Lessons Learned from Implementing an IDAA

Scott Chapman

Enterprise Performance Strategies, Inc.

scott.chapman@epstrategies.com

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides **education**, professional **networking** and industry **influence**.

CELEBRATING 60 YEARS OF SHARE
Influencing IT Since 1955

Contact, Copyright, and Trademark Notices

Questions?

Send email to Scott at scott.chapman@EPStrategies.com, or visit our website at http://www.epstrategies.com or http://www.pivotor.com.

Copyright Notice:

Trademarks:

Enterprise Performance Strategies, Inc. presentation materials contain trademarks and registered trademarks of several companies.

The following are trademarks of Enterprise Performance Strategies, Inc.: **Health Check®, Reductions®, Pivotor®**

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries: IBM®, z/OS®, zSeries® WebSphere®, CICS®, DB2®, S390®, WebSphere Application Server®, and many others.

Other trademarks and registered trademarks may exist in this presentation

Performance Workshops Available

During these workshops you will be analyzing your own data!

- WLM Performance and Re-evaluating of Goals
  - Instructor: Peter Enrico and Scott Chapman
  - September 28 – October 2, 2015 – Columbus, Ohio, USA

- Parallel Sysplex and z/OS Performance Tuning
  (Web / Internet Based!)
  - Instructor: Peter Enrico and Scott Chapman
  - November 17 – 19, 2015

- Essential z/OS Performance Tuning Workshop
  - Instructors: Peter Enrico, Scott Chapman, Tom Beretvas
  - October 19 - 23, 2015 – Dallas, Texas, USA

- z/OS Capacity Planning and Performance Analysis
  - Instructor: Ray Wicks

EPS Sessions at Share

Peter Enrico

Day	Time	Location	Presentation
Wed	11:15	Asia 3	SMF 113 Processor Cache Counter Measurements – Overview, Update, and Usage
Wed	1:45	Asia 3	WLM – Effective Setup and Usage of WLM Report Classes
Thu	11:15	Asia 3	zProcessor Consumption Analysis (including z13), or What is Consuming All the CPU?

Scott Chapman

Day	Time	Location	Presentation
Tue	11:15	Asia 3	Memory Management in the TB Age
Tue	3:15	Southern Hemisphere 4	Lessons Learned from implementing an IDAA
Fri	11:15	Asia 3	WLM in One Page

Disclaimer / Trademarks

- “IDAA” in this presentation short for IBM DB2 Analytics Accelerator*
- IBM and other trademarks hereby acknowledged and are owned by organizations other than me
- All information herein believed to be correct, but products change and environments differ, so you should only consider the information herein as being correct at some point in time in some environment, but may or may not apply to your situation

* Not the International Doctors in Alcoholics Anonymous, nor the Idaho Auto Auction

Agenda

- What is an IDAA and why you might want one
- What to consider before you buy
- Change your thinking: IDAA <> DB2
- Incremental update: performance, measurement, tuning
- Testing
- Examples of performance improvements

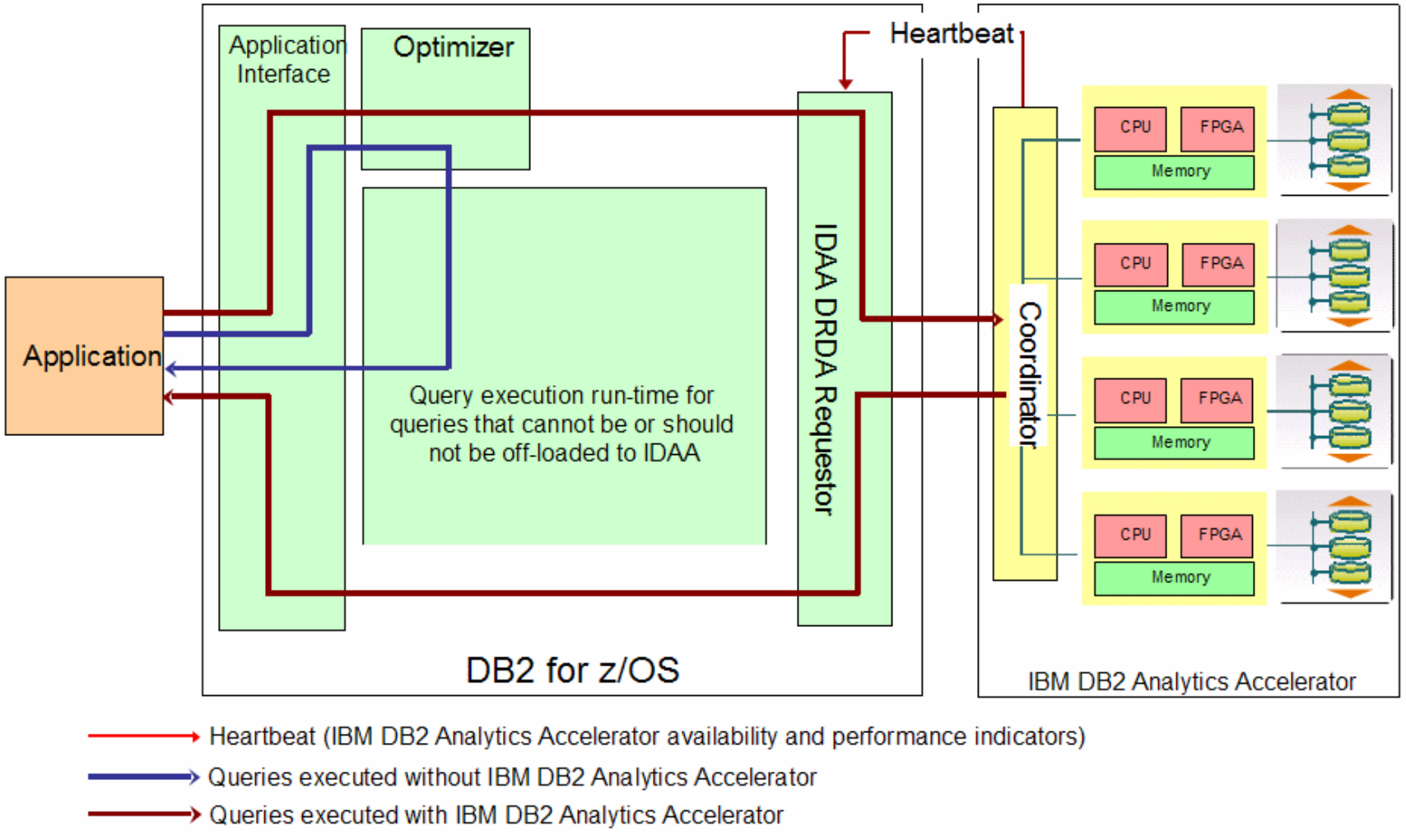
Red = surprises or particularly important notes

Blue = PTF 5 updates, as I understand them

“Any sufficiently advanced technology is indistinguishable from magic.”

Arthur C. Clarke’s Third Law

IBM DB2 Analytics Accelerator

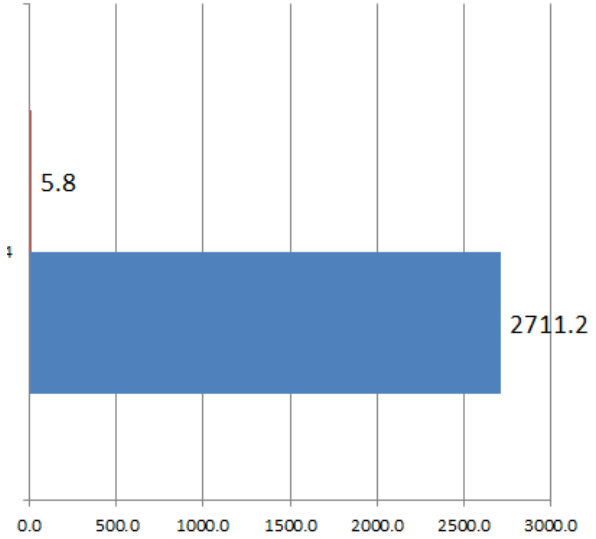


From IBM Redbook "Optimizing DB2 Queries with IBM DB2 Analytics Accelerator for z/OS"

Why you want one

- Potential huge improvement in query performance
  - Elapsed time reductions on the order of 99% possible!
  - Enables queries and business functions that might have been previously impractical
    - Data warehousing on the mainframe
    - Avoid copying data off for reporting and worrying about securing that copy of the data
    - Bring the analytics to the data, not the data to the analytics
  - Reduces or eliminates query-tuning efforts
- Mainframe DB2 CPU offload / cost savings
  - Big “maybe” here—has to be the exact right situation to actually save money
  - Incremental update will cost money

Real life results



“ I was thinking ‘why is this running so fast?’ Then I realized it went to the IDAA ”

Subject: Looks like took about 10 clock minutes to go thru all opcos. Normal clock runtime is 6-8 hours....thanks

Good Morning
I have a query that I normally would run separately for each Operating Company (run 11 times). It would take about 45 minutes for the each to run. I would run some concurrently so it would take overall about 4 or 5 hours to retrieve all of the data...
This morning I ran 1 query that included all Operating Companies and it completed in less than 20 seconds.
OUTSTANDING!!!

IDAA works differently than DB2

- Every query in IDAA is a tablespace scan
  - Highly parallel, high speed scans
  - Scans can limit pages read with zone maps
- Very little DBA work to do
  - Add & enable tables for replication / acceleration
  - Impossible to schedule or manually trigger reorgs
- Limited performance knobs to turn
- No indexes, but “keys”
  - Keys may influence performance, but are not necessarily required

Distribution Keys

- Directs rows to particular nodes
- Used primarily to improve join query performance by keeping rows from different tables with the same distribution key on the same node
- Avoids “broadcast”: sending between nodes to satisfy join condition
- To be effective the two distribution key column lists must be the same!
- Consider using only single columns in the distribution keys!

Organizing keys

- Ensures that there are zone maps for the columns in the organizing key
- Will also affect physical order of data when the table is groomed and you can't control when that occurs
- Allows IDAA to skip reading data extents that don't contain values the query is looking for
- The order of the keys doesn't matter
  - It's magic
  - Consider using a single high-cardinality column
- Do not use organizing keys on EBCDIC character columns prior to IDAA v4 PTF 3!

More math than magic

- Uses a space filling curve such as the Hilbert curve

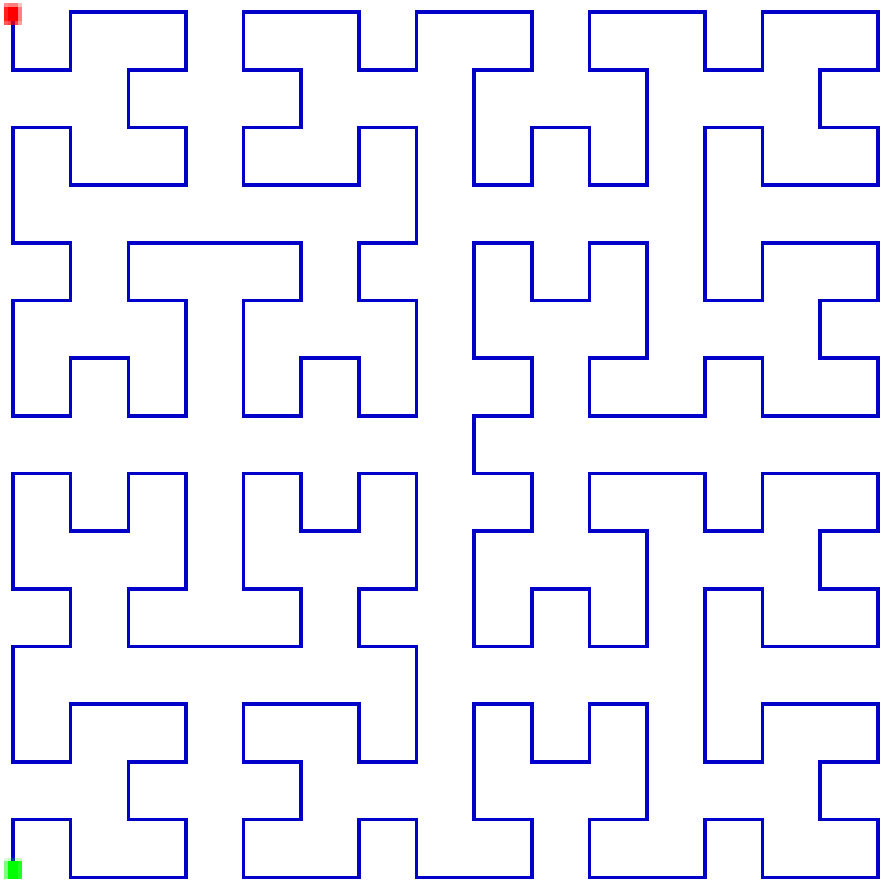


Image courtesy wikimedia cc-by-sa-3.0, see:
http://commons.wikimedia.org/wiki/Hilbert_curve#mediaviewer/File:Hilbert-Kurve.png

Before you buy…

Do a suitability study

- Not all queries are IDAA-eligible, depending upon:
  - Version of DB2
  - Functions used in the queries
  - Data types
  - Expected DB2 response time
  - Whatever else the optimizer decides to use
- Engage IBM to do a study of your queries
  - Will be some effort to collect the data, but well worth it
- Results:
  - Percentage of query CPU time expected to offload
  - Approximate speed-up benefit from running in the IDAA
  - Percentage of queries expected to offload

Consider Capacity

- Disk capacity
  - IDAA generally compresses better than DB2
  - But plan for more just barely enough capacity
    - Deleted data not immediately deleted
  - Disk throughput likely more important than TB capacity
- Processor capacity
  - IBM study should give you a general guideline
- Consider replication as a new query workload
  - Requires disk throughput and processor capacity
  - Replication and query workloads can impact each other's performance
    - Replication is supposed to be the higher priority
    - Running query snippets must finish before giving resources to replication

Don’t forget connectivity

- Need 10GbE cards on the mainframe(s) connecting to the IDAA
  - Not inexpensive—for 2 mainframes with redundant connections to a small IDAA, the connectivity can be a big portion of the IDAA cost
- Ideally need 10GbE switches
  - Not absolutely required for small installations, but highly recommended
  - May be able to use ports on existing switches

Before production…

High-level Plan

- Installation
  - Physical placement / connectivity
  - DB2 maintenance / configuration
- Replication (if planning on using)
  - Enable replication for everything that you intend to replicate
  - Validate replication latency
- Query testing / validation
  - Acceleration enabled only by setting special register
- Production (acceleration enabled by default)

Before Installation

- Note Pre-reqs and maintenance needed, as well as known issues:
  - Note need for current CLI/ODBC and DB2 connect to avoid problems with selecting timestamps
  - http://www-01.ibm.com/support/docview.wss?uid=swg27039487
- Create ZFS for holding the IDAA software that will need to be transferred to the IDAA
  - Transfer all the software before starting the install
- Installation requires the XML Tool Kit loadlib; you might want to identify this now

Installation

- The order the components/maintenance are installed is important
  1. DB2 PTFs
  2. IDAA GUI
  3. IDAA Access Server
  4. IDAA Replication Engine
  5. IDAA Server
  6. IDAA Stored Procedures

Note the Stored Procedures come with security set to “USER”, which caused us issues. Resetting them to “DB2” and then limiting user access to them worked better for us.

Installation

- Be sure network switches allow jumbo frames
  - Failure to allow jumbo frames will cause all manner of seemingly random problems
- Run network diagnostic stored procedure to verify network performance
- Load a few tables and make sure you can route queries to the IDAA

IDAA Monitoring

- Consider adding automation to monitor IDAA
- If replication stops or is behind, do you want to stop acceleration?
- If acceleration is stopped, do you want to be notified?
- DSN8X* messages sent to console, consider trapping them and taking appropriate action

Incremental Update

- Incremental Update = Replication = Change Data Capture (CDC)
- Uses Infosphere CDC to send updates from DB2 to IDAA
  - Only committed data is sent to IDAA
  - Utilizes log data to capture changes
- CDC started task on z/OS may need a few GBs of memory for the staging area
  - Look out for single SQL statements that do mass changes
  - Run out of memory = replication stops
    - Optional in PTF 5: restart without the offending table

Replication performance

- Expect your DB2 to be able to sustain more update / delete activity over time than your IDAA
  - Depending, of course, on lots of things
- IDAA is optimized for queries, not updates
  - Inserts are relatively cheap (put at “end”)
  - Deletes are expensive because the IDAA has to find the row that was updated
  - Remember all queries are scans!
  - Updates are handled as a delete and then an insert
    - Most expensive change to replicate
    - But the deletes / inserts for updates are included in all the other deletes / inserts for a table

Replication operational issues

- For any table that is being replicated, if you make a table structure change (alter/add column):
  - Remove table from replication/acceleration
  - Remove table from IDAA
  - Make table change
  - Reorg the table
  - Add table back to IDAA
  - Enable replication for table
  - Enable replication, do initial load
- Tables without any unique indexes may be problematic
  - Duplicate records may not get to the IDAA (if multiple records affected in one unit of work)

How replication works

- Committed units of work copied to CDC staging area (memory)
- Sent from staging to IDAA continuously
- IDAA puts updates for individual tables into “buckets” (each table has its own bucket)
  - If the bucket fills, the changes are written to IDAA disk, but not committed to IDAA
- At expiration of target latency period:
  - Further changes from CDC held
  - All buckets are emptied (written to disk)
  - Table updates are done serially
    - Optional in PTF 5: apply 4 streams in parallel
- Once all tables have been updated, the changes are committed to IDAA
  - Optional PTF 5 parallel apply commits in parallel too (data may be inconsistent)
- Wait for next latency target period expiration

Latency measurement

- Latency is reported by the IDAA and updated at the end of every replication interval
- Reported latency = current time – time of last update that was applied
- This is a point in time, best case (minimum) view of latency
- Actual maximum time between an update being committed in DB2 and committed in IDAA may be much more than the reported latency

What impacts latency

- Number of tables changed per interval
  - Minimum delay of about 500ms per table with changes
    - Delay is 2x if both deletes and inserts are present
    - This may change in the future (we can hope!)
- Number of rows deleted/updated per interval
- Updates and deletes impact latency much more than inserts
  - Have to find the row affected in the IDAA
  - Updates/deletes to larger tables more impactful than to smaller tables

Important note

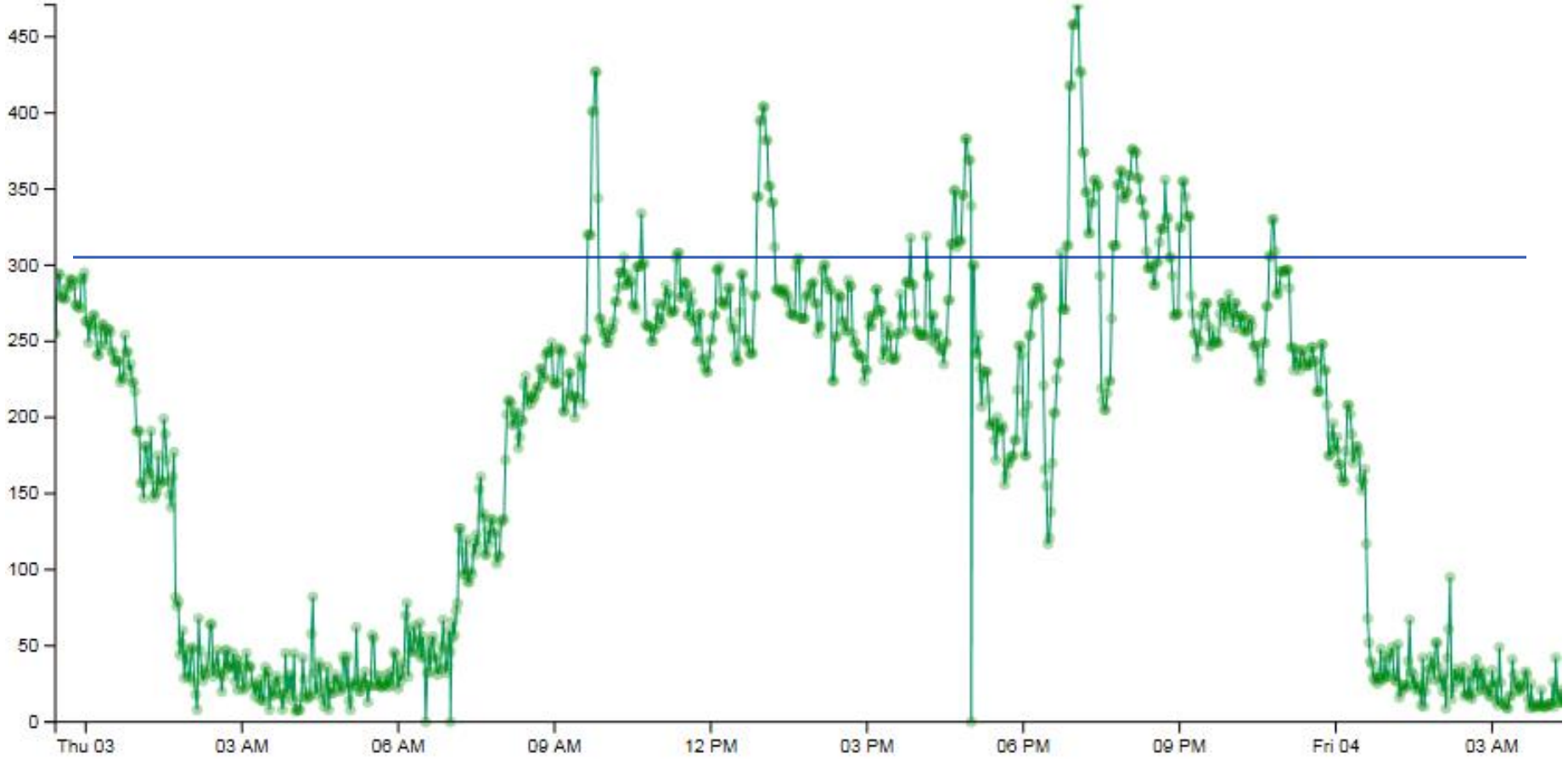
- There is only one latency per DB2plex in the IDAA
  - Changes for all tables are processed together
  - Can’t set individual replication targets per table
  - Makes sense: you want the data to be consistent!
- So if a large table is receiving lots of changes, latency may suffer and it will suffer for all tables!
- Scott’s Suggestion: enable all replication for all tables you expect to need it on before opening up the IDAA to production queries

Verify latency

- You might want to consider capturing latency value and tracking it over time
- Call SYSPROC.ACCEL_CONTROL_ACCELERATOR with <getAcceleratorInfo>

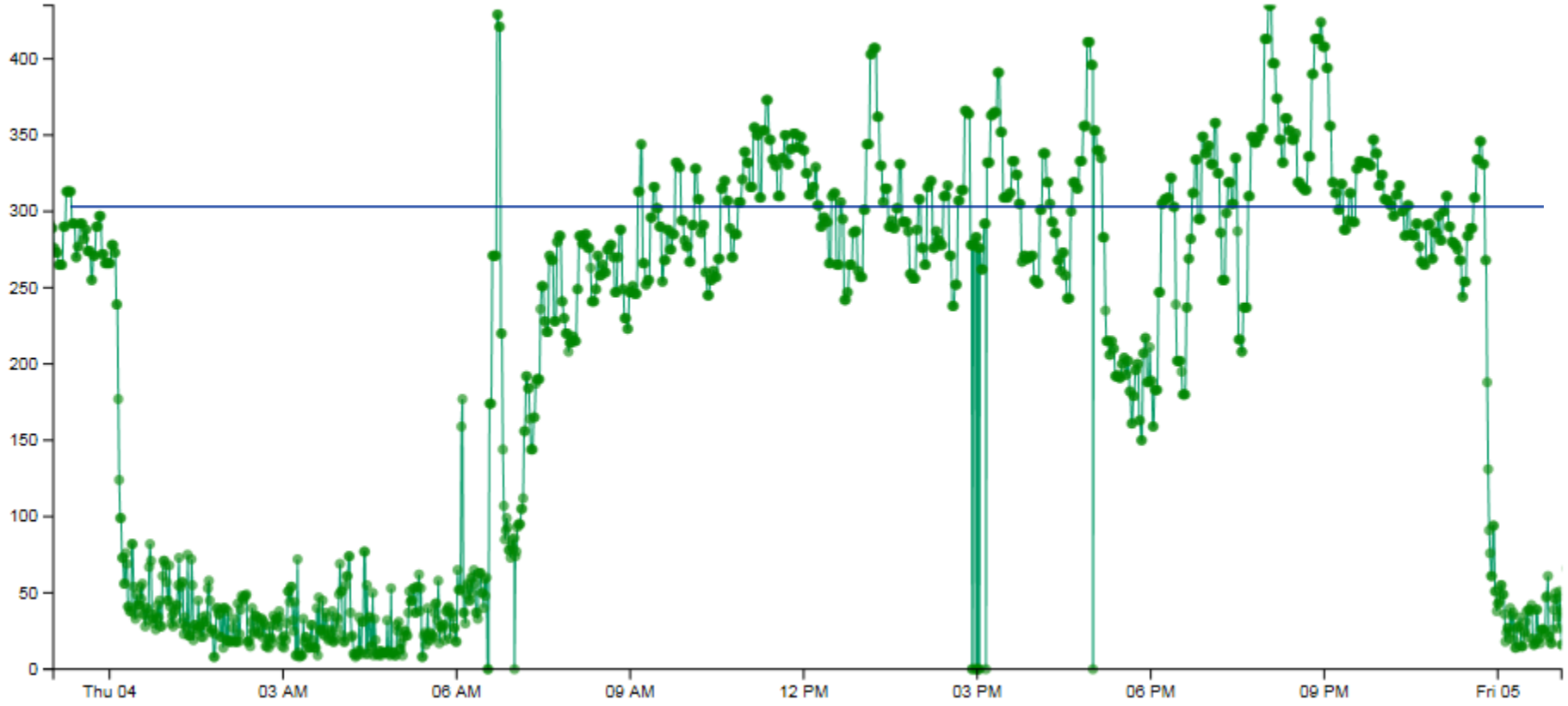
```
<replicationInfo state="STARTED"
                 lastChangeTimestamp="2014-07-02T13:24:20.266071Z"
                 latencyInSeconds="49"
                 activeAccessServerVersion="10.2.1.2221"
                 activeReplicationEngineVersion="10.2.1 [Build CCTRJYPP_20_36]">
   <sourceAgent insertCount="78544229"
                 updateCount="60328914"
                 deleteCount="17650066" />
   <targetAgent insertCount="116760677"
                 updateCount="0"
                 deleteCount="55866618" />
</replicationInfo>
```

Latency Tracking



No production query workload

Latency Tracking



With production query workload

Tuning replication

- Make sure CDC started task on z/OS classified appropriately in WLM
  - Probably want it as SYSSTC
- IDAA parameters:
  - Bulk size (likely little benefit to changing)
  - Target replication latency (likely should leave at default 1 min)
- Organizing keys
  - Need to contain columns from a unique index to be useful
  - May not take effect immediately after definition
    - Table needs to be groomed—which the IDAA will do when it wants to
- Distribution keys?!?
  - Have been told they may be helpful in certain replication situations, but I don't believe this to be true
- Consider turning off replication for tables getting regular bulk updates
- IDAA resource split between queries and replication
  - Replication is supposed to take priority

Replication Performance Analysis

- There are a number of logs and messages you can look for to find information about replication
  - z/OS started task
  - IDAA trace dump
- In the trace dump, pg.log is particularly interesting
  - Can determine time to apply changes per table
  - Can determine that minimum overhead (~500ms)
  - Likely will change in the future as they improve the replication process

Replication worst case scenario

Your latency goal may not be compatible with the mix of tables you want to replicate

- Consider removing tables
  - Remember all tables in the query must be in the IDAA for the query to be processed by the IDAA!
- Possibly consider upgrading to larger IDAA
  - “Upgrade” = remove current box and install new larger box
  - Larger box may not always improve replication performance substantially!
    - Understand where your performance bottleneck is: if it’s because you have too many tables with updates, a bigger box might not help replication enough

Query Testing

- Queries will go to IDAA if:
  - Acceleration enabled on IDAA
  - Acceleration enabled on all tables in query
  - Special register “query acceleration” set to allow it
    - Either by DB2 ZPARM default or “set current query acceleration = …” before query
- For controlled testing:
  - Set ZPARM to “NONE” so by default no queries will go to IDAA
  - Do a “set current query acceleration = …” before executing test queries

Query Acceleration values

- None
  - Do not send query to IDAA
- Enable
  - If the query is eligible and DB2 thinks it will benefit from running there, send it to the IDAA
- Enable with failback
  - Same as previous, but if the query fails during prepare or open, run it in DB2
- Eligible
  - If the query is eligible, send it to the IDAA regardless of whether it seems like it would benefit
- All
  - Send to IDAA regardless of whether DB2 thinks it's eligible and/or useful to run in the IDAA

Initial ZPARM setting during initial testing

Regular production usage should be this

Possibly useful for ongoing verification testing

Formal query testing / verification

- It may be useful to gather a number of eligible queries and run them daily both in IDAA and DB2 to verify the timing and output
- If replicating, also consider running “select count(*)” from both IDAA and DB2
- Goals
  - Validate performance improvements
  - Validate correct results from IDAA
  - Get everybody comfortable with setting the ZPARM to “enable with failback”

My test program design

- For every SQL file in input directory:
  - Read SQL
  - For both DB2 and IDAA:
    - Execute SQL, noting elapsed time
    - Save query results to file
    - Summarize one variable from query result (optional)
    - Hash output file
    - Write timing info, output hash value, summarization value, and row count to a DB2 table for later reporting
- Produce report comparing DB2 and IDAA execution of queries

Query results report

OK = query results hash exactly matches
Probably OK = hashes don't match, but row count and value summary differ by less than 0.01%
Probably not OK = row count or value summary differ by more than 0.01%

Row Labels	8 - probably ok	9 - ok	7 - probably not ok	Grand Total
7/4/2014	**14**	**353**		**367**
count	4	327		331
other	10	26		36
7/3/2014	**56**	**283**	**28**	**367**
count	41	263	27	331
other	15	20	1	36
6/28/2014	**12**	**354**	**1**	**367**
count	3	328		331
other	9	26	1	36
6/27/2014	**19**	**348**		**367**
count	9	322		331
other	10	26		36

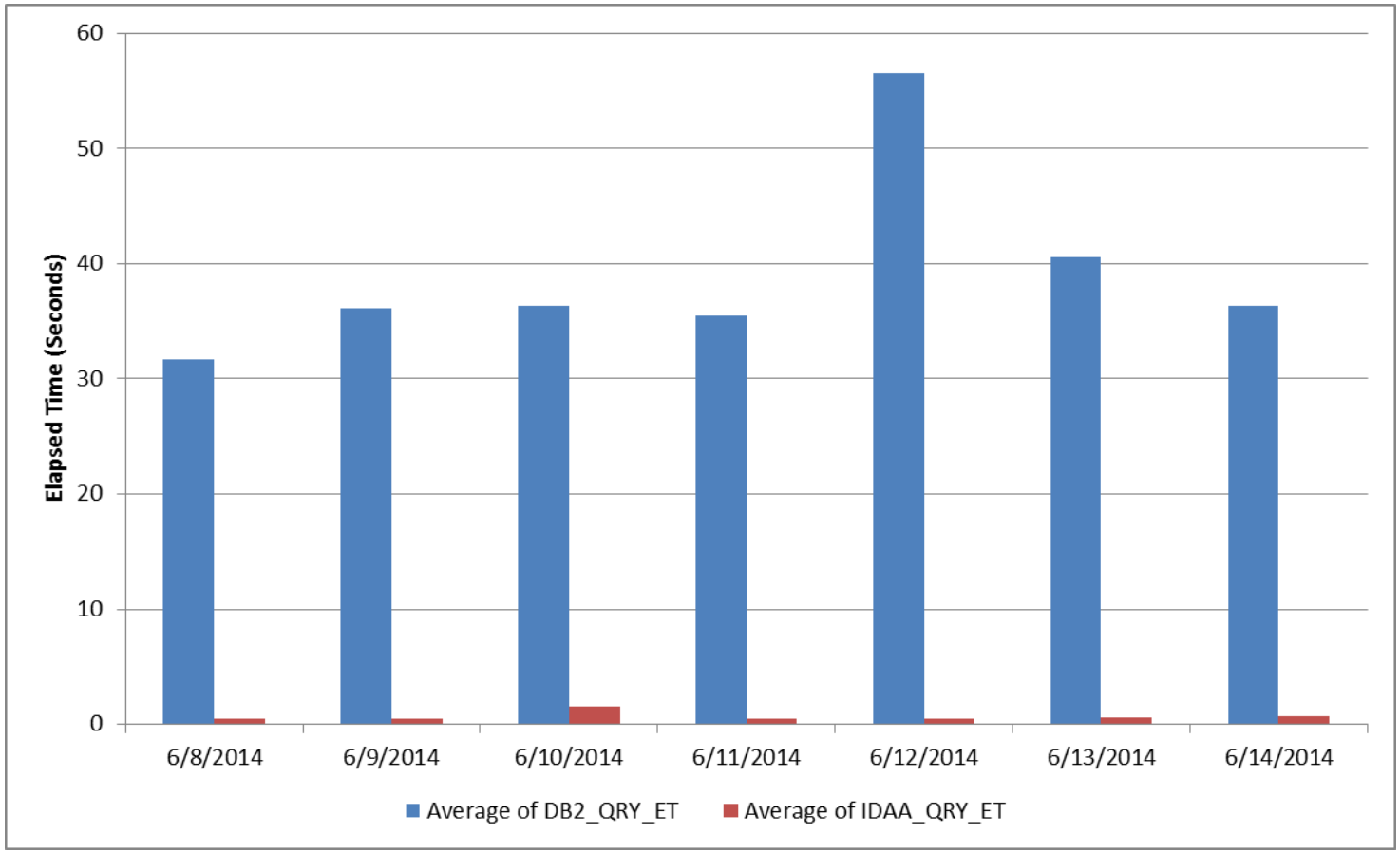
7/3 results differ because ran during the day with higher latency
6/28 result was due to IDAA bug related to organizing keys

Query performance results

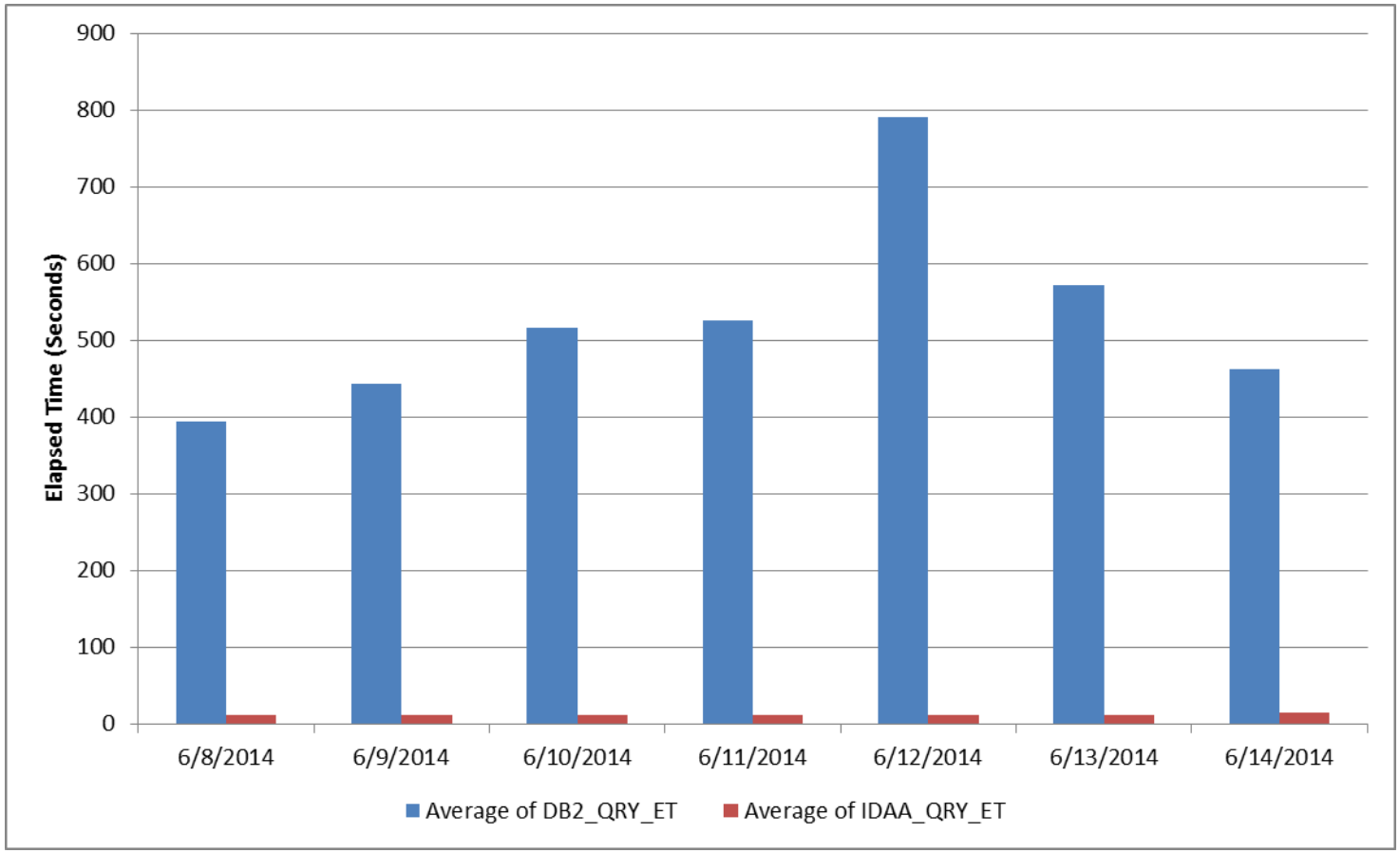
Row Labels	Count of QRY_NM	Sum of DB2_QRY_ET	Sum of IDAA_QRY_ET	Min of IDAA_SPEEDUP_X	Average of IDAA_SPEEDUP_X	Max of IDAA_SPEEDUP_X
7/4/2014	**367**	**13491.024**	**484.574**	**0.0**	**9.9**	**1,330.5**
count	**331**	**3170.863**	**294.085**	**0.0**	**2.6**	**38.6**
001 sub-second	243	30.234	67.717	0.0	0.4	3.8
010 quick	65	226.245	47.625	0.5	5.9	14.2
060 short	11	271.596	21.943	8.5	13.4	22.0
300 medium	10	1408.211	110.381	4.9	19.0	38.6
301+ long	2	1234.577	46.419	24.1	26.1	28.1
other	**36**	**10320.161**	**190.489**	**0.0**	**76.2**	**1,330.5**
001 sub-second	6	1.249	17.749	0.0	0.5	1.9
010 quick	8	34.699	16.821	0.9	3.0	6.7
060 short	7	165.889	23.672	1.3	19.3	82.0
300 medium	6	971.141	50.084	8.4	23.9	42.3
301+ long	9	9147.183	82.163	39.1	270.8	1,330.5

Note that sub-second queries generally take longer in IDAA than DB2, but queries between 1 and 10 seconds may benefit from running in the IDAA. No query over 10 seconds ran faster in DB2.

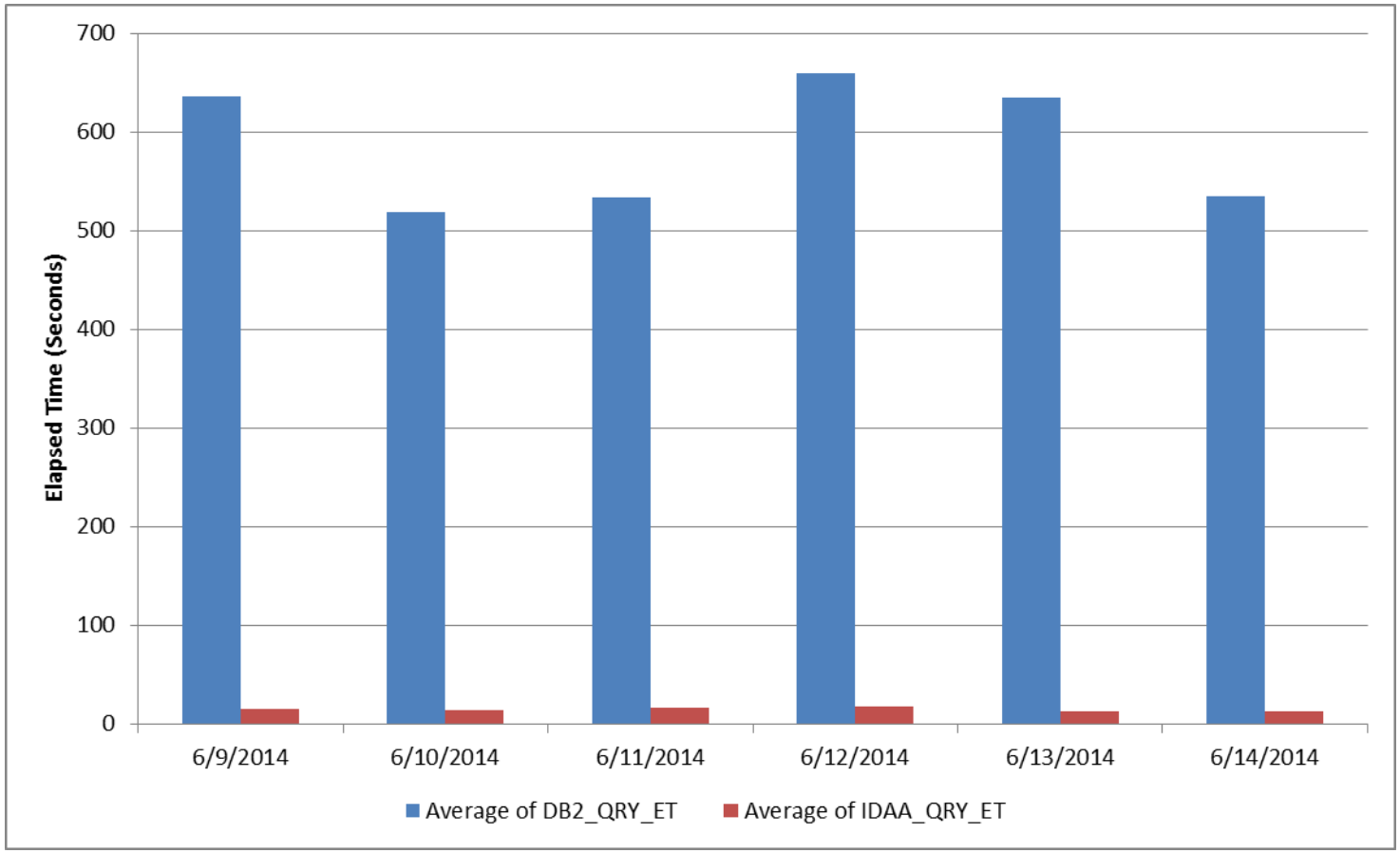
Query ET example 1



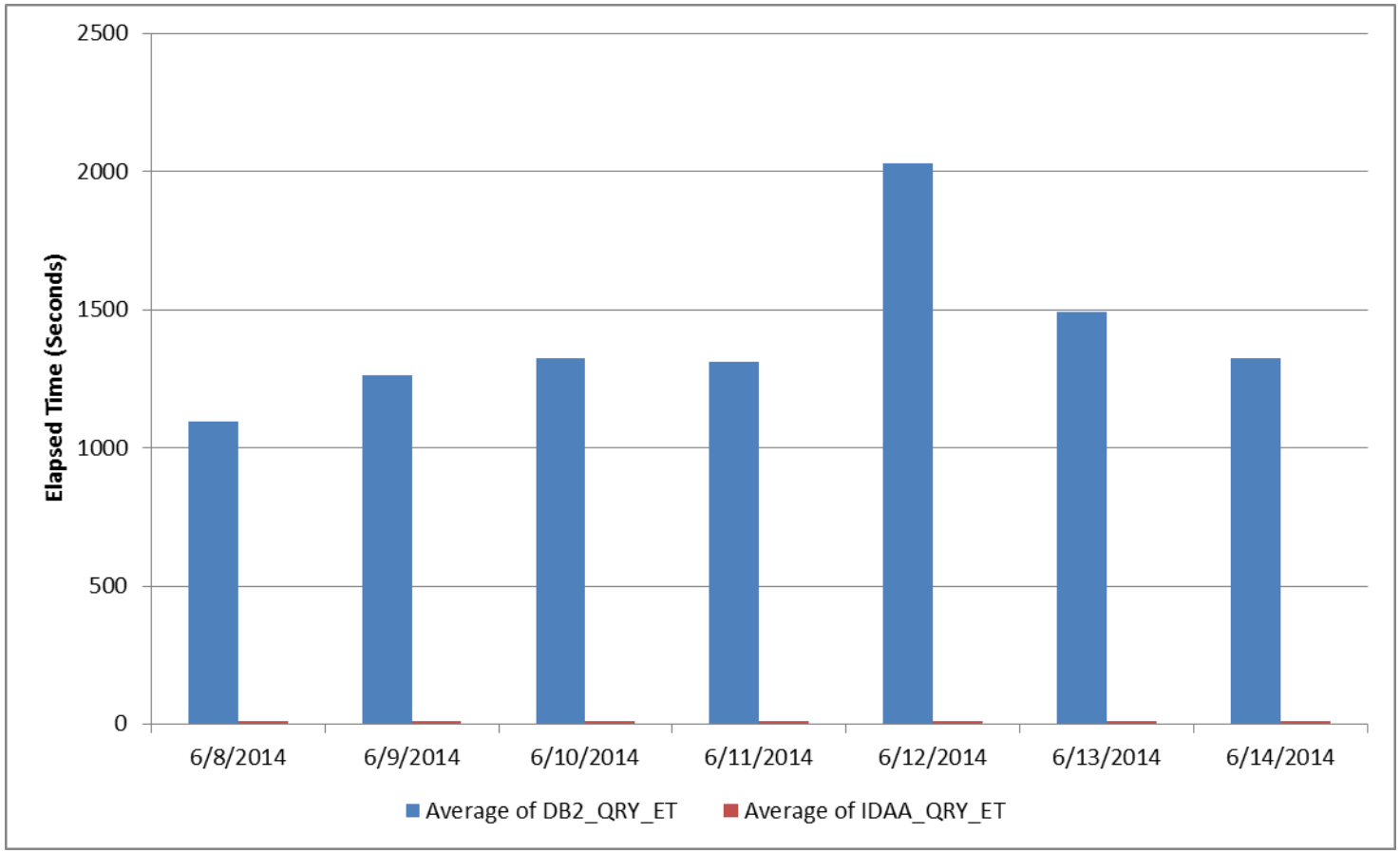
Query ET example 2



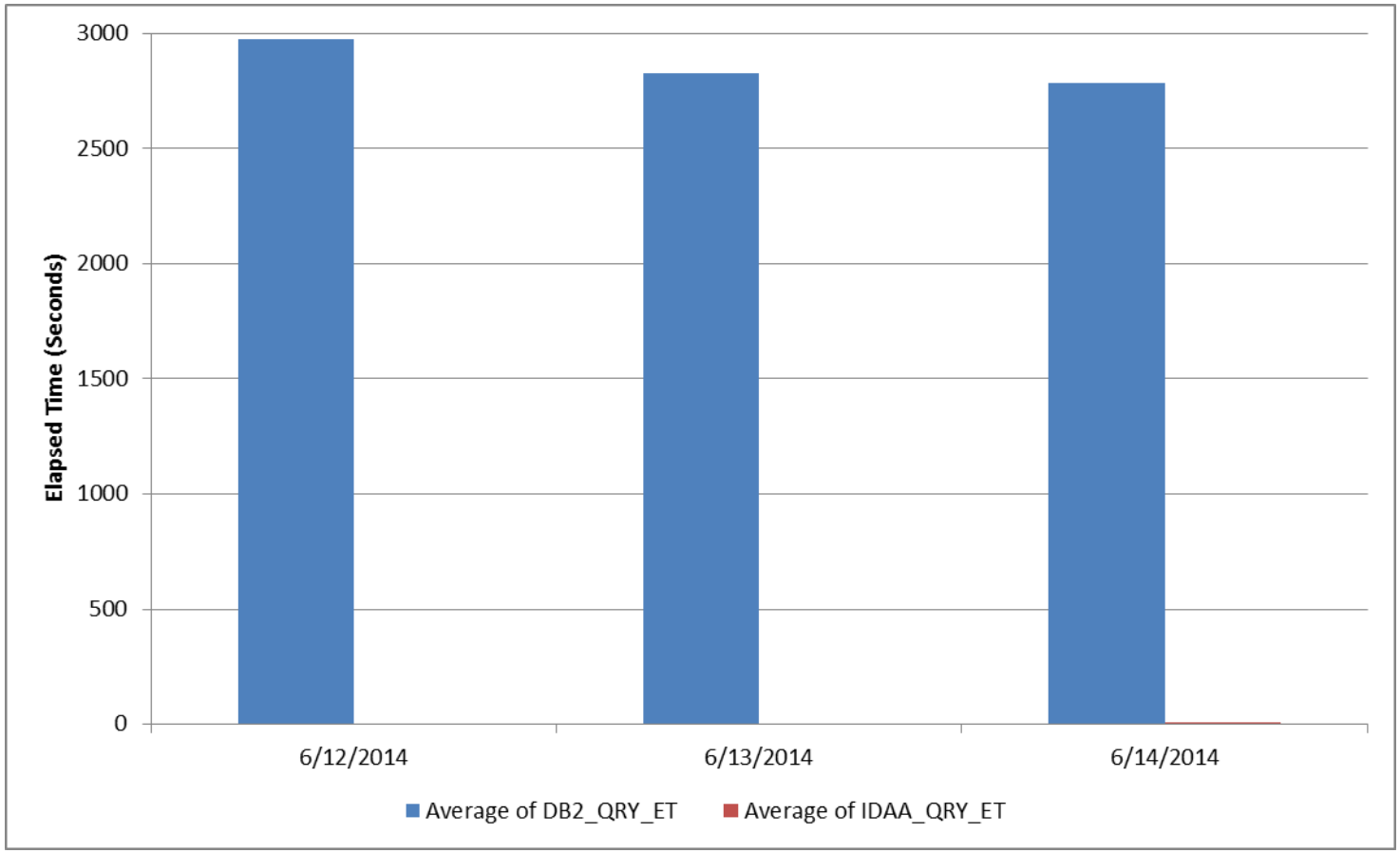
Query ET example 3



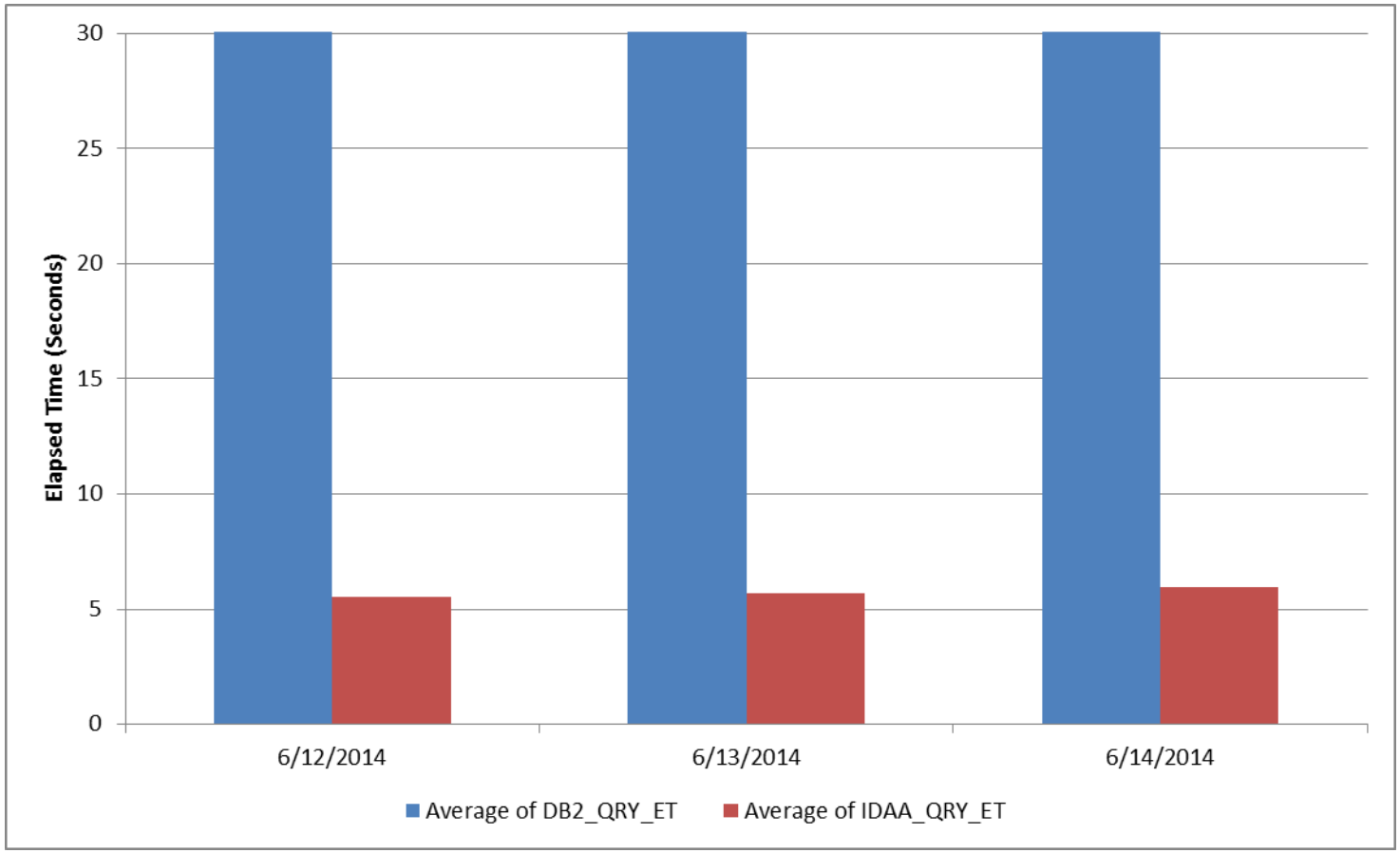
Query ET example 4



Query ET example 5



Query ET example 5, zoomed in 100x



WLM

- There is a link between WLM and IDAA
- Query priority in IDAA set by WLM importance level that was assigned to the query
  - This is the period 1 importance
  - WLM System, 1, 2 = IDAA Critical
  - WLM 3 = IDAA High
  - WLM 4 = IDAA Normal
  - WLM 5, Discretionary = IDAA Low
- Default for IDAA load = Normal
- Default for IDAA groom and genstats = Low
- Hopefully IDAA performance makes this a moot point!

Summary

- IDAA has the potential to revolutionize DB2 performance
- While implementation is transparent* to the applications, there's a fair bit of systems work to do
- IDAA Capacity planning needs to consider replication
  - Replication performance is non-trivial consideration
- Performance gains make the effort worth-while

* Need current DB2 Connect for timestamps

Further questions / discussion?