Benchmark every week? Why? Are you crazy?

Cheryl Watson
Watson & Walker, H&W Computer Systems
cheryl@watsonwalker.com
www.watsonwalker.com, *www.hwcs.com*
Session 19155

#SHAREatl

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

Who we are

- Watson & Walker, Inc.
  - Founded in 1987 by Cheryl Watson & Tom Walker; Frank Kyne joined in 2014
  - ***Cheryl Watson's Tuning Letter*** quarterly, published since 1991
  - ***Cheryl Watson's System z CPU Chart***
  - Public and private classes and consulting on z/OS new features, WLM, performance, Parallel Sysplex, high availability, software pricing, outsourcing contract reviews, and chargeback
  - ***GoalTender*** – WLM policy analyzer
  - ***BoxScore*** – After-the-fact benchmark
- H&W Computer Systems
  - Leading provider of mainframe software solutions since 1979
  - Serves customers worldwide including Global 500 customers
  - Has built a reputation for top-notch service
  - Provides business software like industry-leading **SYSB-II** for VSAM filesharing between CICS and batch, among other products

New Partnership

- In May 2015, Watson & Walker joined with H&W Computer Systems to develop a new version of BoxScore called BoxScore II

- Slides used in this presentation are from this new product

Benchmark every week?

- “Cheryl Watson, close watcher of all things performance and capacity related, recommends that installations perform a benchmark every week. Is she crazy or is she on to something? We know that many, if not most, sites perform a benchmark after a CPU upgrade to ensure that the new machine is giving them what they paid for. But every week? Cheryl describes her reasons and gives suggestions for resolving the underlying issues, as well as tips for how to reduce RNI to reduce MIPS. She'll be using examples from the new BoxScore II tool jointly developed with H&W Computer Systems.”

Agenda

- What is benchmarking?
- Types of benchmarks
- Why benchmark?
- How does RNI affect benchmarks?
- Tips for reducing RNI
- Why benchmark weekly?
- Special benchmarks on demand

What is benchmarking?

- Comparing performance of one execution to another based on a set of measurements
- Common benchmarks:
  - Performance/handling of new cars from different makers
  - Comparison of stock performance
  - Validate speed/capacity of new computer

Types of benchmarks

- Typical IT benchmark methods:
  - Synthetic jobs
  - Representative programs in a stand-alone or dedicated environment
  - Representative programs in a shared environment
  - Actual stable workloads (BEST)

Types of benchmarks

- Synthetic (generated)
  - Simulates same amount of CPU time, number of I/Os, working set size (locality of storage reference), number of concurrent transactions or jobs (e.g. soaker jobs)
  - Much more difficult today than 20 years ago
  - WORST method to determine change

Types of benchmarks

- Representative workloads on stand-alone machines
  - Create typical job streams or transactions that are similar to actual workload
  - Currently used by IBM to produce their Large Systems Performance Reference (LSPR) ratings
  - Biggest problem is finding stand-alone machine time
  - Second problem is ensuring that job streams and transactions are still typical of current actual workload
  - Better solution than synthetic or on shared machine

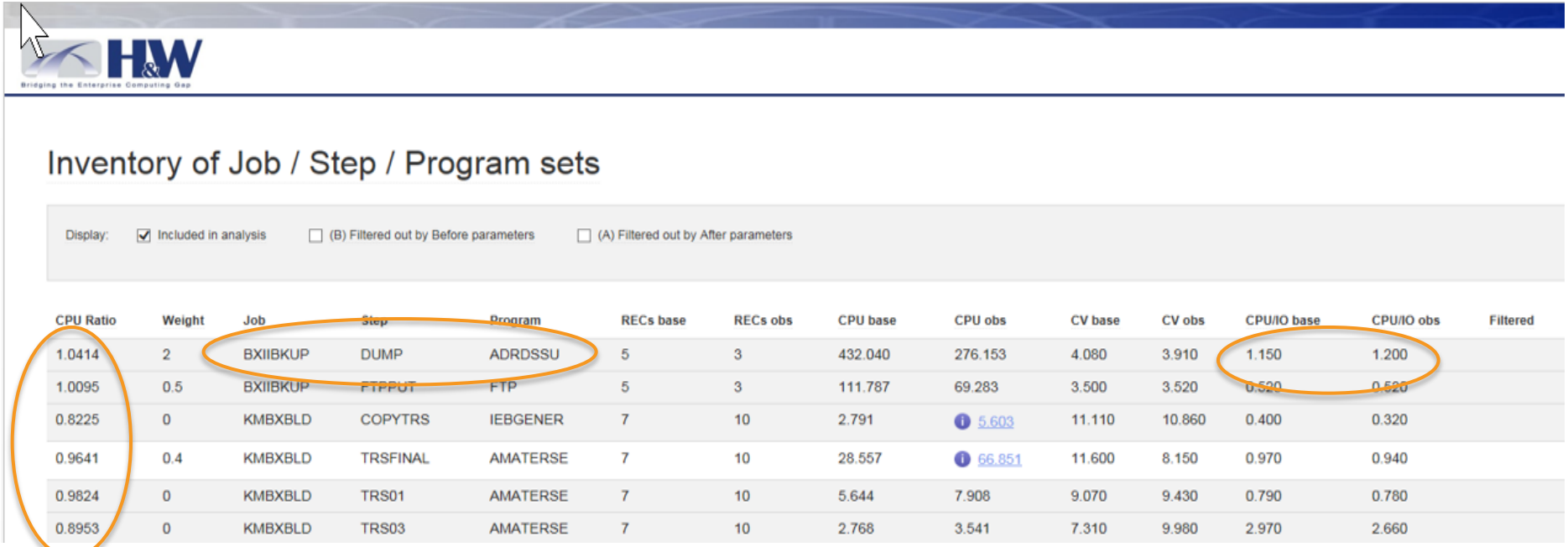
Types of benchmarks

- Representative workloads on shared machine
  - Biggest problem is finding periods of time where the overhead of other LPARs or other work is minimal or similar
  - Second problem is ensuring that job streams or transactions are still typical of current actual workload
  - This is what most installations use today; but their job streams might be 20 years old!
  - Most representative workloads don't match today's actual workloads (can old IEBGENER step compare to a Java step?)

Types of benchmarks

- Actual stable workloads on shared machines
  - Also called dynamic or ‘after-the-fact’ benchmarks
  - Biggest problem is finding stable work because of the current variability of CPU usage; but this can be done using CPU per I/O or CPU per transaction techniques
  - Solves the problem of ensuring that job streams are typical of current actual workload because they ARE the current workload
  - This provides the most accurate benchmarking and can be done after the fact using existing SMF data

After the fact benchmarks

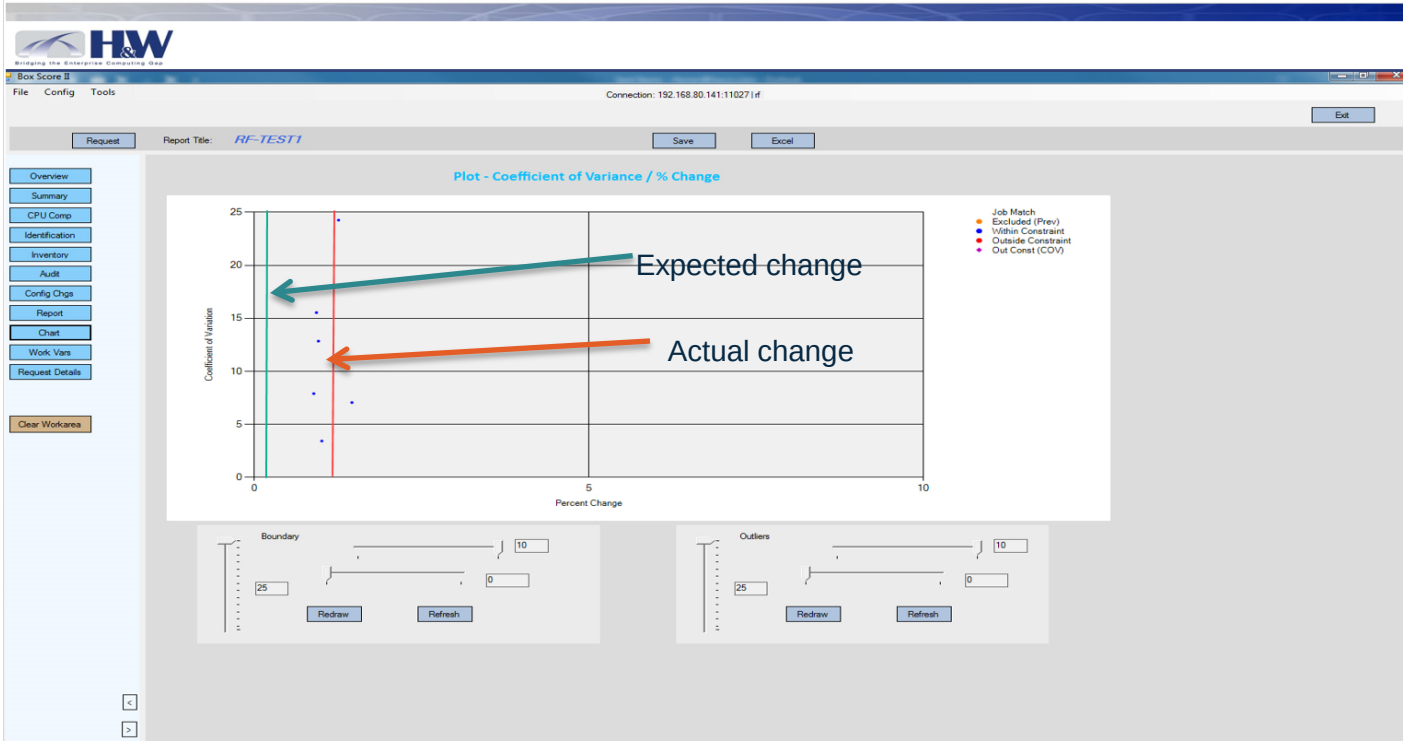


Inventory of Job / Step / Program sets

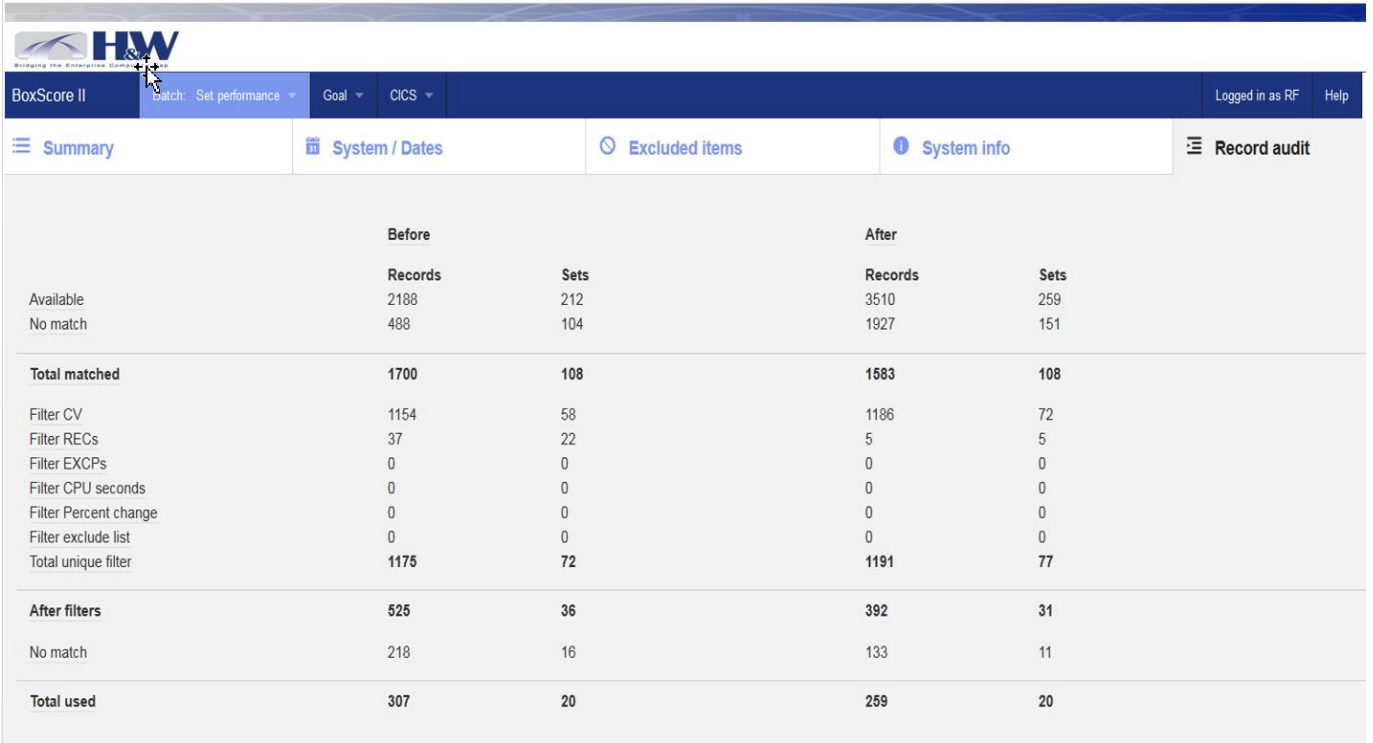
Display: ☑ Included in analysis ☐ (B) Filtered out by Before parameters ☐ (A) Filtered out by After parameters

CPU Ratio	Weight	Job	Step	Program	RECs base	RECs obs	CPU base	CPU obs	CV base	CV obs	CPU/IO base	CPU/IO obs	Filtered
1.0414	2	BXIIBKUP	DUMP	ADRDSSU	5	3	432.040	276.153	4.080	3.910	1.150	1.200	
1.0095	0.5	BXIIBKUP	FTPPUT	FTP	5	3	111.787	69.283	3.500	3.520	0.520	0.520	
0.8225	0	KMBXBLD	COPYTRS	IEBGENER	7	10	2.791	5.603	11.110	10.860	0.400	0.320	
0.9641	0.4	KMBXBLD	TRSFINAL	AMATERSE	7	10	28.557	66.851	11.600	8.150	0.970	0.940	
0.9824	0	KMBXBLD	TRS01	AMATERSE	7	10	5.644	7.908	9.070	9.430	0.790	0.780	
0.8953	0	KMBXBLD	TRS03	AMATERSE	7	10	2.768	3.541	7.310	9.980	2.970	2.660	

After the fact benchmarks



After the fact benchmarks



	Before		After	
	Records	**Sets**	**Records**	**Sets**
Available	2188	212	3510	259
No match	488	104	1927	151
Total matched	**1700**	**108**	**1583**	**108**
Filter CV	1154	58	1186	72
Filter RECs	37	22	5	5
Filter EXCPs	0	0	0	0
Filter CPU seconds	0	0	0	0
Filter Percent change	0	0	0	0
Filter exclude list	0	0	0	0
Total unique filter	**1175**	**72**	**1191**	**77**
After filters	**525**	**36**	**392**	**31**
No match	218	16	133	11
Total used	**307**	**20**	**259**	**20**

Why benchmark?

- Confirm the speed and capacity of a new machine
  - Most installations purchase/lease a new machine every few years
  - If the speed is slower or capacity lower than expected, a later upgrade could be very expensive, or cost of software could go up
  - If the speed is faster or capacity higher, users might end up paying more

Why benchmark?

- Determine the impact on chargeback
  - Extremely important to outsourcers and their customers
  - If the speed is slower or capacity lower than expected, customers bills can go up for the same amount of work
  - If the speed is faster or capacity higher, outsourcers can see reduced revenue
  - Finding the ‘sweet spot’ is important

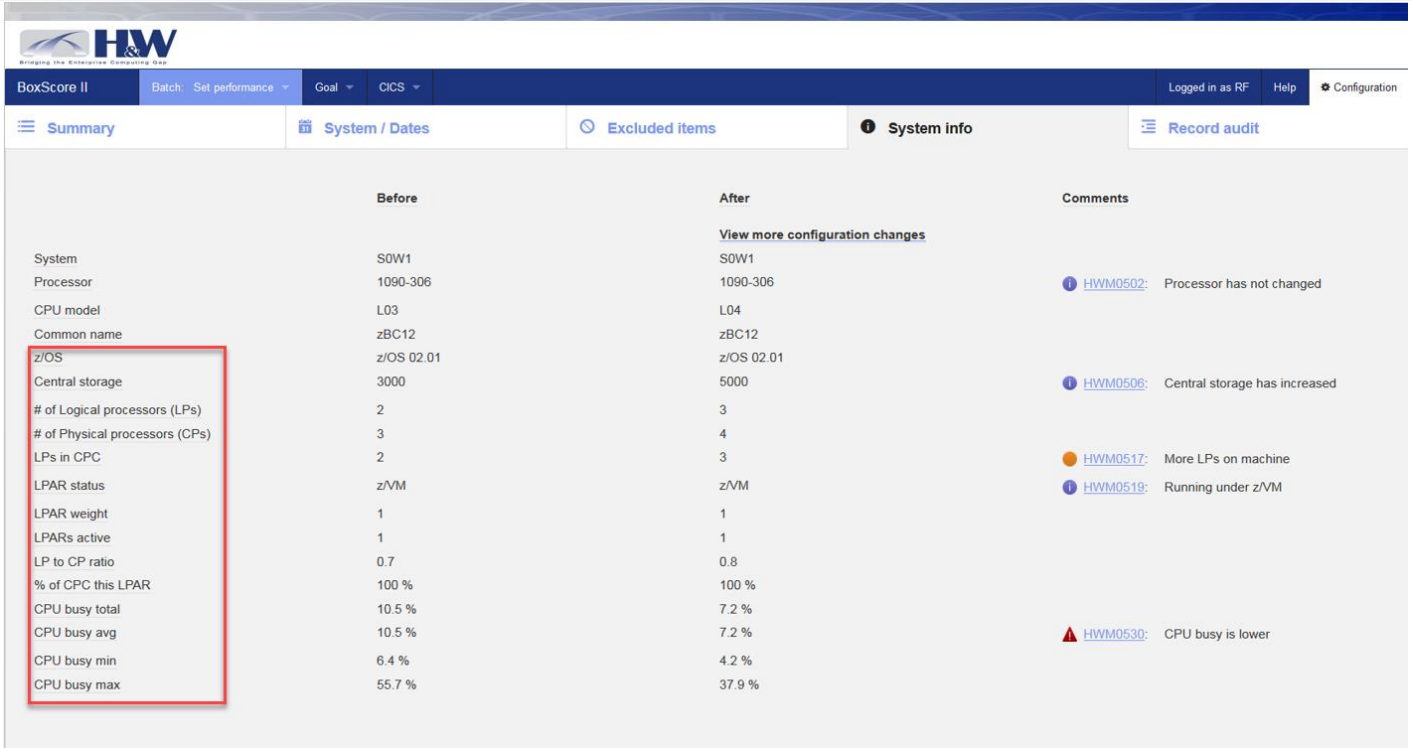
Why benchmark?

- Estimate capacity for new application
  - If new application's expected usage is validated prior to going into production, benchmarking can provide an estimate for the new capacity
  - Or it can verify that estimated capacity is correct

Why benchmark?

- See the effect of software or hardware change
  - Examples:
    - Tuning an application can change CPU time
    - Adding an LPAR can increase CPU
    - Adding LPs to an LPAR can increase CPU
    - Turning on HiperDispatch can decrease CPU
    - Adding a virtual tape controller can decrease CPU
    - Running at a higher CPU utilization can increase CPU

Why benchmark?



H&W

BoxScore II | Batch: Set performance | Goal | CICS | Logged in as RF | Help | Configuration

Summary | System / Dates | Excluded items | System info | Record audit

	Before	After	Comments
		View more configuration changes	
System	S0W1	S0W1	
Processor	1090-306	1090-306	HWM0502: Processor has not changed
CPU model	L03	L04	
Common name	zBC12	zBC12	
z/OS	z/OS 02.01	z/OS 02.01	
Central storage	3000	5000	HWM0506: Central storage has increased
# of Logical processors (LPs)	2	3	
# of Physical processors (CPs)	3	4	
LPs in CPC	2	3	HWM0517: More LPs on machine
LPAR status	z/VM	z/VM	HWM0519: Running under z/VM
LPAR weight	1	1	
LPARs active	1	1	
LP to CP ratio	0.7	0.8	
% of CPC this LPAR	100 %	100 %	
CPU busy total	10.5 %	7.2 %	
CPU busy avg	10.5 %	7.2 %	HWM0530: CPU busy is lower
CPU busy min	6.4 %	4.2 %	
CPU busy max	55.7 %	37.9 %	

Why benchmark?

- See the effect of software or hardware change
  - More examples:
    - Upgrading to a new z/OS release can change CPU
    - Upgrading to a new compiler (e.g. COBOL V5) can change CPU
    - Upgrading to a new middleware release (e.g. CICS or DB2) can change CPU
    - Applying standard RSU maintenance can change CPU
    - Change of relative nest intensity (RNI) can change CPU

How does RNI affect benchmarks?

- RNI (Relative Nest Intensity)
  - Indicator of how memory is accessed when not found in on-core cache (L1/L2 cache)
  - IBM CPU Measurement Facility (CPU MF) is microcode on machines starting with z10
  - Hardware Instrumentation Services (HIS) retrieves data from CPU MF and writes to SMF 113 records
  - Look for upcoming Tuning Letter article on 113 records and RNI
  - Also check out session 19488 from Todd Havekost on his z13 experiences

How does RNI affect benchmarks?

- RNI (Relative Nest Intensity)
  - RNI for z13/z13s = 2.6 * ((0.4 * L3P) + (1.6 * L4LP) + (3.5 * L4RP) + (7.5 * MEMP)) / 100
    - L3P = % of L1 misses found in L3
    - L4LP = % of L1 misses found in local L4
    - L4RP = % of L1 misses found in remote L4
    - MEMP = % of on-core misses found in real memory

How does RNI affect benchmarks?

- RNI (Relative Nest Intensity)
  - Workload category is based on RNI and L1MP (% of misses from L1)

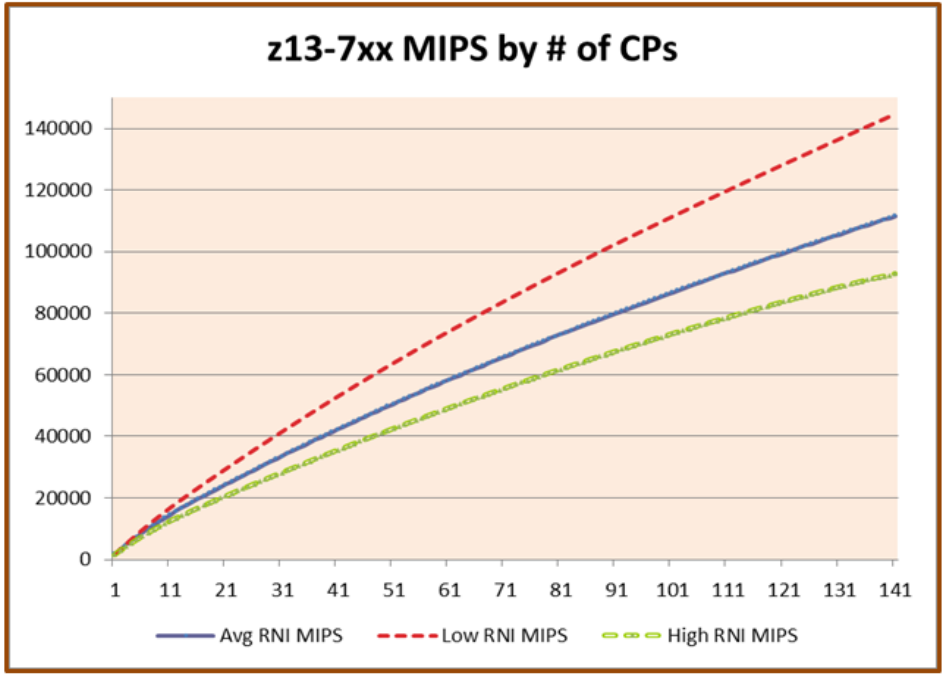
L1MP	RNI	Workload Category
< 3%	>= 0.75	Average
	< 0.75	Low
3% to 6%	> 1.0	High
	0.6 to 1.0	Average
	< 0.6	Low
> 6%	>= 0.75	High
	< 0.75	Average

How does RNI affect benchmarks?

- Work that tends to be high RNI:
  - Has a high I/O rate
  - Has little CPU usage
  - Has many different applications
  - Has little application tuning
  - Has lots of LPARs
  - Consists of transactional types of work (like CICS or WAS)
- Work that tends to be low RNI:
  - Has a low I/O rate
  - Has heavy CPU users
  - Has a single type of application
  - Has a lot of application tuning
  - Has few competing LPARs
  - Exhibits batch characteristics

How does RNI affect benchmarks?

- RNI (Relative Nest Intensity)



How does RNI affect benchmarks?

- RNI Example
  - zEC12-708, 8-way, 10063 MIPS avg RNI, 11077 MIPS low RNI, 8974 high RNI
  - z13-708, 8-way, 11188 MIPS avg RNI, 12724 MIPS low RNI, 9618 MIPS high RNI
  - Extract from *Cheryl Watson's February 2016 CPU Chart*:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Type-Model	# of CPs	IBM PCI	Avg RNI MIPS	Avg RNI MIPS/ CP	MP	Low RNI MIPS	Low-Avg RNI MIPS	Avg-Hi RNI MIPS	High RNI MIPS	SU/Sec	UP SU/Sec	Common Name	Proc Grp	S/W MSUs	MIPS/ S/W MSU	S/W MSUs/ CP	H/W MSUs	MIPS/ H/W MSU	% diff in MSUs
254	2964-707	7	9964	9964.3	1423.5	0.84	11260.2	10572.7	9224.3	8586.6	73059.3607	87431.6940	z13	IMLC	1212	8.2	173.1	1841	5.4	52%
255	2964-708	8	11188	11187.5	1398.4	0.82	12723.9	11906.4	10343.6	9618.1	71748.8789	87431.6940	z13	IMLC	1356	8.3	169.5	2066	5.4	52%

How does RNI affect benchmarks?

- RNI Example (cont.)
  - If RNI doesn't change during machine upgrade, then avg RNI would expect 11.2% increase; low RNI: 14.9% increase; high RNI: 7.0% increase
  - If RNI changes from high to low, expect 41.8% increase; if RNI changes from low to high, expect 13.2% decrease;
  - That's over a 50% difference in expectations!
  - You don't know what new RNI will be; so benchmarking is a MUST!

Tips for reducing RNI

- Focus on peak R4HA because that's where it matters most
- Provide more memory
  - Remember that 3x and 5x current memory can provide significant discounts in cost
  - Allows PR/SM to move memory between drawers and allows LPAR to remain on a single drawer to reduce L4RP
- Tune applications to reduce I/O interrupts (large blocksizes, larger buffers, data in memory (like sorts))

Tips for reducing RNI

- Reduce CPU busy
  - Each 10% CPU busy changes CPU usage by workloads by 3% (low RNI) to 5% (high RNI)
- Maximize vertical high LPs in each LPAR
  - See John Burg's session 18912 from Monday SHARE for more information
- Exploit large memory use in DB2 V11 and others
- Reduce I/O response times

Why benchmark every week?

- The MIPS (i.e. speed and capacity) of a machine or an LPAR will change over time; you need to know why
- It's hard to identify whether change is due to volume or changes on the system – benchmark can help determine
- The RNI of the LPAR might change
- Chargeback will certainly change over time
- As will your software bills

Why benchmark every week?

- Maintenance APARs might change MIPS
- A PARMLIB change or WLM policy change can change MIPS
- A change in the size of a database can change MIPS
- Changes might be made that nobody knows about
- It’s easy to do with the right tool!

Special benchmarks on demand

- There are also reasons to benchmark special situations
  - Application made some tuning changes, and wants to see the effect
  - Sysprog changed the weight of an LPAR, and wants to see the effect
  - New DDF transaction just introduced, what is impact on the rest of the workloads?
  - Performance team changed sort options, and want to see the impact (on sort steps and other steps)
  - Application finished COBOL conversion, wants to see impact

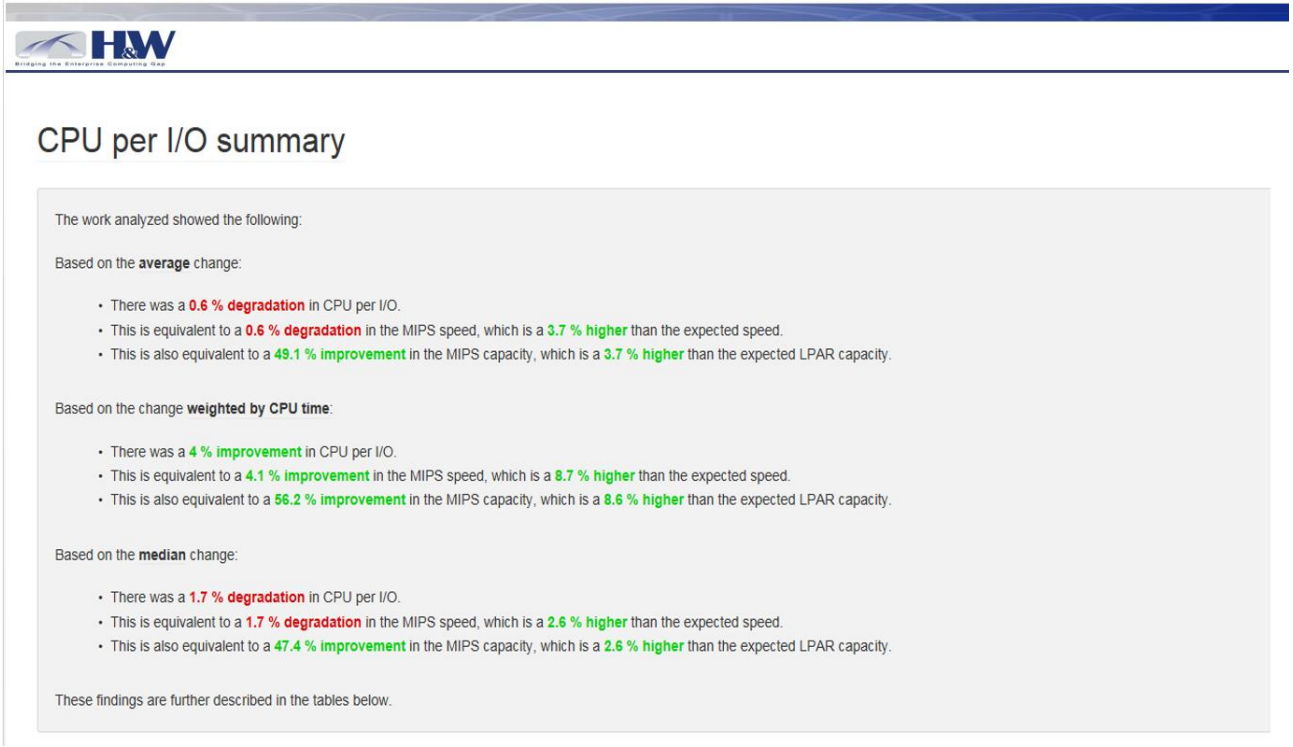
BoxScore II

- In May 2015, Watson & Walker partnered with H&W Computer Systems to develop a new version of BoxScore
- Design provided by Cheryl Watson; development, support, and marketing provided by H&W
- Product web page went up today –
  - http://www.hwcs.com/software/boxscore-ii
  - Contact H&W for additional information

BoxScore II enhancements

- Browser-based interactive/drill-down GUI with **no SAS dependence**
- Data is collected as SMF is written and stored in DB2 database; so is available at all times (no SMF post-processing needed!)
- Extensive help in understanding data; extensive messages to alert you of potential problems
- Can easily exclude outliers and rerun in seconds
- Provides average, weighted, median values
- Provides normalization factor for chargeback
- Data available in csv format

BoxScore II sample screens



H&W
Bridging the Enterprise Computing Gap

CPU per I/O summary

The work analyzed showed the following:

Based on the **average** change:

- There was a **0.6 % degradation** in CPU per I/O.
- This is equivalent to a **0.6 % degradation** in the MIPS speed, which is a **3.7 % higher** than the expected speed.
- This is also equivalent to a **49.1 % improvement** in the MIPS capacity, which is a **3.7 % higher** than the expected LPAR capacity.

Based on the change **weighted by CPU time**:

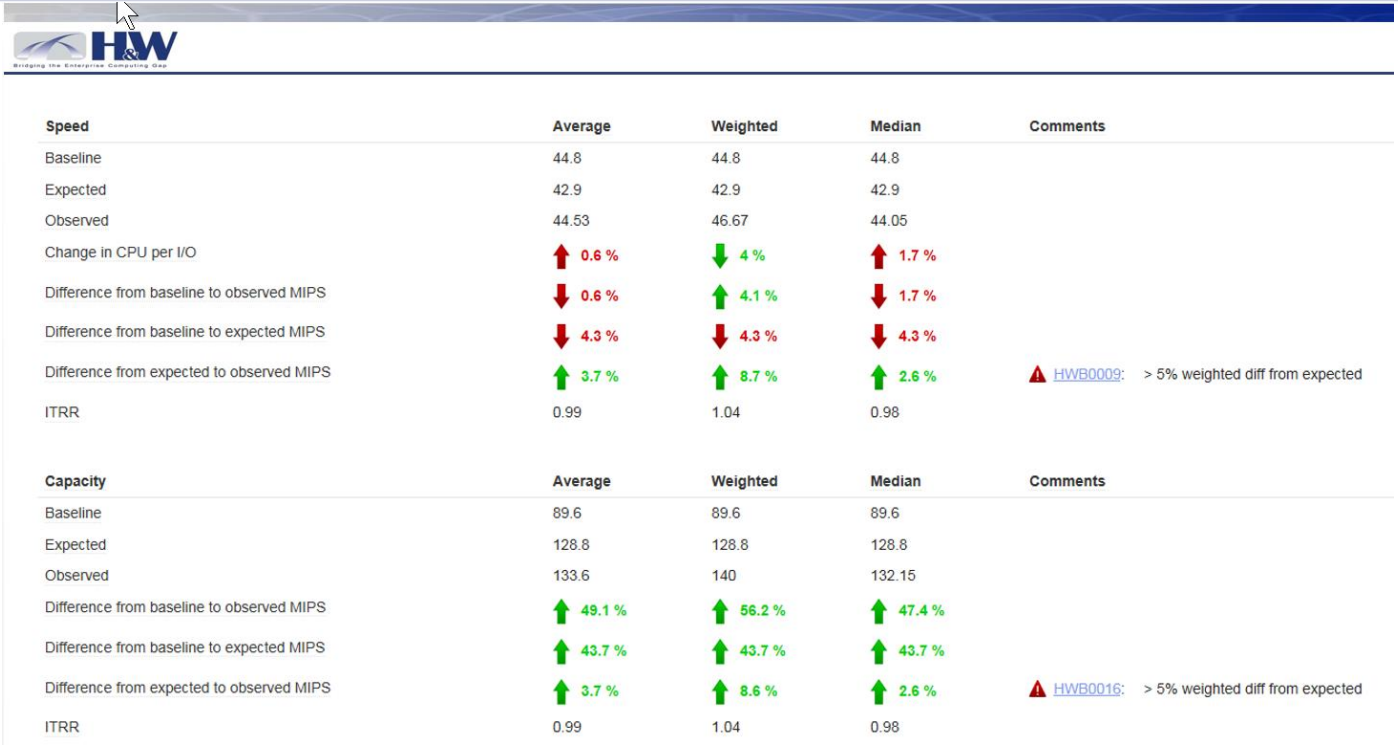
- There was a **4 % improvement** in CPU per I/O.
- This is equivalent to a **4.1 % improvement** in the MIPS speed, which is a **8.7 % higher** than the expected speed.
- This is also equivalent to a **56.2 % improvement** in the MIPS capacity, which is a **8.6 % higher** than the expected LPAR capacity.

Based on the **median** change:

- There was a **1.7 % degradation** in CPU per I/O.
- This is equivalent to a **1.7 % degradation** in the MIPS speed, which is a **2.6 % higher** than the expected speed.
- This is also equivalent to a **47.4 % improvement** in the MIPS capacity, which is a **2.6 % higher** than the expected LPAR capacity.

These findings are further described in the tables below.

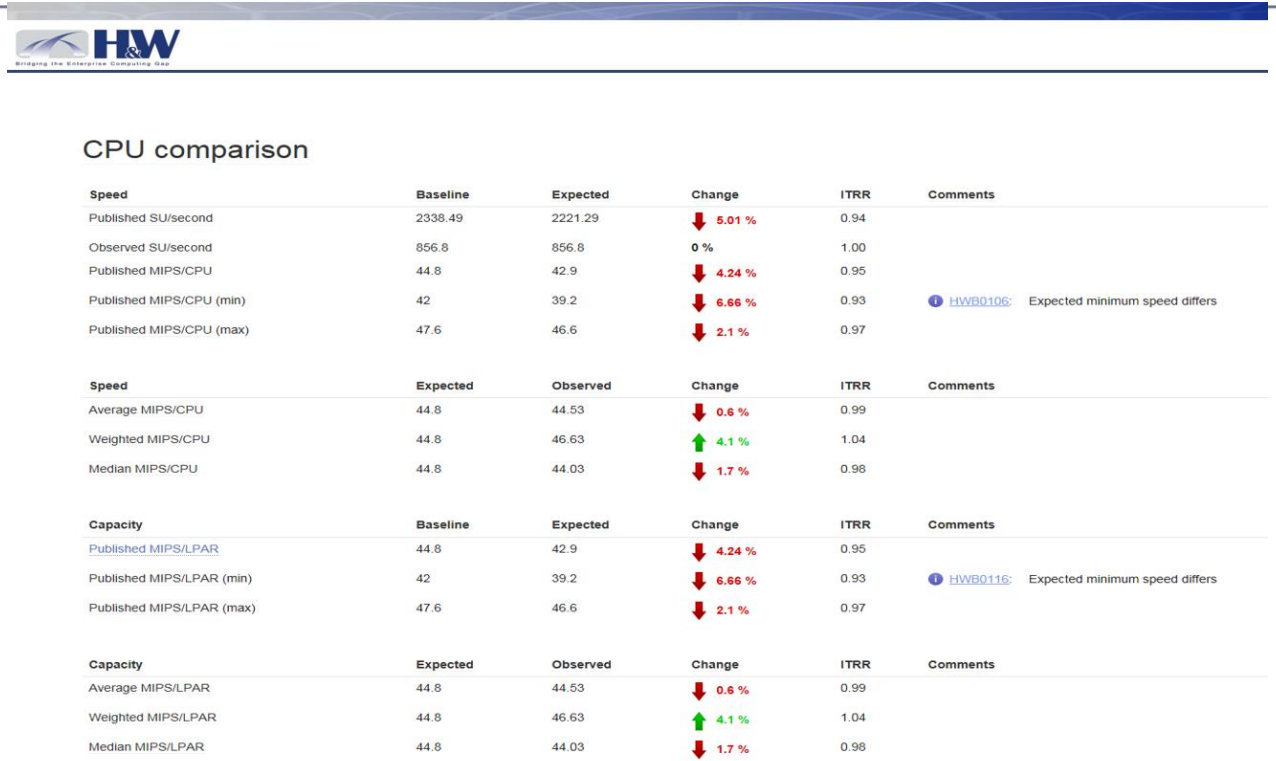
BoxScore II sample screens



Speed	Average	Weighted	Median	Comments
Baseline	44.8	44.8	44.8	
Expected	42.9	42.9	42.9	
Observed	44.53	46.67	44.05	
Change in CPU per I/O	↑ 0.6 %	↓ 4 %	↑ 1.7 %	
Difference from baseline to observed MIPS	↓ 0.6 %	↑ 4.1 %	↓ 1.7 %	
Difference from baseline to expected MIPS	↓ 4.3 %	↓ 4.3 %	↓ 4.3 %	
Difference from expected to observed MIPS	↑ 3.7 %	↑ 8.7 %	↑ 2.6 %	⚠ HWB0009: > 5% weighted diff from expected
ITRR	0.99	1.04	0.98	

Capacity	Average	Weighted	Median	Comments
Baseline	89.6	89.6	89.6	
Expected	128.8	128.8	128.8	
Observed	133.6	140	132.15	
Difference from baseline to observed MIPS	↑ 49.1 %	↑ 56.2 %	↑ 47.4 %	
Difference from baseline to expected MIPS	↑ 43.7 %	↑ 43.7 %	↑ 43.7 %	
Difference from expected to observed MIPS	↑ 3.7 %	↑ 8.6 %	↑ 2.6 %	⚠ HWB0016: > 5% weighted diff from expected
ITRR	0.99	1.04	0.98	

BoxScore II sample screens



CPU comparison

Speed	Baseline	Expected	Change	ITRR	Comments
Published SU/second	2338.49	2221.29	↓ 5.01 %	0.94	
Observed SU/second	856.8	856.8	0 %	1.00	
Published MIPS/CPU	44.8	42.9	↓ 4.24 %	0.95	
Published MIPS/CPU (min)	42	39.2	↓ 6.66 %	0.93	HWB0106: Expected minimum speed differs
Published MIPS/CPU (max)	47.6	46.6	↓ 2.1 %	0.97	

Speed	Expected	Observed	Change	ITRR	Comments
Average MIPS/CPU	44.8	44.53	↓ 0.6 %	0.99	
Weighted MIPS/CPU	44.8	46.63	↑ 4.1 %	1.04	
Median MIPS/CPU	44.8	44.03	↓ 1.7 %	0.98	

Capacity	Baseline	Expected	Change	ITRR	Comments
Published MIPS/LPAR	44.8	42.9	↓ 4.24 %	0.95	
Published MIPS/LPAR (min)	42	39.2	↓ 6.66 %	0.93	HWB0116: Expected minimum speed differs
Published MIPS/LPAR (max)	47.6	46.6	↓ 2.1 %	0.97	

Capacity	Expected	Observed	Change	ITRR	Comments
Average MIPS/LPAR	44.8	44.53	↓ 0.6 %	0.99	
Weighted MIPS/LPAR	44.8	46.63	↑ 4.1 %	1.04	
Median MIPS/LPAR	44.8	44.03	↓ 1.7 %	0.98	

BoxScore II

- In summary - Boxscore II is the new version of BoxScore, the only commercially available after-the-fact benchmark
- For a demo by Cheryl, please stop by the H&W booth tonight or tomorrow
- If you'd like more information, contact H&W or go to their website at http://www.hwcs.com/software/boxscore-ii
- For more news and updates, subscribe to our Cheryl's List blog at www.watsonwalker.com/blog

Visit the SHARE Booth (#303)

SHARE
EDUCATE • NETWORK • INFLUENCE

Educate

- Learn more about **hot topics** from peers in the **Tech Talk Corner**
- Take part in the **Hack-a-Thon** with IBM & Rocket

Network

- **Engage** with fellow attendees
- Connect with the **zNextGen®** community
- Join the **#SHAREatl** social media conversation

Influence

- Discover ways to **get involved** with SHARE
- Meet **SHARE Volunteers**

Complete your session evaluations online at SHARE.org/Evaluation

Thank You for Attending!
Please remember to complete your evaluation of this session in the SHARE mobile app.

Benchmark every week? Why?
Are you crazy?

Complete your session evaluations online at SHARE.org/Evaluation