HITACHI
Inspire the Next

Testing in minimum time with ivy

"seen enough & stop"

April 4, 2016

Allart Ian Vogelesang
ian.vogelesang@hds.com +1 408 396 6511

Hitachi Data Systems

What does it mean – a “valid” measurement

HITACHI
Inspire the Next

- Valid results are *repeatable*
  - If you run the test again, within specified +/- experimental error, you will get the same result again.
  - Valid, repeatable, results are for steady-state, sustainable conditions
- If the workload / subsystem are not steady-state, you can’t make a valid measurement.
  - After imposing a workload on the subsystem, we need to wait for the behaviour to settle down into a steady state before we start measuring

[Go] statement measure=on is "seen enough & stop"

HITACHI
Inspire the Next

- measure = on
- accuracy_plus_minus = "5%"
  - Default is "5%".
- confidence = "95%"
  - How confident you need to be that your measurement falls within the specified plus or minus range around the long term average that you would get measuring forever.
  - Default is "95%"
  - Ivy has a menu of 11 specific pre-loaded confidence values that you pick from.
    - 50%, 60%, 70%, 80%, 90%, 95%, 98%, 99%, 99.5%, 99.8%, and 99.9%
    - http://en.wikipedia.org/wiki/Student%27s_t-distribution

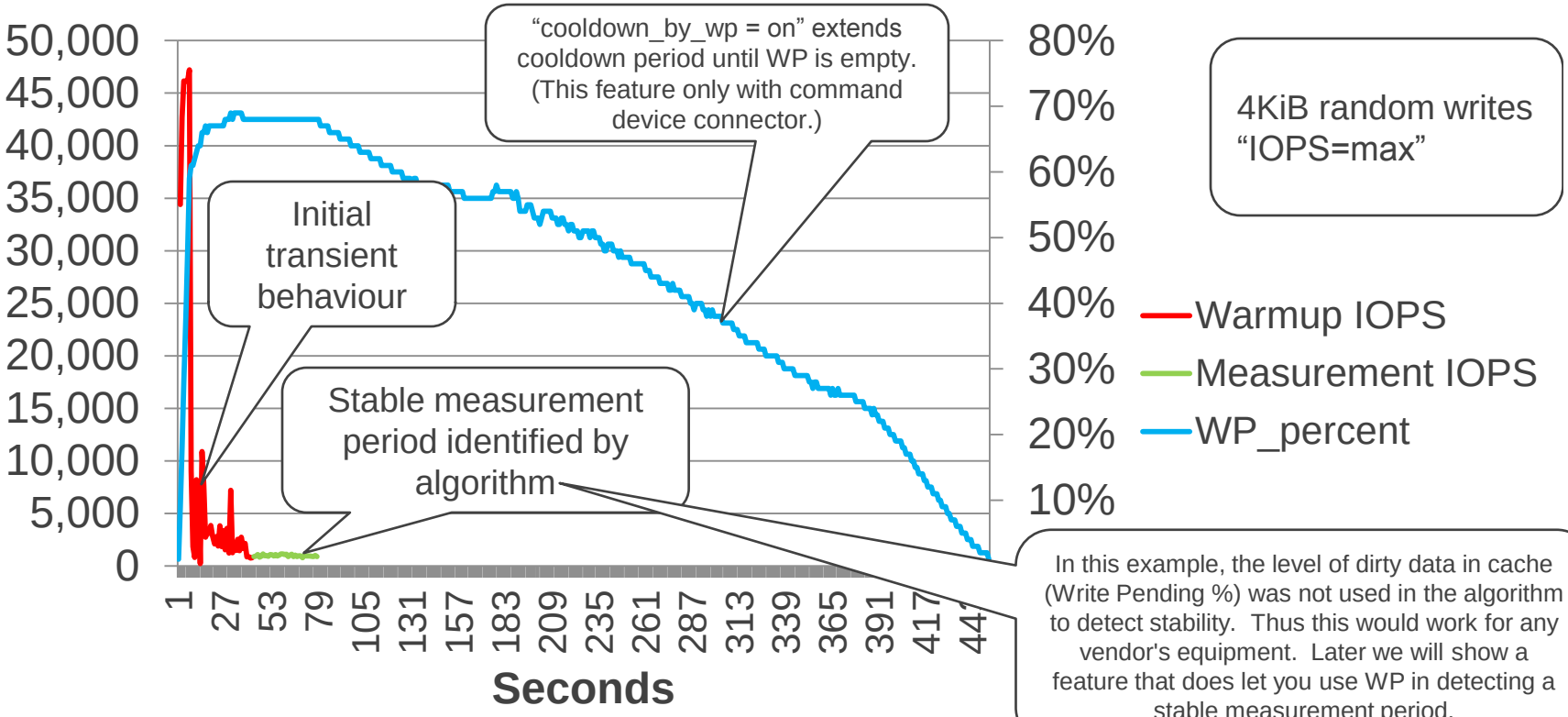
Standard statistical formulas need a tweak

HITACHI
Inspire the Next

- The standard formulas are for when you want to estimate what the mean (average) measurement would be if you sampled across the entire population, a huge number of samples.
- With the standard formulas, a set of samples is taken, each sample a random sample, meaning the selection of each sample is at random, unrelated to how any other sample was collected.
- In ivy, we are sampling *consecutive* subintervals, and there is a correlation from subinterval to subinterval.
- So my rough observation from running ivy a few times is to specify accuracy_plus_minus twice as tightly, e.g. say +/- 1 % to get +/- 2% accuracy

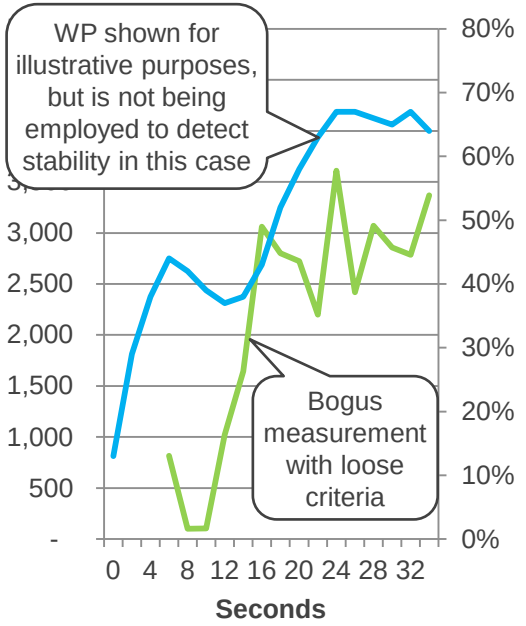
Detecting stability – by observing only IOPS/service time

HITACHI
Inspire the Next

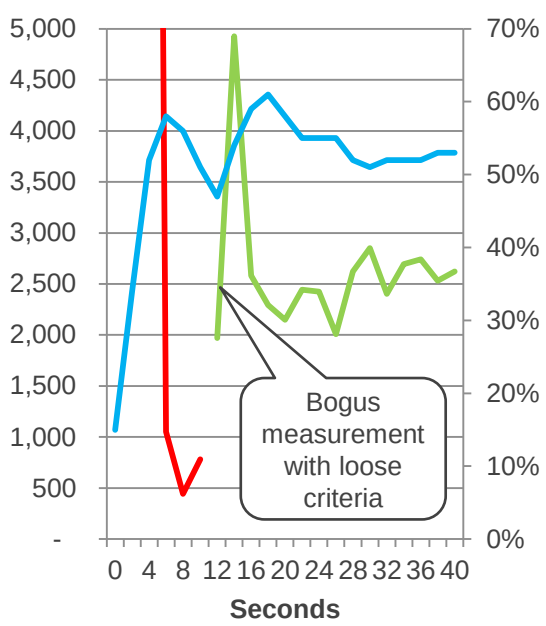


Even without command device, algorithm is effective

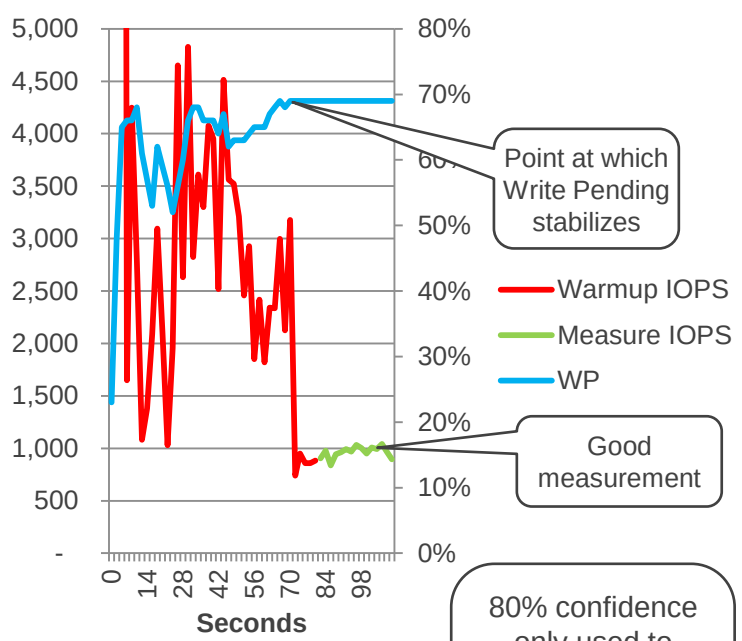
HITACHI
Inspire the Next



Accuracy +/- 20%
Confidence 80%



Accuracy +/- 10%
Confidence 80%

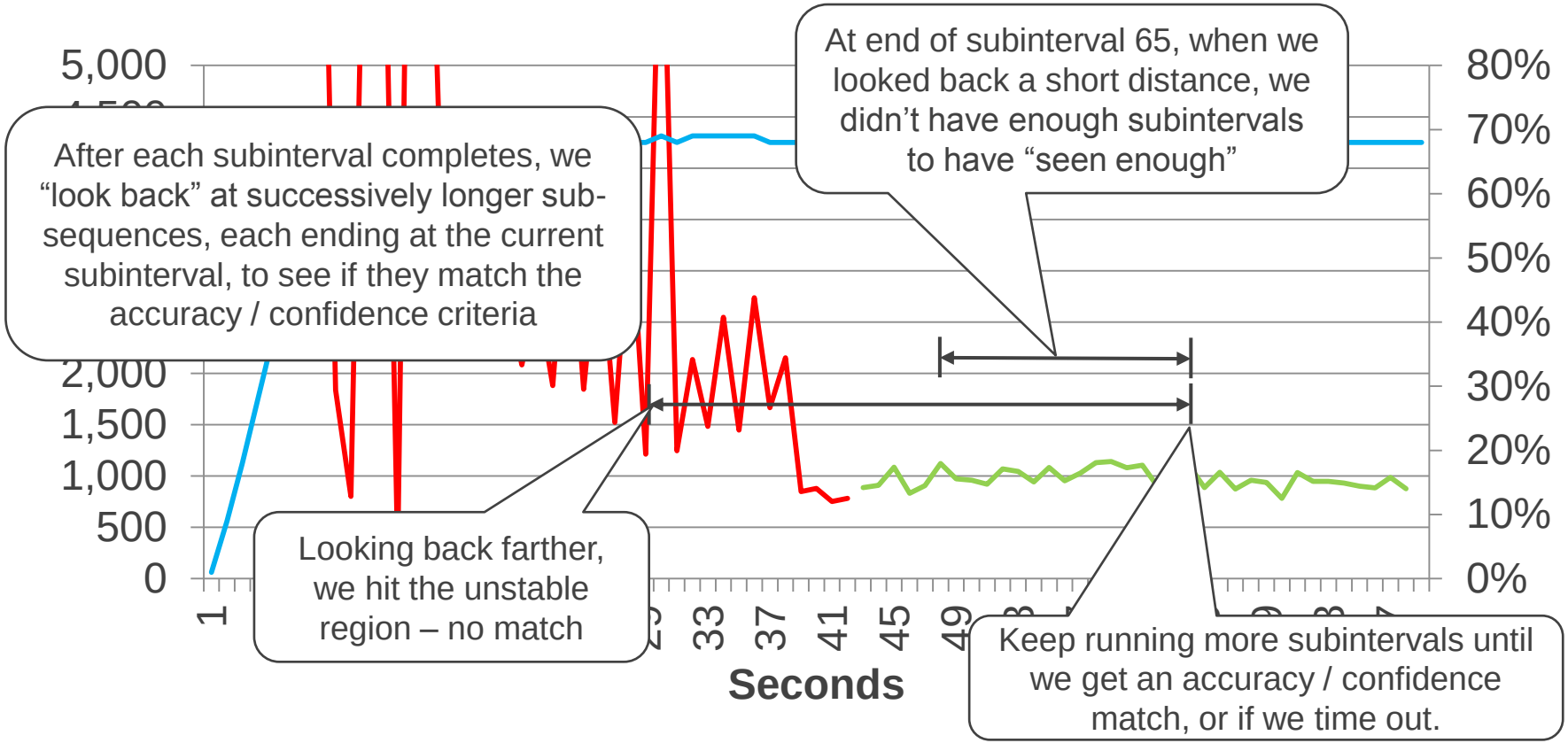


Accuracy +/- 2%
Confidence 80%

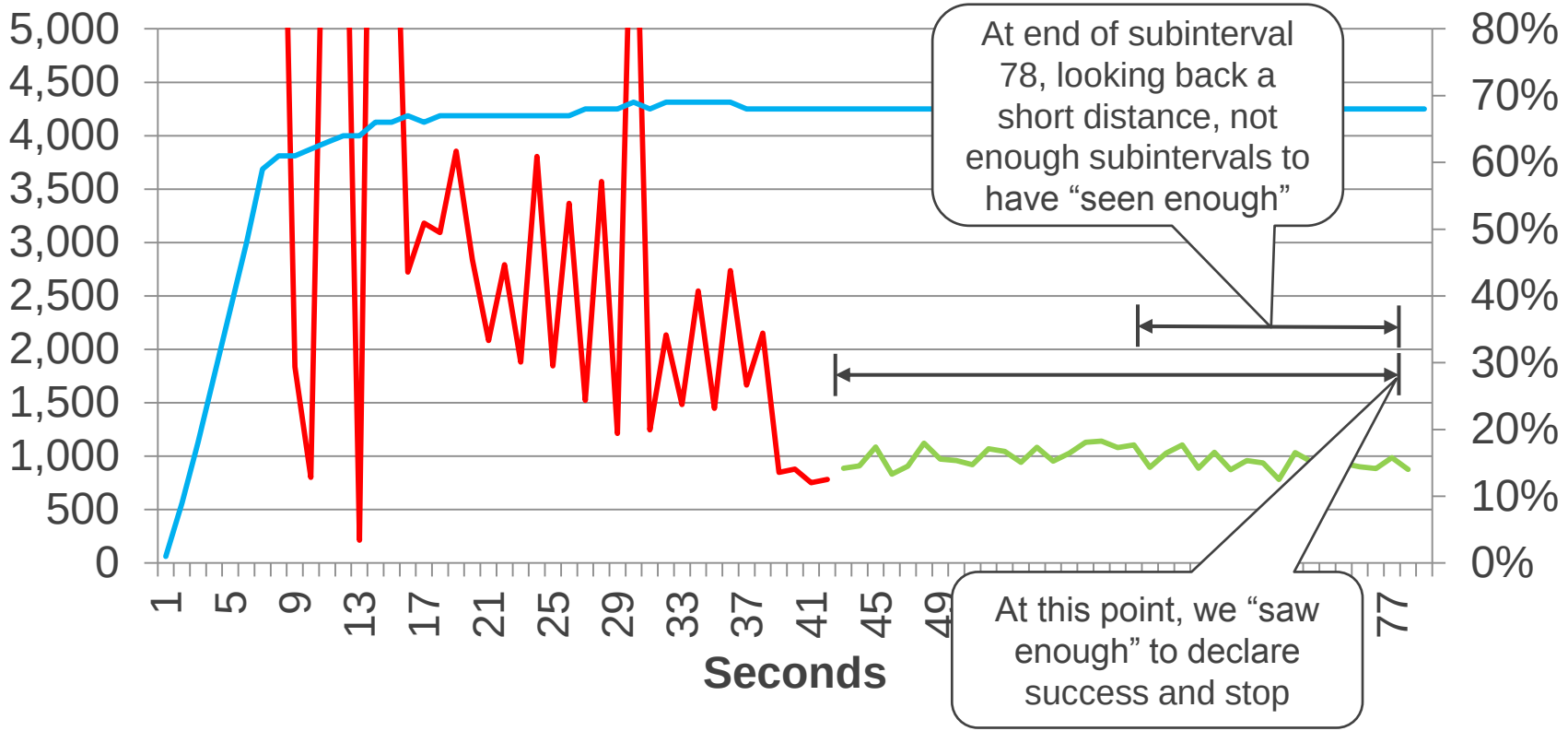
80% confidence only used to show algorithm failure. Normally we would use 95% confidence)

Have not “seen enough”

HITACHI
Inspire the Next



“saw enough & stopped”



We can get good results from just IOPS, service time

HITACHI
Inspire the Next

- With `measure=on`, if we set the `accuracy_plus_minus` and `confidence` parameters reasonably tight, we can detect when the subsystem has “settled down” and the workload has stabilized.
- When we say the subsystem has “settled down”, what we generally mean is that Write Pending has settled down.
  - Detecting stability just looking at IOPS / service time means we need to be sufficiently careful (precise) so that the instability where WP hasn’t yet stabilized is enough to prevent “matching” on the `accuracy_plus_minus` and `confidence` parameters.
- Thus `measure=on` should work just fine for any vendor's equipment, even without a proprietary "connector" to fetch internal performance data from the storage product.

Faster / more robust stability detection with command device

HITACHI
Inspire the Next

- Ivy automatically discovers and uses a command device to the target subsystems(s), if one is found on a test host and the command device connector is available.
- "**`measure=on`**" has Write Pending-based stability criteria you can use with a cmd dev.
  - **`max_wp = "2%"`** - default "100%"
    - A subinterval sequence will be rejected if WP is above the limit at any point in the sequence.
    - Use this for read tests to ensure WP is empty during the test.
  - **`min_wp = "68%"`** - default "0%"
    - A subinterval sequence will be rejected if WP is below the limit at any point in the sequence.
    - Use this for write tests to ensure WP is full during the test.
  - **`max_wp_range = "3%"`** - default "3%"
    - A subinterval sequence will be rejected if WP varies up and down by more than the specified (absolute) amount at any point in the sequence. `max_wp_range="3%"` matches from 0% to 3%, as well as from 67% to 70%.
    - Use this in general all the time so you reject periods with major movement in Write Pending.

The `cooldown_by_wp` setting – optimize elapsed time

HITACHI
Inspire the Next

- Whether or not you use `measure=on`, `cooldown_by_wp` is enabled if a command device is discovered and the command device connector is available.
- The default is `cooldown_by_wp=on`
- Set `cooldown_by_wp=off` when it is valid to carry forward WP from one test step to the next.
  - This can speed up the next test step tremendously if the next step doesn't stabilize until WP is full, AND if both steps place the SAME things into WP. (An example follows in a later slide.)
  - If you are switching, say, from testing 4 KiB blocks to testing 8 KiB blocks, then after you make two trials, one with `cooldown_by_wp` set to `on` and one set to `off`, you can see if carrying WP over from the previous step with the different blocksize makes a difference. After seeing and thinking about it, will know if `cooldown_by_wp=off` is OK.

Designing your test to run in mimimal time

HITACHI
Inspire the Next

- Use `min_wp / max_wp / max_wp_range` with `measure=on` to measure only when wp has stabilized.
  - Lets you use looser accuracy / confidence parameters for a faster test if you don't need a meticulously accurate result, without losing detection of WP stabilization.
- With or without `measure=on`, use `cooldown_by_wp` where appropriate to carry WP contents forward from one test step to another, to minimize warmup time.
- `cooldown_by_wp=off` uses a single cooldown subinterval

warmup_seconds, measure_seconds, subinterval_seconds

HITACHI
Inspire the Next

- warmup_seconds – default = 5
- measure_seconds – default = 60
- With measure=off, the warmup and measurement periods are fixed.
- With measure=on, these values represent minimum periods during automatic detection of a valid measurement.
- subinterval_seconds default is 5
  - Don't make this shorter, but if you are running really long test steps, you could use longer subintervals to reduce the volume of csv output.

Specifying `measure=on – source=workload`

HITACHI
Inspire the Next

- **`source`**`=workload` is always available – the test host view of I/O timing.
  - Fully explained in "programming ivy" material.
  - (There is also `source=RAID_subsystem`, not shown here.)
- I/O events are accumulated over a subinterval, and rolled up according to category
  - **`category`** `= overall, read, write, random, sequential, random_read, random_write, sequential_read, sequential_write`
  - Within these categories, there are also service time / response time histograms that are kept and shown in ivy output csv files.

accumulator_type, accessor

HITACHI
Inspire the Next

- For each I/O, the service time is posted into one "accumulator", the bytes transferred into another, and if `IOPS=max` is not set, we post "response time" into another.
  - **`accumulator_type`** = `bytes_transferred, service_time, response_time`
  - Service time is from I/O launch to I/O completion. Response time is from scheduled I/O start to I/O completion.
    - I/Os will not launch at the scheduled time if there is no "tag" available, so response time can be longer than service time.
    - `IOPS=max` sets every I/O's scheduled time to zero, in which case "response time" is not posted.
- Retrieve data from a (rolled up) accumulator using an accessor
  - **`accessor`** = `avg, count, min, max, sum, variance, standardDeviation`
  - Use avg for `service_time, response_time` which are in seconds.
  - Use sum for `bytes_transferred`. Divide by `subinterval_seconds` to get bytes per second.
  - Divide `service_time`'s count by `subinterval_seconds` to get IOPS.

focus_rollup

HITACHI
Inspire the Next

- The detection of a valid measurement is performed at the granularity of the `focus_rollup`.
  - A measurement subinterval sequence constitutes a valid measurement if for all instances of the focus rollup the validity criteria were met.
- The default is `focus_rollup="all"`, which looks for an overall valid measurement across all workload threads across all test hosts.
- If you have created a rollup by port, by saying
  - `[CreateRollup] "port";`
  - then you can specify `focus_rollup="port"`, and then for `measure=on`, ivy will look for a simultaneous valid measurement period on each instance of the port rollup.

An example of running a test in minimum time

HITACHI
Inspire the Next

- Here we are doing a 4K random I/O run with IOPS=max, and we are showing the IOPS we get for 0%, 25%, 50%, 75%, and 100% writes.
- We start out by doing the 0% read step, with **`min_wp`**`="68%"` and **`cooldown_by_wp`**`=off`
  - It takes some time to warm up because WP needs to fill up and stabilize.
  - The measurement itself takes a little extra time as the IOPS is not quite as stable when the subsystem is in emergency destage, and the subsystem takes some time to "switch gears" and for the IOPS to become stable once WP is full.
- The 25% reads, 50% reads – `cooldown_by_wp=off`
  - The previous step leaves WP full, and we stabilize very quickly. It's OK as what we are putting into WP (4K random writes) is the same thing that is draining out (4K random writes to the same LDEVs)
- The 75% reads step – cooldown_by_wp= on
- The 100% read step – could use **max_wp=**"2%", but not necessary (previously cooldown_by_wp=on)

Example – minimum time % read test - output

HITACHI
Inspire the Next

Warmup includes filling Write Pending and stabilizing after WP is full

cooldown_by_wp = off

Test Name	Step Number	Step Name	Start	Warmup	Duration	Cooldown	Generator type	blocksize	maxTags	IOPS	fractionRead	Overall IOPS
read_vs_write	step0001	read_0	4/22/2015 11:30	02:05.0	00:50.0	00:05.0	random_steady	4 KiB	16	max	0	893.20
read_vs_write	step0002	read_25	4/22/2015 11:31	00:10.0	00:55.0	00:05.0	random_steady	4 KiB	16	max	0.25	1,128.00
read_vs_write	step0003	read_50	4/22/2015 11:32	00:10.0	00:50.0	00:05.0	random_steady	4 KiB	16	max	0.5	1,533.86
read_vs_write	step0004	read_75	4/22/2015 11:33	00:10.0	00:50.0	06:45.0	random_steady	4 KiB	16	max	0.75	2,077.54
read_vs_write	step0005	read_100	4/22/2015 11:41	00:15.0	00:50.0	00:05.0	random_steady	4 KiB	16	max	1	3,492.94

cooldown_by_wp = on

Conclusion

HITACHI
Inspire the Next

- "Seen enough and stop" is effective to perform tests in minimum time.
  - Without a command device, we need tighter accuracy/confidence settings and thus a bit longer test time.
  - With a command device, we directly observe when WP stabilizes.
- Standard small sample set math is effective with a 2x fudge factor.
  - Distribution may not be a "normal distribution", but using the +/- accuracy and confidence % specifications for a normal distribution ("student's t-distribution) is effective to detect stability / establish needed length of measurement to adapt to the degree of instability in the measured results from subinterval to subinterval.
  - Because there is some correlation from subinterval to subinterval, achieved plus/minus accuracy is somewhat looser than specified. Specify 2x tighter accuracy to be sure.

HITACHI
Inspire the Next

Questions and Discussion

Hitachi Data Systems

HITACHI
Inspire the Next

Thank You

Hitachi Data Systems