is 1000x of times increase.

How do you load it, process it, query it and gain the insight? And how do you do it economically?

THIS IS NOT A SOLUTION TRADITIONAL RELATIONAL MODELS SOLVE
City of Seattle – moving to real-time route, passenger and fleet mgmt
Traffic sensing

Thursday @ 09:05 AM
Traffic sensing
So let’s do some math

<table>
<thead>
<tr>
<th>Per meter/sensor</th>
<th>Reads a month</th>
<th>A year</th>
</tr>
</thead>
<tbody>
<tr>
<td>manual reads monthly</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>15 minute intervals</td>
<td>2,880</td>
<td>35,040</td>
</tr>
<tr>
<td>5 minute intervals</td>
<td>8,640</td>
<td>105,120</td>
</tr>
<tr>
<td>10 second intervals</td>
<td>259,200</td>
<td>3,153,600</td>
</tr>
</tbody>
</table>

What if you have 100,000 of these

<table>
<thead>
<tr>
<th>Per meter/sensor</th>
<th>Reads a month</th>
<th>A year</th>
</tr>
</thead>
<tbody>
<tr>
<td>manual reads monthly</td>
<td>100,000</td>
<td>1,200,000</td>
</tr>
<tr>
<td>15 minute intervals</td>
<td>288,000,000</td>
<td>3,504,000,000</td>
</tr>
<tr>
<td>5 minute intervals</td>
<td>864,000,000</td>
<td>10,512,000,000</td>
</tr>
<tr>
<td>10 second intervals</td>
<td>25,920,000,000</td>
<td>315,360,000,000</td>
</tr>
</tbody>
</table>

- In normal relational models, each one is a database row
- No matter what you throw at this, it will not scale if you just keep adding row-after-row
- BTW – 100,000 utility smart meters is equal to a city with population about ~200,000
The CASE Study

- The primary electric company for North Texas
- Completing a roll out of ~3.5 million smart meters
  - 3 million residential and 500,000 business customers
  - As of right now, adding 40,000 a week until completed this fall
- Currently doing reads every 15 minutes
  - Planning to go to 5 minute for commercial customers

**They discovered early in the project when they benchmarked 1 million meters of data for 90 days they were hitting a wall.**
The problem

- **Traditional Database was barely keeping up with the data**
  - Original database taking about 7 hours to read in the data for 1 million meters
  - Reports taking between 2 and 7 hours to run
  - *Problems with reading and writing data simultaneously*
- **What happens when they hit 3.5 million meters?**
  - 73 billion records stored will grow to 220 billion records
  - Even with linear scaling performance inadequate
    - *not be enough time in the day to read and process all this data*
- **Client is looking for ways to cut their costs**
  - 90 days of data for 1 million meters was ~1.3TB.
  - What happens when they hit full roll out and look to store it for 3 years?
  - They would like to do more with the hardware they have
    - Process the data more quickly
    - Allow other applications to run against the data
The Solution - Informix with TimeSeries

- Informix performance and storage comparison is linear with more meter data
- Informix gives even better results if you increase CPUs and storage

<table>
<thead>
<tr>
<th></th>
<th>INFORMIX</th>
<th>Oracle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to Load 1,000,000 Meters</td>
<td>18 Minutes</td>
<td>7 Hours</td>
</tr>
<tr>
<td>Reports</td>
<td>Seconds to 11 min</td>
<td>2-7 HOURS</td>
</tr>
<tr>
<td>Storage Requirements 1,000,000 Meters</td>
<td>350GB</td>
<td>1.3 TB</td>
</tr>
</tbody>
</table>
What Informix does different

**Traditional relational answer**
- ROWS!!!
  - INSERT
  - INSERT
  - INSERT
- And this would create new records in a table
- For our case study, that would be 15 minute read rates for 3.5 million meters = ~122,640,000,000 rows PER YEAR

**Informix’s answer**
- There is a column *datatype* type ‘time series’
- You still ‘insert’, but the datatype grows to the right (like an array)
- Another way to look at it is the datatype grows like a columnar table - vertically
- For our case study, that would be 15 minute read rates for 3.5 million meters = 3,500,000 rows
The solution – Informix with time series

- A relational database that is still one of the top used in corporations today
  - You make a reservation at Hyatt, you touch Informix
  - You do online banking with Bank of America, you touch Informix
  - Wal-Mart has over 18,000 machines running Informix…..with 2 DBAs
  - CISCO embeds Informix in 25 different products they sell every day

- It is Scalable
  - Starting small and scaling up can happen on the fly

- AND autonomic/self healing/self running/self tuning/secure
  - All ‘tuning’ and ‘maintenance’ happens while the machine is running

- AND extensible
  - Pre-packaged extensions like time series, spatial, text search, etc
  - and the ability to code your own using SPL, C and JAVA extensions

- AND highly concurrent
  - 10s of users to 10,000s concurrent users

- AND mixed workloads
  - Balance transactional work against ‘those ugly queries’

- AND all transparent to your application, tools and code
What is next?

- **Better analytics**
  - If you can ask comparative questions with a few second response time across meter data, think of the opportunity
  - Roll trucks smarter, better predict power needs, when and where to buy power.
  - Ask ‘what if’ questions with greater granularity

- **Enable spatial analysis**
  - Informix also allows them to ask mapping questions with spatial extension
  - NOW you can compare ‘WHEN’ and ‘WHERE’ components like never before.
So – everyone loves a good comparison.......... 

- Oracle published a benchmark around their ability to process smart meter data in September 2011
- IBM published a benchmark around Informix’s ability to process smart meter data in October, 2011
- The solution did this using a meter data management (MDM) solution over the respective databases:
  - Loading of smart meter data from millions of meters
  - Ran the data thru a VEE (Validation, Estimation and Editing) phase
  - Processed a set of meters thru a billing cycle
- So I felt compelled to show the results
## Meter Data Management benchmark

<table>
<thead>
<tr>
<th>Metric</th>
<th>Oracle</th>
<th>Informix</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date published</td>
<td>Sept, 2011</td>
<td>Oct, 2011</td>
<td></td>
</tr>
<tr>
<td>CPU cores for database</td>
<td>96</td>
<td>16</td>
<td>6x fewer</td>
</tr>
<tr>
<td>Number of meters</td>
<td>5,500,000</td>
<td>100,000,000</td>
<td>18x more</td>
</tr>
<tr>
<td>load and validate per sec</td>
<td>277,777</td>
<td>421,000</td>
<td>1.5x more</td>
</tr>
<tr>
<td>per processor performance</td>
<td>2,894</td>
<td>26,313</td>
<td>9x more</td>
</tr>
<tr>
<td>Power Needs</td>
<td>Max 14 kW</td>
<td>Max 1.6 kW</td>
<td>8.75x less</td>
</tr>
</tbody>
</table>

**IBM P770/Informix**

**Oracle/Sun Exadata**
So what is my point?

- It is possible for any sized organization to capture and analyze sensor data
- It does not take a refrigerator to process it and scary storage subsystems to manage it.

- You can go from *automation* to *anticipation*
  - Based on X, you get Y
  - Greater amps pulled by a motor, means a filter needs cleaning
  - Lower MPG means tune up is needed
  - After X miles, time to get tires checked
  - If tomorrow’s weather is 92F and sunny, this city will consume 200MW/hr of power through this substation
  - If across a product line, a consistent product failure takes place, re-adjust Y (maint, usage, replace bad part, some combination).
Key Takeaways

- Traditional relational database structures do not manage sensor data well
  - Load
  - Query
  - Cost to store
  - Non-scalable

- IBM Informix with time series solves this problem
  - Open relational database
  - Extends to store, access and secure ‘other’ data
    - Like time series, spatial and others.
  - 66% less storage – this pays for it by itself
  - Load data 100x times faster
  - Over 100 functions in the database for time series
Next steps and questions

- **Download and try it anytime**
  - [www.ibm.com/informix](http://www.ibm.com/informix) and click ‘trials and demos’ and download the ‘Developer Edition’
  - The sample database has time series data in it to walk you through the capabilities
- **Check out** [www.youtube.com](http://www.youtube.com) and search on ‘Informix time series’
- **Ask me for further details**

Tom Rieger
IBM
tomrieger@us.ibm.com
952-221-6514