WLRA Spring 2014

HIGH PERFORMANCE SMART METER DATA ANALYSIS

WITH OPEN SOURCE SOFTWARE

Jason Mar
Electric Load Analysis

SDGE

A Sempra Energy utility

ROADMAP

SDGE
A Sempra Energy utility
1 2 3 4 5 6 7 8 9
2011 Smart Grid Roadmap
INTEGRATED AND CROSS-CUTTING SYSTEMS
Key
Policy Goal
Forecast
SDG&E Smart Grid Vision
In Flight Project
New Project - Policy
New Project - Value
New Project - Pilot
Enterprise Project
Figure 6-11
50,000 users of online energy tools
Economic DR at 5% of peak
11% GHG emission reduction
12,000 MW peak reduction
17,000 MW peak reduction
Default dynamic electric rate – small commercial customers
Optional dynamic electric rate – residential customers
Zero net energy – 100% new residential construction
30% GHG emissions reduction to 1990 levels
20% reduction in non-petroleum fuel use
33% Renewable Portfolio Standard
>260,000 PEVs
Provide usage data
10,000 customers using online energy tools
13,000 customers, 49 MW residential/90 MW total distributed renewables
>25,000 distributed renewables customers
>168 MW PV (CEC)
>60,000 PEVs
10% EE reduction from 2006
CSI: 3,000 MW of distributed solar
Provide real or near real-time access to usage data
1,000 PEVs
PRECISION TIME SYSTEM
DATA MANAGEMENT & ANALYTICS
LOW POWER WIDE AREA COMMUNICATIONS NETWORK
COMMUNICATIONS INFRASTRUCTURE
GRID COMMUNICATION SYSTEMS (INCLUDES NEXT GENERATION)
NEXT GENERATION APPLICATION PLATFORM
IPV6
CISCO DATA ARCHIVING
CISCO REPLACEMENT/UPGRADE
2006
2010
2015
2020
SDG&E Smart Grid Vision: by 2015
1 SDG&E continues to enhance near real-time analytic services for control room analysis (via time-series or alternative analytic tools) enabling predictive and geospatial analyses
2 Robust grid communications are enabling the near real time and real-time information access required operate the grid more efficiently
3 Smart Grid communications enabled along distribution lines connecting Distributed Energy Resource and energy storage providers, enabling timely and secure information exchange with the CAISO markets
4 SDG&E's system integration platform is providing improved interoperability among different systems
SDG&E Smart Grid Vision: by 2020
5 Fully integrated data systems enable a holistic approach to data management improving overall utility situational awareness regarding the state of the grid
6 SDG&E provides customers with opportunities to participate in the market using open standards and applications by leveraging time series data and event correlations, as well as meter data brought together using system integrations
7 SDG&E is enabling automatic server workload migration capabilities, based on availability, cost, and source (fossil fuel, solar, wind generated) of power
www.sdge.com/smartgrid
© June 2011 San Diego Gas & Electric Company. All copyright and trademark rights reserved.

PROBLEMS FACED BY LOAD RESEARCHERS

Slow Query Response Times
Funding and Procurement Complications
Difficulty Requesting Changes
Communication, Access, and Configuration Issues

SOLUTIONS

Slow Query Response Times

Data Caching and Compression

Funding and Procurement Complications

Open Source Software

Difficulty Requesting Changes from IT

Change Management Access

Communication, Access, and Configuration Issues

Organizational Change

COMPRESSION BENEFITS

Example:
One month of Hourly Interval Data
~1.5M customers
~48M rows
63 columns

Uncompressed CSV: 16 G
GZipped CSV: 3.5 G
GZipped Parquet Format: 2.5 G
(85% Reduction)

CACHING BENEFITS

Example:
Reading One Month of Hourly Interval Data
~1.5 M meters
~48 M rows
63 columns

Select avg(hour_01_kwh) from
Cache 1.41 seconds
Parquet 24.93 seconds
Data Warehouse: 20 minutes

USE CASES

Data Quality Analysis
Bill Impact Calculation

USE CASE 1:
INTERVAL DATA QUALITY

Workflow Steps

1. Load interval data for one month
2. Load billed consumption for month into table A
3. Group interval data by customer into table B
4. Select difference into table C
5. Select customers within difference threshold from table C
6. Determine cause for customers outside threshold

SDGE

GROUP INTERVALS BY CUSTOMER

```
shark> CREATE TABLE dec_usage_cached
AS SELECT id
, sum(hour_01_kwh) as i1 , . . .
, sum(hour_24_kwh) as i24
FROM parquet_mrres
WHERE bill_yr_mo = 201312
AND fl_billable = 'Y'
GROUP BY id;
```

Time taken: 165.403 seconds

48,033,631 Input Rows
1,539,649 Output Rows

SDGE

SELECT DIFFERENCE

```
shark> CREATE TABLE dec_qa_cached
AS SELECT A.id, A.prem_id, A.serv_pt_id, A.chnl_id,
FLOOR(A.total) as intrvl_consm
, B.billed_consm
, FLOOR(A.total) - B.billed_consm as diff
FROM dec_intrvl_consm_cached A
JOIN dec_consm_cached B
ON A.prem_id = B.prem_id
AND A.serv_pt_id = B.serv_pt_id;
```

Time taken: 6.684 seconds

1,539,567 Output Rows

SELECT CUSTOMERS WITHIN THRESHOLD

```
shark> CREATE TABLE dec_bipop_ids_cached
AS SELECT id
FROM dec_qa_cached
WHERE abs(diff) < 8;
```

Time taken: 3.693 seconds

1,539,567 Input Rows

1,351,333 Output Rows

SDGE

USE CASE 2:
BILL IMPACT CALCULATION

Workflow Steps

1. Load interval data for population
2. Select determinants into table A
3. Select bill into table B
4. Analyze results

LOAD INTERVAL DATA FOR POPULATION

```
shark> CREATE TABLE dec_bipop_usage_cached
AS SELECT A.id,
B.i1, . . . , B.i24
FROM dec_bipop_ids_cached A
JOIN dec_usage_cached B
ON A.id = B.id;
```

Time taken: 34.55 seconds

1,351,333 Output Rows

SELECT DETERMINANTS

```
shark> CREATE TABLE dec_bipop_bill1_det_cached
AS SELECT id,
i1 + i2 + i3 + i4 + i5 + i6 + i7 + i8 as OFFPK,
i9 + i10 + i11 + i12 + i13 + i14 + i15 as SEMIOFFPK,
i16 + i17 + i18 + i19 as ONPK,
i20 + i21 + i22 + i23 + i24 as SUPEROFFPK
FROM dec_bipop_usage_cached;
```

Time taken: 7.063 seconds

1,351,333 Output Rows

SELECT BILL AMOUNT

```
shark> CREATE TABLE dec_bipop_bill1_bill_cached
AS SELECT offpk * 0.15 as offpk_amt,
semi offpk * 0.17 as semioffpk_amt,
onpk * 0.22 as onpk_amt,
superoffpk  * 0.13 as  supero ffpk _amt,
offpk  * 0.15 + semi offpk * 0.17
+ onpk * 0.22 + supero ffpk * 0.13 as bill_amt
FROM dec_bipop_bill1_det_cached;
```

Time taken: 8.26 seconds

1,351,333 Output Rows

ANALYZE OUTPUT

```
shark> SELECT
ROUND(MIN(bill_amt), 2) as min_bill,
ROUND(MAX(bill_amt), 2) as max_bill,
ROUND(AVG(bill_amt), 2) as avg_bill,
ROUND(SUM(bill_amt), 2) as total
FROM dec_bipop_bill1_bill_cached;
min_bill   max_bill    avg_bill  total
0.0        381976.0    148.26    200,354,652.68

Time taken: 0.748 seconds
1,351,333 Input Rows
```

NEXT STEPS

Investigate cause of non-matching data
(Bad Estimates? Missing Reads?)

Calculate Additional Rate Structures and Parameter Sets
(Use cached interval data to avoid I/O)

Compare New Rate Structures and Prices
Store Queries in Version Control System

SDGE

10
18
38
15
40
46
58
106
16
20
48
12
26
36
104
44
52
28
50
156
158
30
56
190
54
34
24
22

FIG. 1

SOFTWARE USED

Shark - SQL interface to Spark
Spark - RAM-based Computation Engine

Parquet Format - Columnar, Compressed Storage

Hadoop File System - Scalable Storage

Scala/Java - ETL and Runtime Environment
D3.js - Browser-based data visualization
Linux - Operating System

WHY IS IT FAST?

HOW IS THIS DIFFERENT?

Why are previous MR-based systems slow?

- Disk-based intermediate outputs.
- Inferior data format and layout (no control of data co-partitioning).
- Execution strategies (lack of optimization based on data statistics).
- Task scheduling and launch overhead!

SPARK

Task Scheduler

Runs general DAGs

Pipelines functions within a stage

Cache-aware data reuse & locality

Partitioning-aware to avoid shuffles

SHARK

Shark Architecture

PARQUET FORMAT

BEFORE: IO BOUND

```
mware

  0  [          0.0%]    6  [||||||||||98.8%]    12 [          0.0%]    18 [          0.0%]
  1  [|         1.2%]    7  [          0.0%]    13 [||        1.9%]    19 [||        1.2%]
  2  [          0.0%]    8  [          0.0%]    14 [          0.0%]    20 [          0.0%]
  3  [          0.0%]    9  [          0.0%]    15 [          0.0%]    21 [          0.0%]
  4  [          0.0%]    10 [|         0.6%]    16 [          0.0%]    22 [          0.0%]
  5  [||        3.1%]    11 [          0.0%]    17 [||        1.2%]    23 [          0.0%]
  Mem[||||||||||||||||||||||||| 27413/96548MB]    Tasks: 512, 1212 thr; 2 running
  Swp[|                            0/16386MB]    Load average: 1.12 1.08 1.18
                                                 Uptime: 226 days(!), 12:07:23

  PID USER      VIRT  CPU%  MEM%   TIME+  Command
18074 sassrv    572M   0.0   0.1  0:00.00 /app/sasinstall/SASFoundation/9.3/sasexe/sas -noterminal -
18075 sassrv    572M   0.0   0.1  0:00.99 /app/sasinstall/SASFoundation/9.3/sasexe/sas -noterminal -
18076 sassrv    572M   0.0   0.1  0:01.34 /app/sasinstall/SASFoundation/9.3/sasexe/sas -noterminal -
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

AFTER

mware
0 [|||||||||77.3%]
1 [|||||||||69.1%]
2 [|||||||||87.2%]
3 [|||||||||64.4%]
4 [|||||||||76.7%]
5 [||||||||| 53.7%]
6 [|||||||||84.7%]
7 [|||||||||69.7%]
8 [||||||| 44.5%]
9 [|||||||||79.1%]
10 [|||||||||66.5%]
11 [|||||||||62.6%]
12 [|||||||||68.5%]
13 [|||||||||74.4%]
14 [|||||||||65.0%]
15 [|||||||||95.7%]
16 [|||||||||98.2%]
17 [|||||||||98.2%]
18 [|||||||||63.2%]
19 [|||||||||98.2%]
20 [|||||||||97.5%]
21 [|||||||||89.6%]
22 [|||||||||98.8%]
23 [|||||||||93.3%]
Mem[||||||||||||||||||||||||30855/96548MB]
Swp[| 0/16386MB]
Tasks: 512, 1381 thr; 10 running
Load average: 14.33 7.14 4.63
Uptime: 226 days(!), 12:27:33
PID USER VIRT CPU% MEM% TIME+ Command
8339 root 94728 0.0 0.0 0:00.00 /opt/VRTSpbx/bin/pbx_exchange
8360 dbus 40980 0.0 0.0 0:00.00 dbus-daemon --system
8358 dbus 40980 0.0 0.0 0:19.80 dbus-daemon --system
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice - F8Nice + F9Kill F10Quit

DEMO

SHARK WEB INTERFACE

METASTORE AND HDFS WEB INTERFACE

D3.JS BILL IMPACT VISUALIZATION

SHARK WEB INTERFACE

METASTORE BROWSER

HDFS BROWSER

BILL IMPACT VISUALISATION

Made with Crossfilter

SDGE

HISTORY

Open Source is the renewable energy of technology
Freedom + Security

View Timeline