OPTUM™

CMG
THE DIGITAL TRANSFORMATION ASSOCIATION

CMG March 14 2019
Storage – The Final Frontier of Innovation

IO Benchmarking at Optum

Mark Weber

March 14, 2019

Introduction

Some of the tools and some of the processed used to IO test storage

Talk is a summary of how I do storage performance testing at Optum. Practical speeds and feeds stuff to support the company.

End user experience not selling or pitching anything by the guy doing the work.

1. Why testing
2. Tools overview
3. Strategies, process
4. Example findings
5. Live test running (with Workload Wisdom)


```
fio --filename=<path to test block device>
--direct=1 --rw=randrw --refill_buffers --
norandommap --randrepeat=0 --ioengine=libaio
--bs=8k --rwmixread=70 --iodepth=8 --
numjobs=4 --runtime=60 --group_reporting --
name=8k7030RandwrTest --
output=8k7030RandwrTest-run1.log
```

Mark Weber

West Publishing, Xiotech, UHG

Remember the time, the cube, when I figured out Windows Performance Monitor and the relationship between IO counters

- I had stumbled across Littles Law

Performance has always been fascinating to me

I sit in meetings thinking “I think I can ____ that”

```
fio --filename=<path to test block device>
--direct=1 --rw=randrw --refill_buffers --
norandommap --randrepeat=0 --ioengine=libaio
--bs=8k --rwmixread=70 --iodepth=8 --
numjobs=4 --runtime=60 --group_reporting --
name=8k7030RandwrTest --
output=8k7030RandwrTest-run1.log
```

OPTUM®

Disclaimers

- This is about the process and not our particular results
- We own VI Workload Wisdom (aka Load Dynamix aka LDX)
- I like the LDX and use it all the time but this is not an LDX demo per se
- I look forward to your questions, discussion, alternate techniques and points
- Put this together somewhat last minute. User content not groomed by marketing
- All opinions are my own

Why test anyway?

We all have our reasons

Here are reasons we test:

- Verify vendor claims – “trust but verify”
- Optimize Configs – ex RAID levels, settings
- Understand a system under load – more than a “copy *.*”
- Problem recreation – ex Flash garbage collection
- Full or stressed system test – everything works on an idle system
- To understand before we find out in prod
- We have been burned a few times
- Fun to try to break things
- Test drive, get stick time, practice, and experience.


```
fio --filename=<path to test block device>
--direct=1 --rw=randrw --refill_buffers --
norandommap --randrepeat=0 --ioengine=libaio
--bs=8k --rwmixread=70 --iodepth=8 --
numjobs=4 --runtime=60 --group_reporting --
name=8k7030RandwrTest --
output=8k7030RandwrTest-run1.log
```

Part of a larger decision

Performance testing we do is ultimately part of a larger decision based also on

- Product
- Price
- People
- Politics


```
fio --filename=<path to test block device>
--direct=1 --rw=randrw --refill_buffers --
norandommap --randrepeat=0 --ioengine=libaio
--bs=8k --rwmixread=70 --iodepth=8 --
numjobs=4 --runtime=60 --group_reporting --
name=8k7030RandwrTest --
output=8k7030RandwrTest-run1.log
```

Tools overview

There are many tools

I use Load Dynamix, IOMeter, and FIO the most.

- VI LDX (Virtual Instruments Load Dynamix aka Workload Wisdom)
  - We own this device.
  - Purpose built IO testing device
  - Very powerful, well written, keeps detailed data
- IOMeter
  - I have been using for 20 years.
  - Simple and quick for windows.
  - Well accepted in the real world.
- FIO
  - Linux tool
  - Works but since it's a command line it is more manual.
  - The graphing libraries sent with it
- VDBench
  - Powerful, well used in industry
  - Linux based, windows too.
  - Have not learned, in my queue

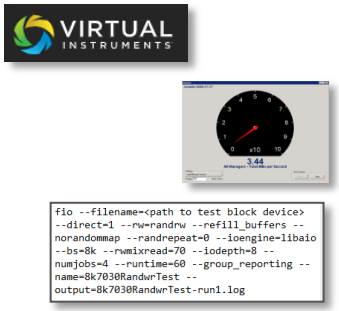

```
fio --filename=<path to test block device>
--direct=1 --rw=randrw --refill_buffers --
norandommap --randrepeat=0 --ioengine=libaio
--bs=8k --rwmixread=70 --iodepth=8 --
numjobs=4 --runtime=60 --group_reporting --
name=8k7030RandwrTest --
output=8k7030RandwrTest-run1.log
```

The juggling act

Lots of balls to keep in the air

- Read or write
- Random or Sequential
- IO size
- Queue depth
- Capacity (GB) tested
- Block or stripe sizes
- Drive type (SSD, spinning)
- Drive speed (SSD, 10K, 7200 RPM).
- RAID level
- Interconnect type
- Adapter settings
- File system tuning
- Switch type
- Switch settings
- MPIO and pathing
- OS disk tuning settings
- Array cache
- Array Processor
- Array internal paths
- Array magic (unique features)
- Compression
- Dedupe
- Encryption
- Garbage collection
- Other/competing work running (workloads, rebuilds, data moves)

Strategies and process



What are we testing?

- Whole system?
  - Isolate port, controller, node, channel, pool or group, disk, memory
- Just a component or setting?
  - Isolate just that setting, set knobs, test, turn knobs, test, repeat

Process

- Setup – 1 to 4h
  - *everything* needs to be configured and verified
    - “Oops I was only testing against 50GB of space”
    - Garbage in garbage out
- Burn in – 2 to 24 hours
- Test, either one by one real time or leave an iteration run over a longer time
- Review, question, consult with local SMEs, ask vendors
- Save files, make or update documentation
- Report findings. Pictures are very powerful

Thing to consider: sample interval

Granularity gives perspective

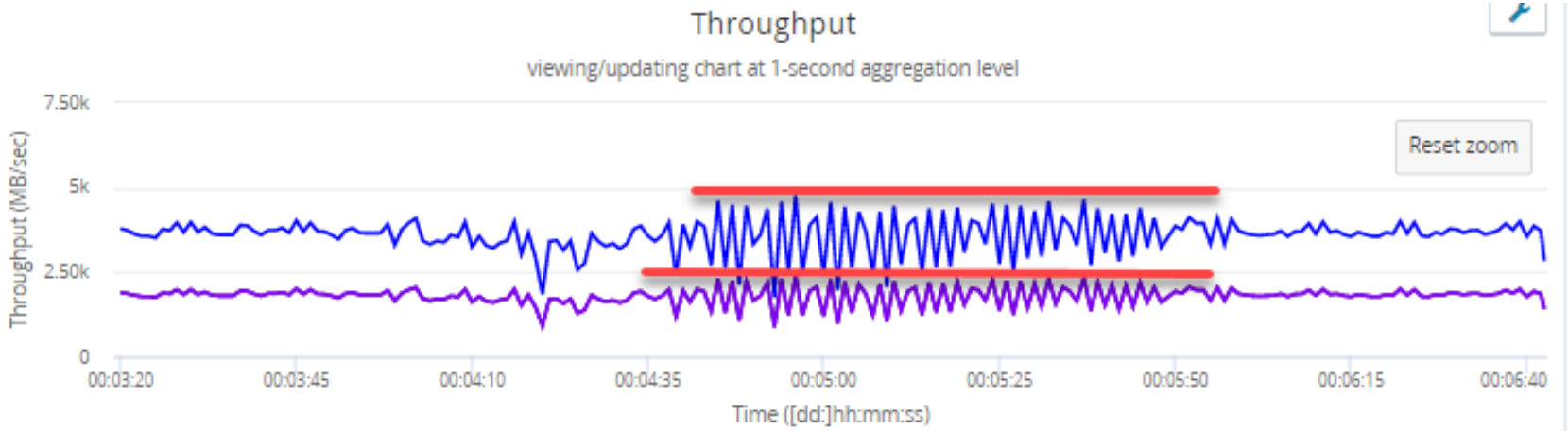
5 min data is common on arrays, can hide a lot in 5 min data.

Observed: oscillation

Severe saw tooth in 100% write workload

Suspect caused by internal cache loading process

Not consistent, one second you get 2,000MB/sec, the next 5,000MB/sec



Name	Visible	Anomalies	Trend Line	Current Value	Maximum	Minimum	Mean	Std Dev
RX Throughput (MB/sec)	☑	☐	☐	1,416.1 MB/sec	2,387.4 MB/sec	874.1 MB/sec	1,799.0 MB/sec	275.1 MB/sec
Throughput (MB/sec)	☑	☐	☐	2,835.8 MB/sec	4,771.8 MB/sec	1,753.8 MB/sec	3,598.4 MB/sec	550.4 MB/sec
TX Throughput (MB/sec)	☑	☐	☐	1,419.7 MB/sec	2,384.4 MB/sec	879.7 MB/sec	1,799.4 MB/sec	275.4 MB/sec

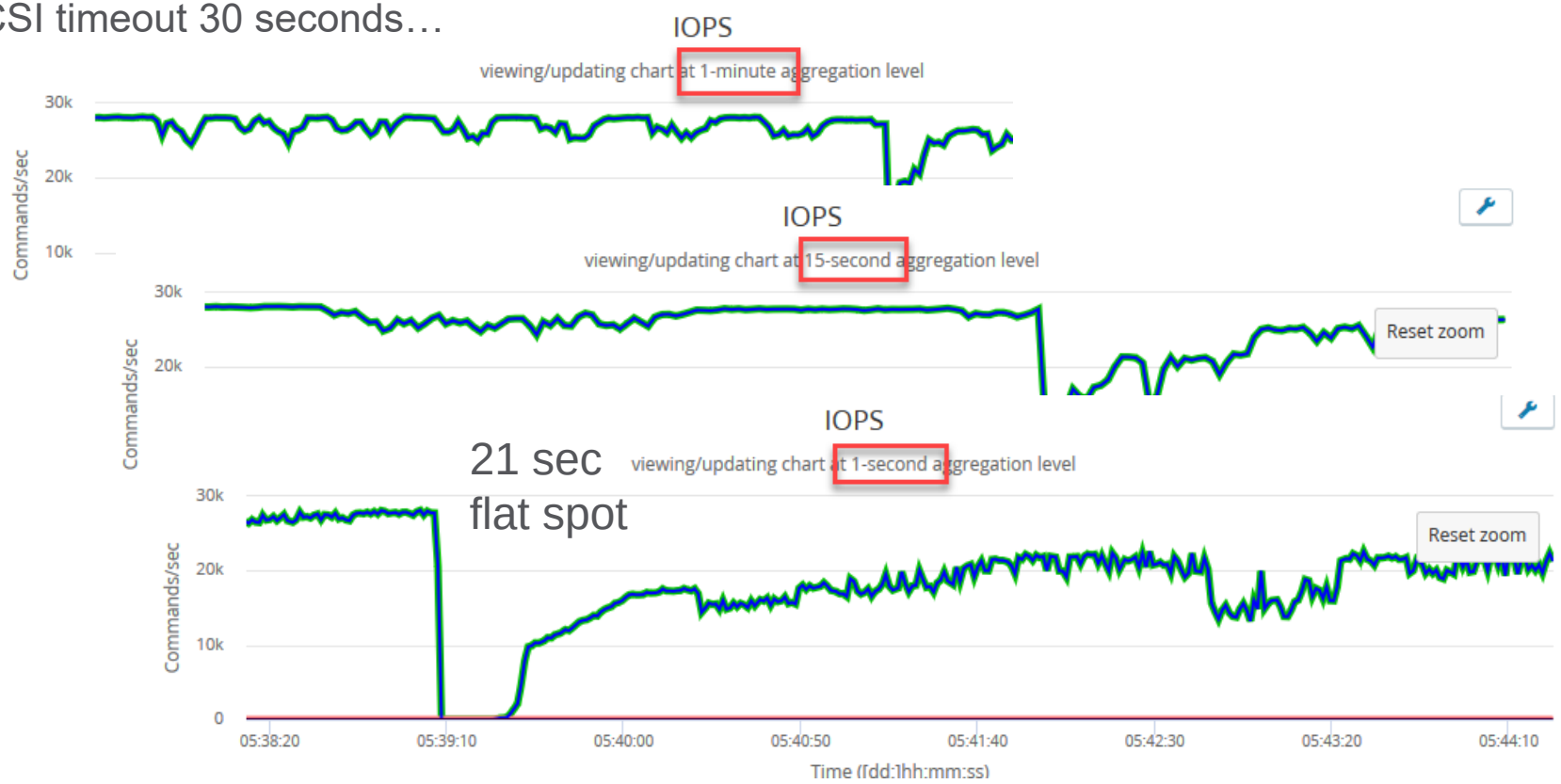
Observed: upgrade process

What does an array upgrade really look like?

(same chart used for time sample comparison)

I just started looking at this level. Have only one data point so far

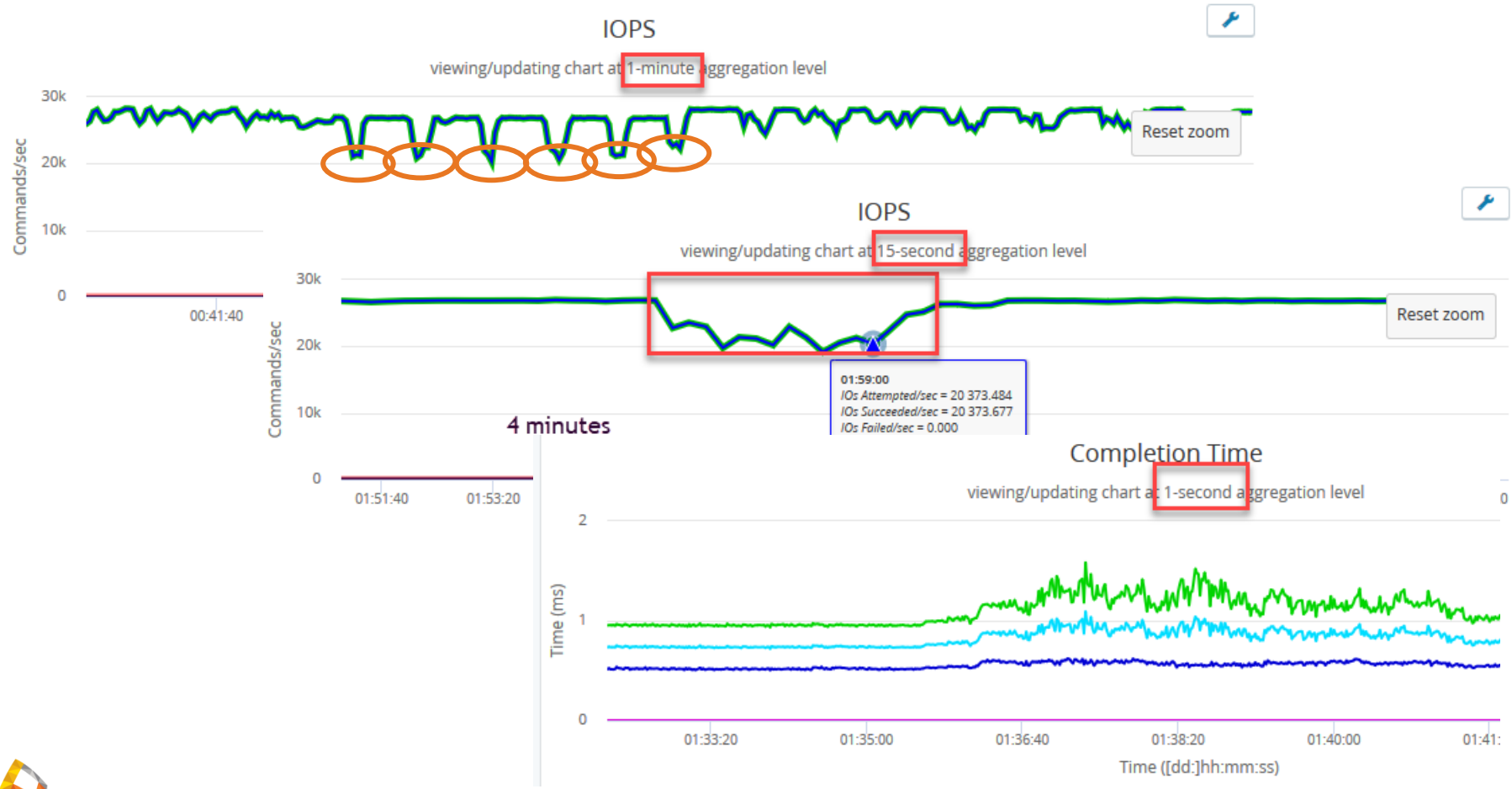
SCSI timeout 30 seconds…



Observed: drive insert

“You won’t be able to see it”

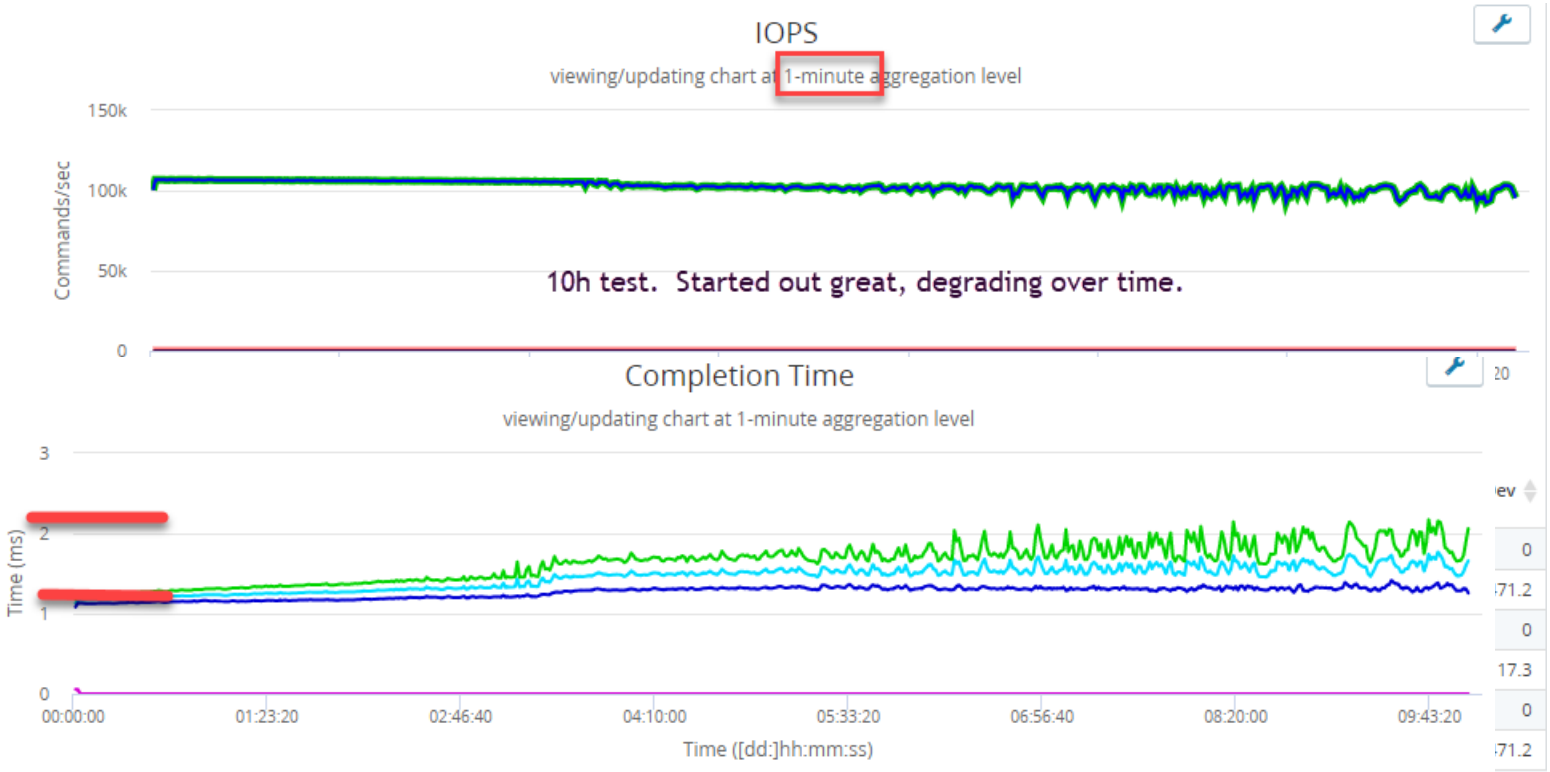
Vendor claims we wont see the array add in expansion drives. 6 drives added



Observed: long running UCI.v2 test

I wouldn't have guessed this would happen

The IOPS degradation does not look terrible but the response time increase is more concerning. In the queue to re-run this over a weekend. Also would like to rerun with a different tool



Observed: maxed out SAS paths to disk

One lab array maxed out SAS paths on proposed solution

We would have bumped into this limit in production

Adjusted bill of material proposal

Time – I will take all you got

It seems like there is never enough

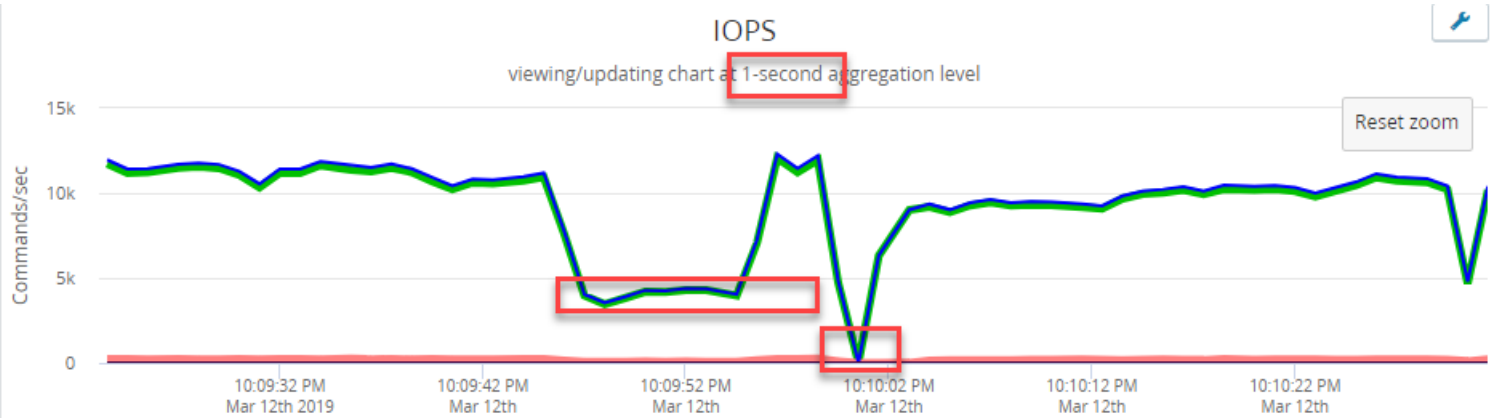
- ✓ The time needed to do benchmarking right is usually underestimated
  - ✓ I might quote a half week or a week, but in my cartoon bubble I am thinking two weeks minimum
- ✓ Ad-hoc exploratory what-if testing burns lots of time
- ✓ Double check the tool, trust but verify (its just software)
  - ✓ Verify littles law
  - ✓ Does it look right, feel right, seem like it makes sense?
- ✓ When is good enough good enough? Perfect the enemy of good
- ✓ Documentation
  - ✓ Critical to find things later. Critical to identify the infrastructure/test configuration/goals
  - ✓ I type a lot into LDX Test Run Description box.
    - ✓ Updatable on the fly
  - ✓ Coding, key words, Vendor, Ports, capacity, length etc. Search, sort, filter
  - ✓ My LDX has 1090 completed tests in it
  - ✓ LDX is my database

Documentation

The easiest to skip or forget

Goal: condense everything down to a bullet or sound bite

Markup – help the user. I use Snagit. Help the user see the point



Name	Visible	Anomalies	Trend Line	Current Value	Maximum	Minimum	Mean	Std Dev
IOs Aborted/sec	☑	☐	☐	0	5	0	0.07	0.6
IOs Attempted/sec	☑	☐	☐	10,396	12,299	3	9,340.9	2,780.2
IOs Failed/sec	☑	☐	☐	212	269.3	0	178.3	63.9
IOs Pending	☐	☐	☐	5	8	3	5.1	1.0
IOs Retried/sec	☐	☐	☐	0	2	0	0.06	0.3
IOs Succeeded/sec	☑	☐	☐	10,183	12,057	4	9,162.6	2,723.5

Live demo

LDX in the lab

1. Testbeds
2. Start a workload test on an AFA running my UCI.v2 test
3. Custom configure test
4. Start
5. Update comments on the fly
6. Look at a historical iteration run
7. Look at a report
8. Check back on running test, zoom in
9. Look at All Charts | SCSI Details

Thank you.

Contact information:

Mark Weber

Storage Engineer
Tel: 952-833-7373
Email: mark_weber@optum.com

OPTUM®