SHARE San Jose 2017

Mainframe Performance and You

Frank Kyne, Watson & Walker

Cheryl Watson, Watson & Walker

www.watsonwalker.com

technical@watsonwalker.com

Watson Walker

Welcome

- Thank you for attending this session – we hope to give you lots of useful tips to bring home.
- Who are we?
  - Watson & Walker Inc established 1986; Cheryl has been working on IBM mainframes since 1965; Frank joined Watson & Walker in 2014.
  - Publish Cheryl Watson's Tuning Letter (since 1991).
  - Trainers, consultants, IBM Business Partner, software vendor (see us in Technology Exchange).
  - z/OS evangelists, Subject Matter Experts in Parallel Sysplex, Workload Manager, and software pricing.

- If you have questions, please ask as we go along.

What *Is* Performance?

Acceleration?

Top Speed?

MPG?

Yes! Can I have them *all*?

AND for the price of a Nissan Versa?

Lucky you!

Why Should We Care About Performance?

- We *all* work for IT companies. Some of our companies also provide banking services, manufacture airplanes, lease motor cars, or sell clothes. But ALL of our customers interact directly with our z/OS systems – to many customers, we *are* our systems. If they are not available or responsive, we lose a customer.
- Performance is one aspect of Availability.
- Poor response times are no longer a number on a report, they are lost productivity and lost customers.
- One way to address poor performance is to add more capacity – this costs $s. Or, you could improve performance and free up capacity, delaying the need to upgrade.

- Costs. For many reasons, modern processors are more sensitive to basic workload performance characteristics.
  - We are going to focus on the intersection between cost and performance in this session.

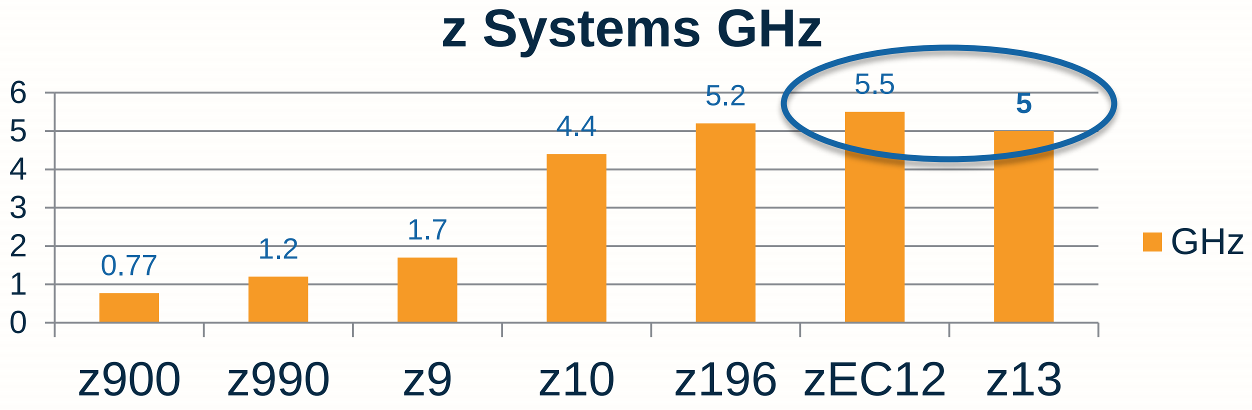
Complete your session evaluations online at SHARE.org/Evaluation

It Doesn’t Happen Automatically

- Many of the performance-enhancing functions described in this presentation are included in the base hardware or software.
- **BUT**, the majority of them *do not just happen automatically*. In order to protect compatibility, most new IBM functions are shipped in the ‘turned off’ mode, and need to be explicitly enabled.
- How do your staff know about these new functions and which are appropriate to your environment? *By attending conferences like this one.*
- This becomes *even more important* as IBM accelerates its move to a continuous delivery strategy.

Moore’s Law (RIP)

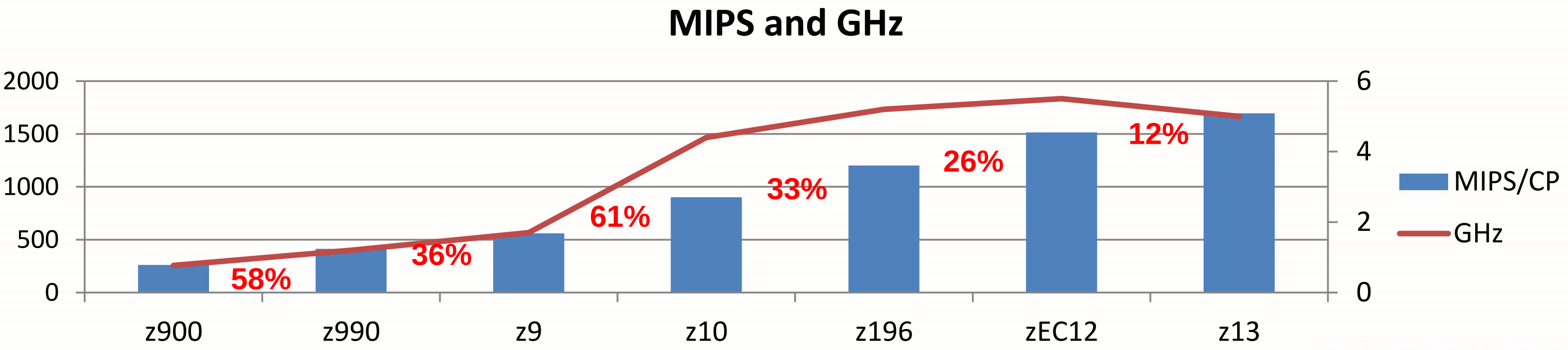
- In 1965, Intel co-founder Gordon Moore observed “the number of transistors per square inch doubles approximately every 12 months.” It was subsequently modified to be every 18 months rather than every 12. However, this is an exponential equation, so it cannot carry on indefinitely. In 2017, if we are not at the end of that period, we are very close to the end.



Complete your session evaluations online at SHARE.org/Evaluation

MIPS Growth

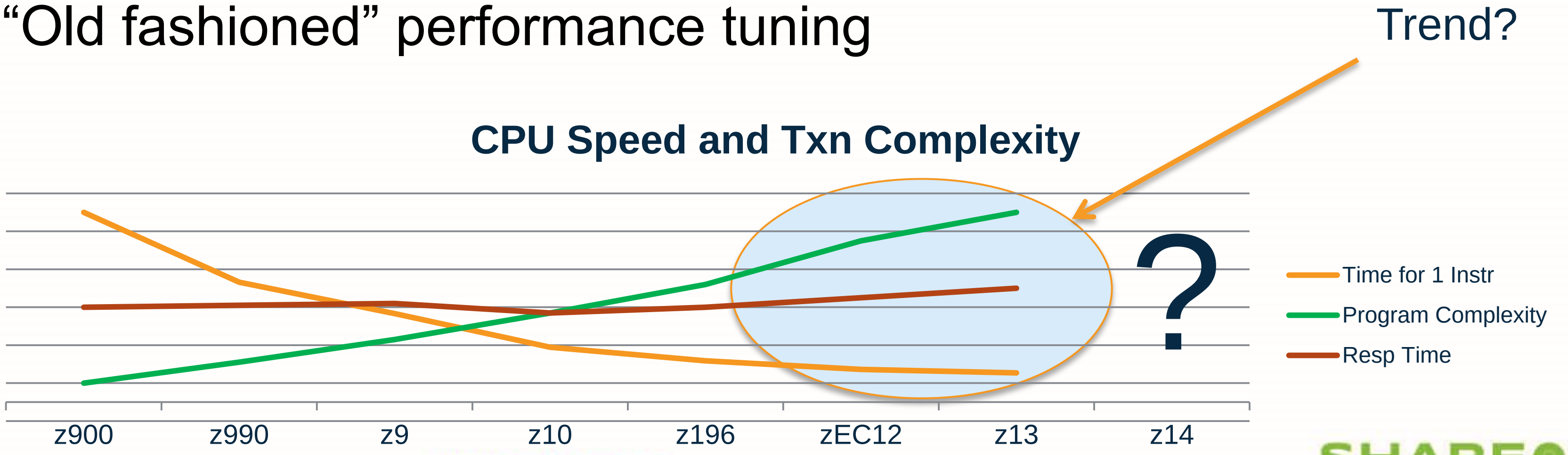
- For a long time, increases in processor speed (MIPS) were closely tied to increasing chip speeds. Starting with z10, we see a divergence between chip speed changes and processor speed changes. In particular, on z13, chip speed declined, but CPU speed was still higher than the previous generation.



Complete your session evaluations online at SHARE.org/Evaluation

Workload and Response Time Trends

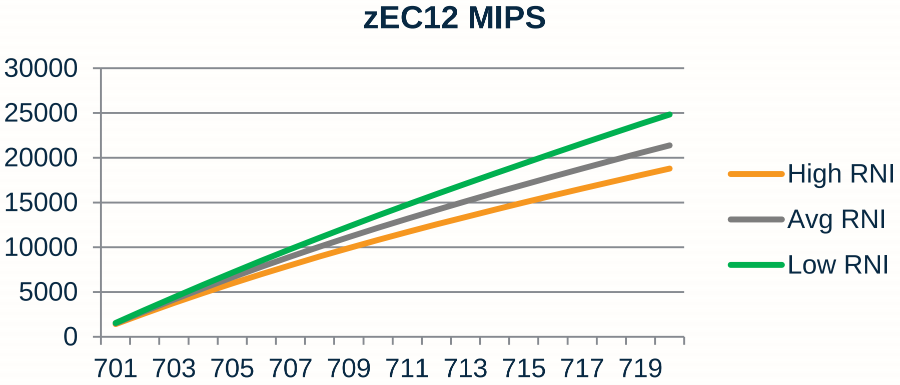
- Over the years, increased transaction complexity and function has been offset by increases in chip and processor speed, resulting in stable response times. But what will happen when complexity continues to increase (more security, in-transaction analytics, etc), but chip and processor speeds level off?
- A: “Old fashioned” performance tuning

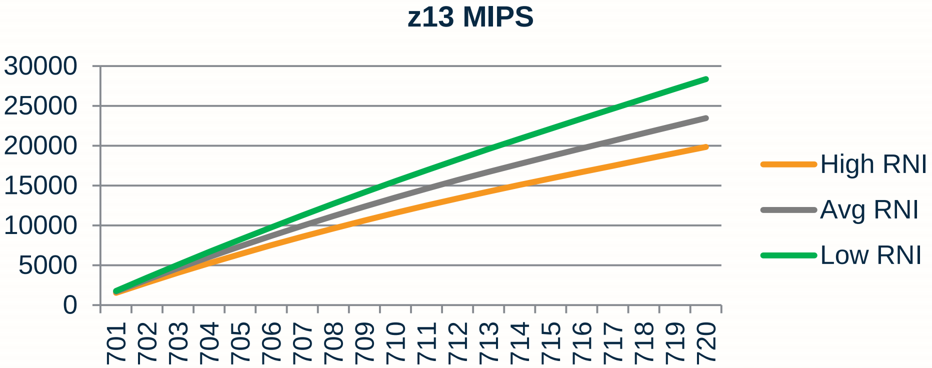


Complete your session evaluations online at SHARE.org/Evaluation

Relationship Between Cost and Performance

- Relative Nest Intensity (RNI) is an indicator of the memory and processor cache usage pattern of the workload.
  - On a 20-way zEC12, a ‘Low RNI’ workload runs 32% faster than a High RNI workload.
  - On a 20-way z13, the difference is 43%. On a top-end z13, the difference is 56%.
- Software charges are based on the capacity delivered for an Average RNI workload. If your workload has a Low RNI on a z13, you might get an extra 21% for ‘free’. If your workload has a High RNI, you might see 18% less capacity, meaning a higher software bill to do the same amount of work as someone with an Average RNI workload.

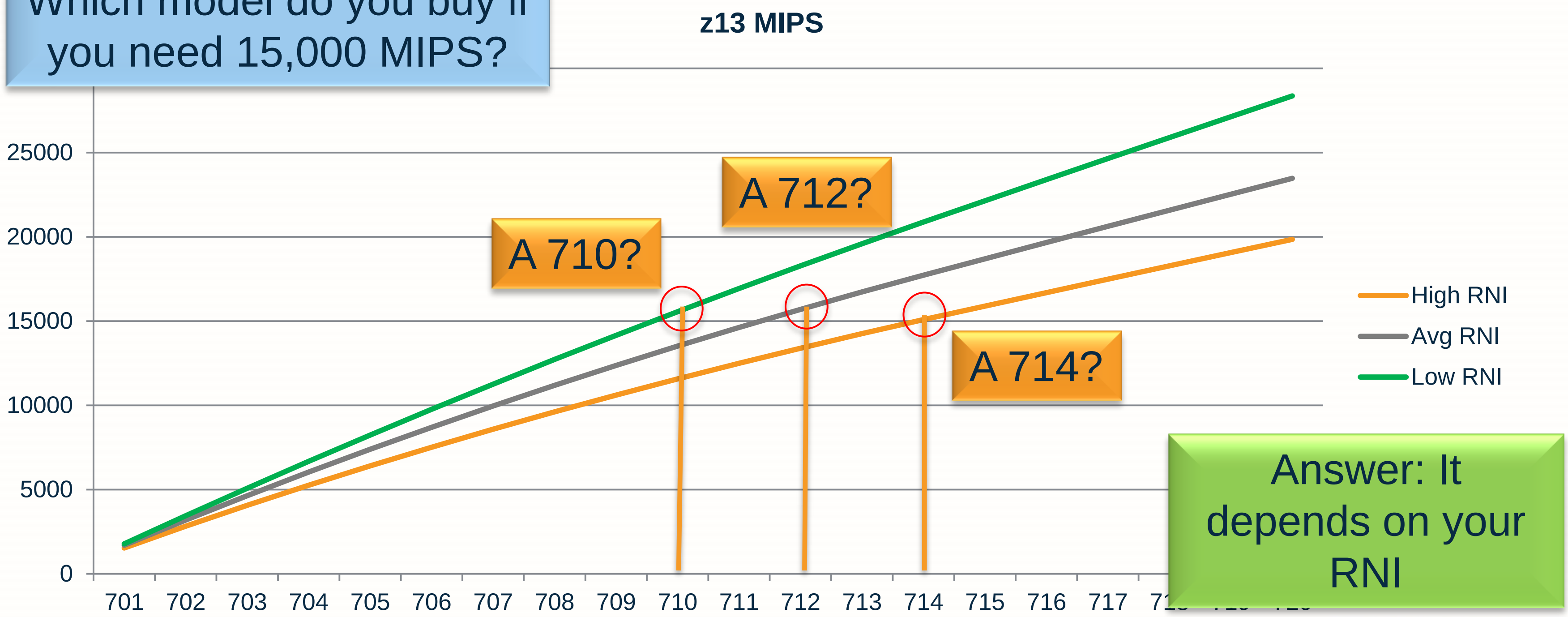




Complete your session evaluations online at SHARE.org/Evaluation

Size Matters (To Your Wallet)

Which model do you buy if you need 15,000 MIPS?



A 710?

A 712?

A 714?

Answer: It depends on your RNI

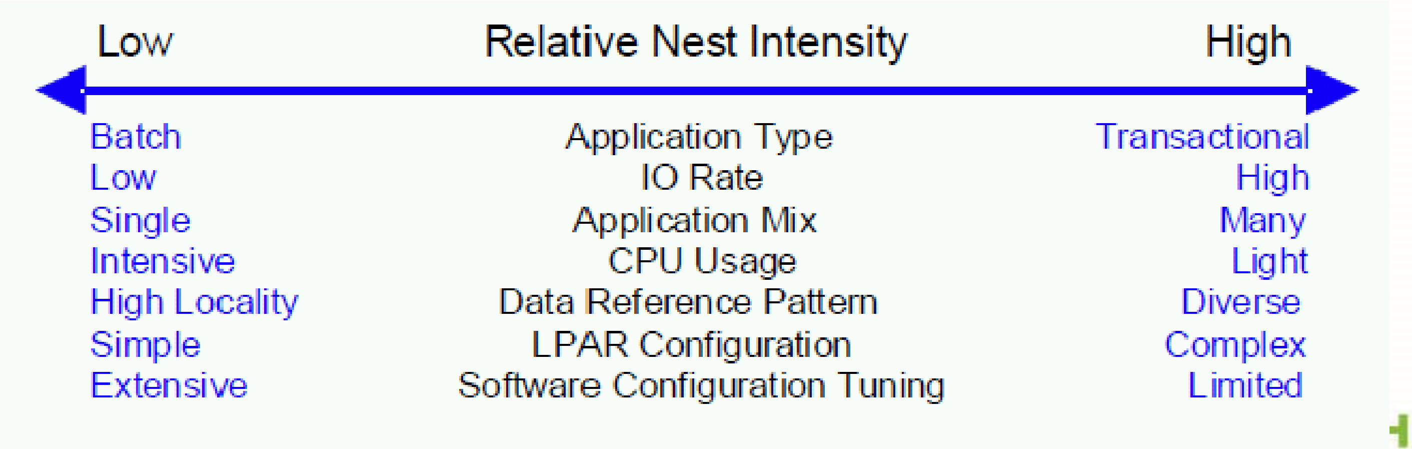
What is YOUR RNI?

- How do you know the RNI of *your* workload? And can you influence it?
- Information about usage of the processor cache and memory is gathered by the CPU Measurement Facility and saved in SMF type 113 records. There are a variety of excellent products to process that data and report your RNI.
  - NEVER purchase a CPU upgrade without consulting IBM's z Systems Processor Capacity Reference (zPCR) tool.
  - And never do a capacity planning exercise without including the SMF type 113 records.
- And yes, there are *many* things that you can do to influence the RNI of your workload.

Influencing Your Relative Nest Intensity

- A workload's performance is sensitive to how deep into the memory hierarchy the processor must go to retrieve the workload's instructions and data for execution. The best performance (in other words, the max MIPS) occurs when the instructions and data are found in the cache(s) nearest the processor. The RNI indicates the level of activity to shared caches and memory resources (L3, L4, memory). The higher the RNI, the deeper into the memory hierarchy the processor must go to retrieve the instructions and data for that workload.

Lower cost per txn

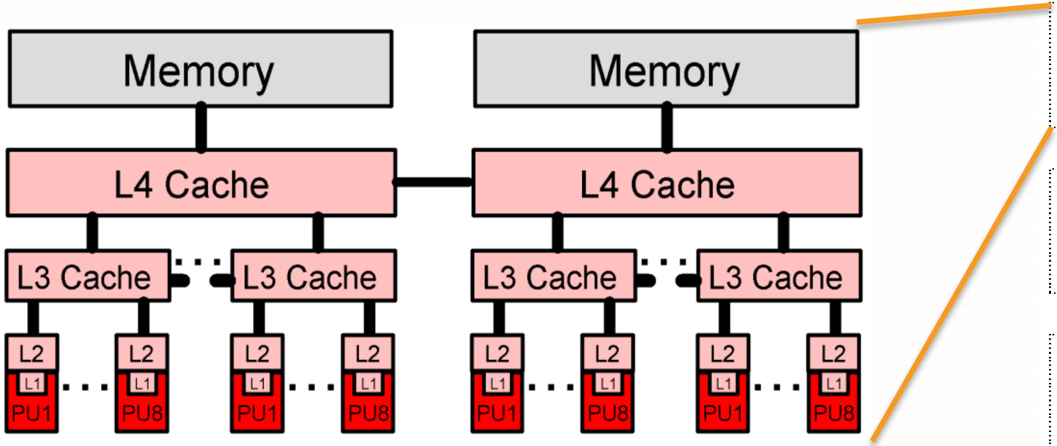


Low	Relative Nest Intensity	High
Batch	Application Type	Transactional
Low	IO Rate	High
Single	Application Mix	Many
Intensive	CPU Usage	Light
High Locality	Data Reference Pattern	Diverse
Simple	LPAR Configuration	Complex
Extensive	Software Configuration Tuning	Limited

Higher cost per txn

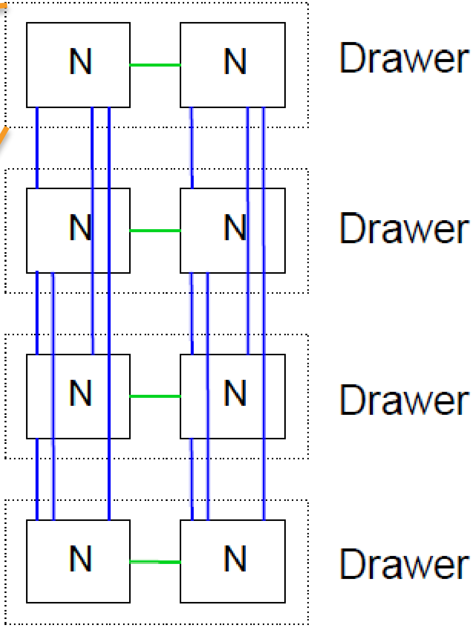
z13 Memory and Caches

z13 Single Drawer View



L1 private 96 KB I, 128 KB D	**1 Cycle**
L2 private 2 MB I + 2 MB D	**6-8 Cycles**
L3 shared 64 MB / chip	**30-500 Cycles**
L4 shared 480 MB / node	**100-400 Cycles**
Memory – up to 2.5TB/drawer	**500-1000 Cycles**

z13 4-Drawer Processor View



4 Drawer System Interconnect

Thanks For The Memory(s)

- On generations prior to z13, each LPAR's memory was striped across all books in the processor.
  - On average, this delivered uniform access times, regardless of which book the LPAR's CPs were located in.
- Access time to *local* memory is faster on a z13 than on a zEC12, but access to *remote* memory is slower on the z13.
  - To deliver the optimum efficiency, PR/SM *tries* to place all the memory for an LPAR in the same drawer as the LPAR's CPs.
  - To achieve this, PR/SM can configure installed memory on and offline in the different drawers. The online memory will not exceed the purchased amount, but the amount of online memory can vary from one drawer to another.
  - IBM always installs more memory than you pay for. The more physical memory there is installed in the CPC, the more flexibility PR/SM has to collocate memory and CPs. Refer to Table 2-5 in IBM Redbook *z13 Technical Guide*, for very helpful information about optimizing purchased and installed memory.

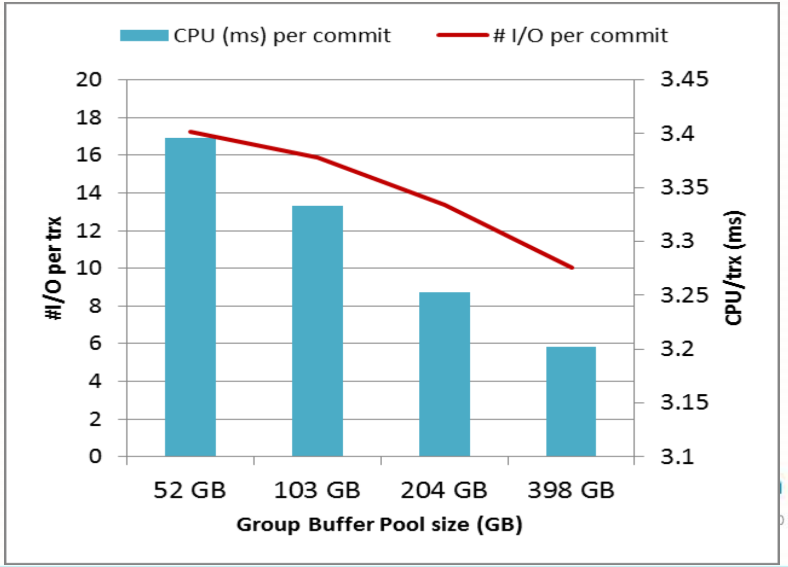
Benefits of Large Memory

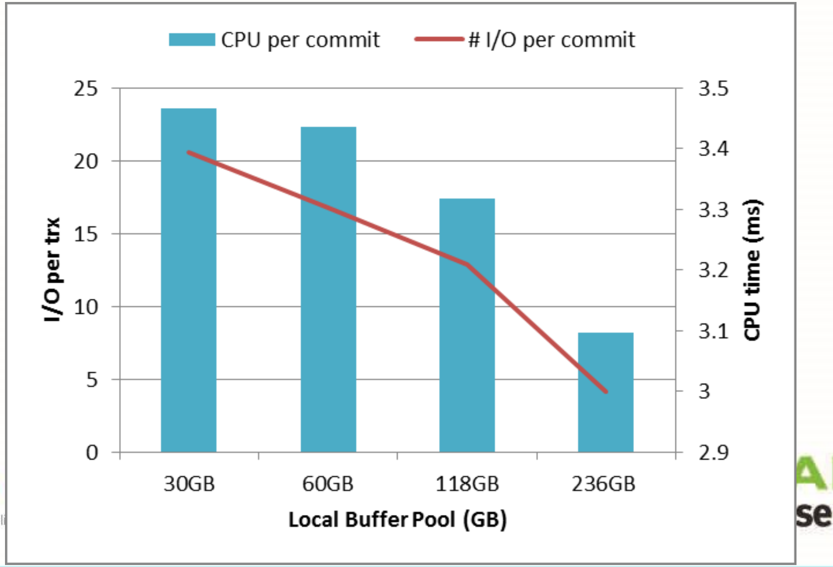
- There are many benefits to be achieved from intelligently exploiting large memory:
  - Even though memory is much slower than L1 or L2 cache, it is WAY faster than disk. So any I/O you can avoid by buffering data in memory delivers response time benefits.
  - The CPU cost of starting and completing an I/O is between 20 and 70 mics. If you can use large memory to buffer data and eliminate I/Os, that can save you CPU time as well as response time.
  - For optimum performance, you want as much of your data and instructions in L1 and L2 as possible. But if your program needs to issue an I/O, it gives up control of the cache until the I/O is complete. It is extremely unlikely that any of the program's data or instructions are still in the L1 or L2 cache at that point.
  - IBM is offering very attractive pricing if you purchase 3x your current memory when moving to z13. There is an even bigger discount if you install > 2.5TB.
  - And remember – memory is a one time cost. But your SW bill is the gift that keeps giving!

Exploiting Large Memory

Sample IBM Measurement:

- Banking (60M account) workload with 2-way sysplex data sharing :
- Increasing GBP from 52GB to 398GB with same LBP size (60GB) for both members reduced response time by 11% and CPU by 6%.
- Increasing LBP from 30GB to 236GB for both members, with same reasonable GBP size (60GB), resulted in **reducing response time by 40%** and **CPU time by 11%**.





Benefits of ‘Large Pages’

- As recently as z900, the maximum processor memory was 64 GB. On z13, the maximum memory is 10,240 GB.
- In order to support such large memory without an equivalent increase in the cost of managing that memory, z/OS 1.9 and z10 introduced support for larger pages.
  - Initially 1MB, then 2GB as well.
- There are situations where the use of large pages is ideal (buffer pools, for examples), and other situations where traditional smaller pages are more appropriate.
- Therefore, the default continues to be to use traditional, small, pages unless you specify otherwise.
- IF you spend the time to optimize, the savings can be significant.

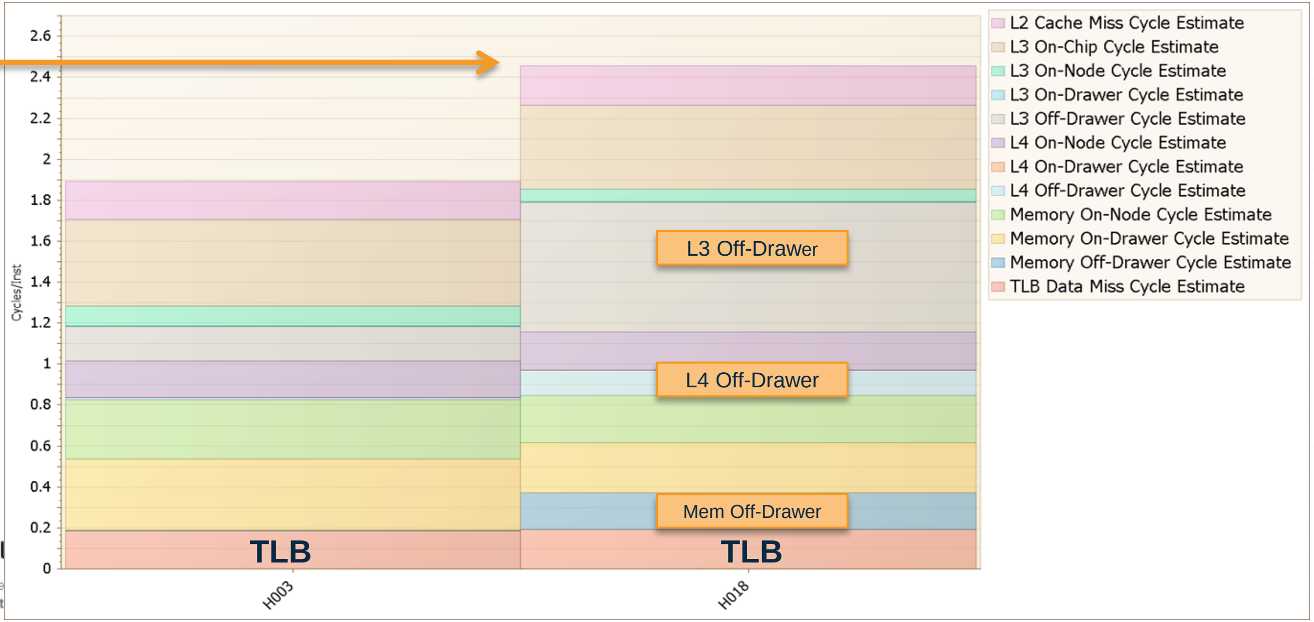
IBM Exploiters of Large Pages

- There are many exploiters of large pages and large memory on z/OS, but the most common ones are:
  - DB2. Each new version of DB2 contains enhancements to use large memory more effectively.
  - WebSphere Application Server.
  - Java.
  - CICS Transaction Gateway.

Using Memory Effectively

- This chart (from IntelliMagic Vision) illustrates the importance of collocating CPs and memory in the same drawer, and of optimizing page sizes (especially on z13).

+31%

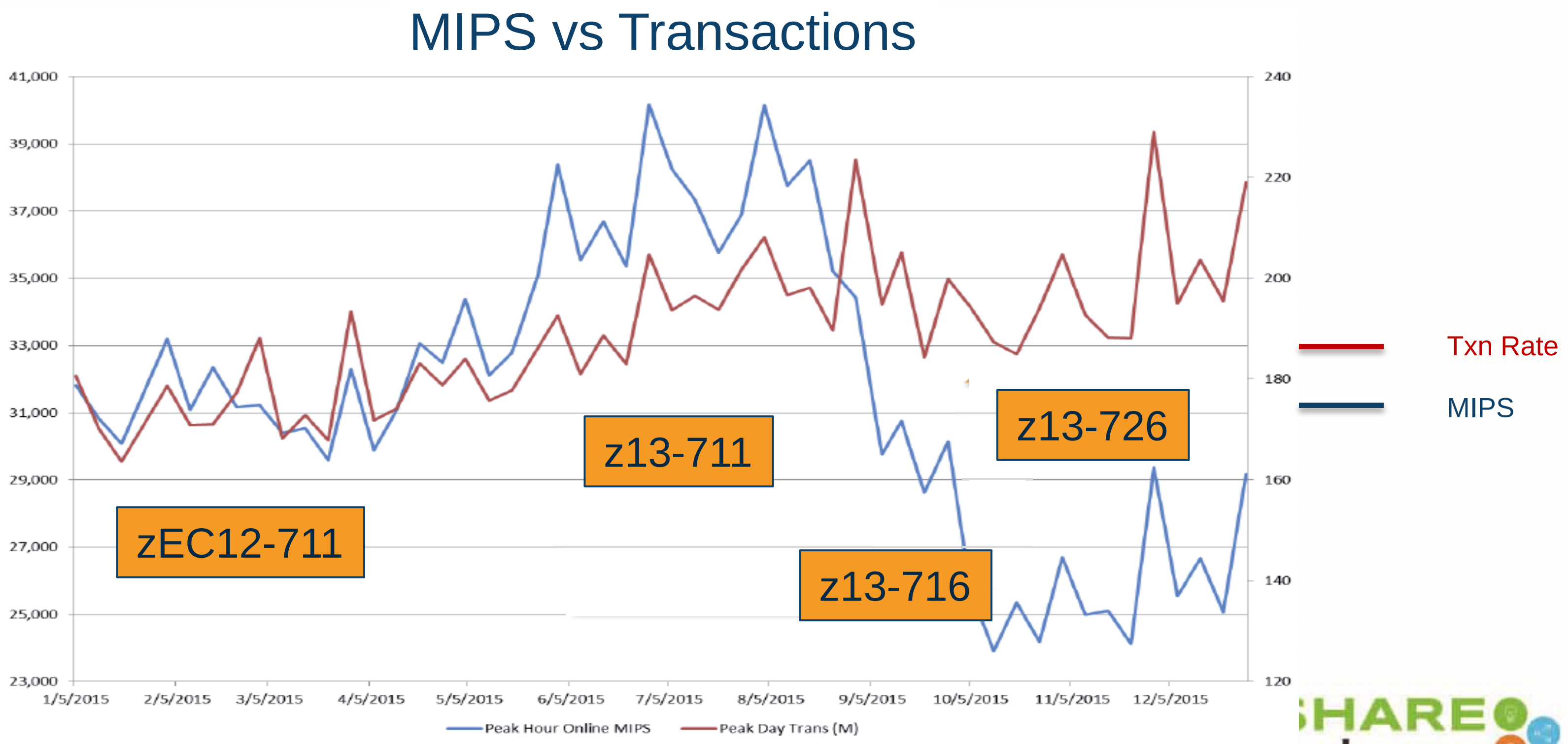


Over-provisioning MIPS

- Speaking of being generous with your hardware…
- Conventional wisdom is that you get the maximum value from your z Systems CPCs by running them as full as possible, for as long as possible.
- You still CAN do that, and meet your Service Level Agreements *as long as you have displaceable work in the system*. But the question is whether you still *want* to do that.
- The load placed on CPC caches by running systems at very high utilization decreases the efficiency and speed of the CPC:
  - IBM estimates that every 10% increase in CPU utilization increases the CPU time for a transaction by 3-5%.
- So, does running a CPU at lower utilizations really make them more efficient? Let’s look at a real customer scenario.

Customer Experience

Courtesy of Todd Havekost, IntelliMagic



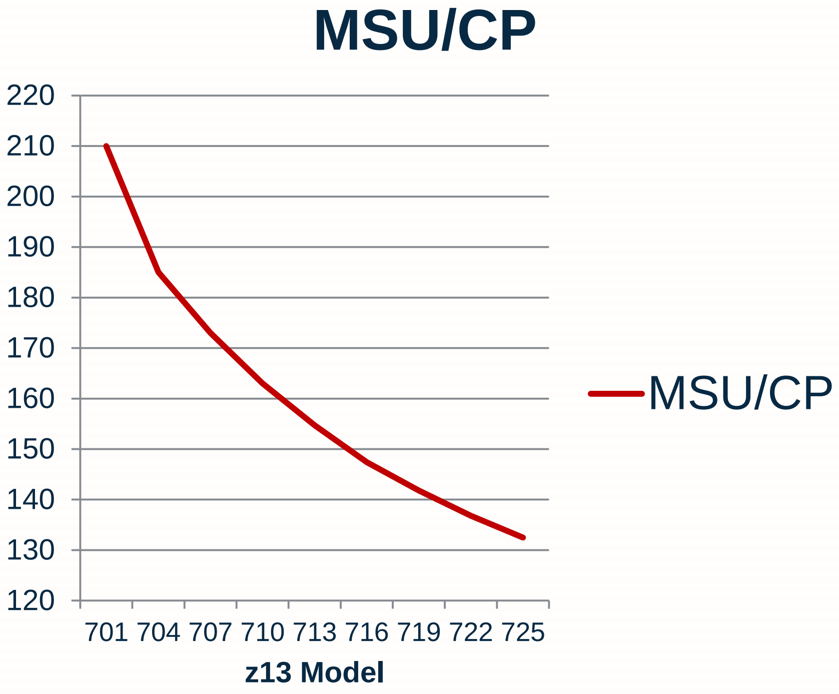
Complete yc

Over-provisioning MIPS

- This customer had an unusually High RNI, resulting in the z13 delivering only 4% more capacity than the zEC12, rather than the 10% that an Average RNI workload would expect to see:
  - Because IBM MSU ratings are based on Average RNI workloads, this resulted in them paying more for the same amount of work.
- In the previous chart, customer had ‘banked’ MIPS that they were able to turn on for no additional cost:
  - The total cost/benefit case would have to also include hardware maintenance costs, OTC software, and non-IBM software.
    - If possible, always get sub-capacity-based SW billing.
- In addition to the reduced utilization as a result of the additional CPs and caches, they also fine-tuned their LPAR configuration to lower their RNI.

Exploiting the N-Way Effect

- IBM MSU values for each model are based on IBM measurements taken at very high utilizations (95+%).
- The more work the system has to manage, the higher is the overhead – this is known as the n-way effect. It means that CPC capacity doesn't scale linearly as you add CPs – each additional CP adds a little more overhead, resulting in the average speed of each CP decreasing. This is reflected in a decreasing MSUs per CP value as you add more engines. Because IBM software prices are based on MSUs, they automatically take this effect into account.
- However, the n-way effect is partially due to the increased volume of work. What would happen if you ran a 711-worth of work on a 726? Normally, you would expect the overhead to be closer to 711 levels than 726 levels. So you get 711 level MSUs per CP, but your bills that are based on an 726 levels of MSUs per CP.



Over-provisioning MIPS

- By combining additional capacity with configuration tuning and exploiting the N-Way effect, this customer saw:
  - An INcreased consumption of 4K MIPS when they moved to the z13 711s (because of their very High RNI workloads).
  - A DEcrease in consumption of 9K MIPS when they moved from the 711s to 716s.
  - A further 4K MIPS DEcrease when they moved from 716s to 726s.

Over-provisioning MIPS

- This is not an isolated case. Another large customer reported a 25% reduction in MSUs when they enabled all CPs for a DR test.
- Every installation needs to determine its own ‘sweet spot’, bearing in mind hardware costs, software costs (both IBM and ISV) and maintenance costs.
  - We are not proposing that you should only run your z Systems CPCs at only 20 or 30% utilization, but we DO recommend that you upgrade before utilization is consistently running in the high 90%s.
- An ancillary benefit of the additional capacity was a measurable availability improvement due to the system having spare capacity to handle stress or out-of-line situations.

zIIP It

- zIIPs are your best friend if you are trying to reduce z/OS costs:
  - Their purchase price is a fraction of the price of a general purpose CP. Depending on how good a negotiator you are, you might be able to get **15** zIIPs for the price of ONE general purpose CP.
  - Also, maintenance costs of a zIIP are lower than the maintenance cost of a general purpose CP.
  - Work that runs on a zIIP is NOT included in your R4HA, meaning that it doesn't impact your software bill.
  - **zIIPs always run at the full rated speed of a CPC**. This means that if you have a z13 4xx, 5xx, or 6xx CPC, your zIIPs will be faster than your general purpose CPs – up to nearly 7x faster if you have a z13 4xx, and 21x faster if you have a z13s A0x.

zIIP It

- Only certain types of work can run on a zIIP (Java, eg). In general, more recent versions of a product (DB2, for example), are more likely to run more work on a zIIP.
  - Under current IBM pricing rules, there is generally no price increase when you move to newer versions of a product, so always watch for zIIP enhancements in newer versions of products that use a lot of your capacity.
- You control what happens if there is a queue of work for a zIIP:
  - You can specify that all zIIP-eligible work MUST run on a zIIP.
  - Or, you can allow zIIP-eligible work to overflow to a general purpose CP.
  - A pending WLM APAR (OA50845) will allow you to control this at a more granular level. THIS IS GOOD!
- Traditionally, z/OS customers only consider an upgrade when CPU utilization gets very high.
  - With zIIPs, this is counter-productive. Work that runs on a zIIP is ‘free’, and zIIPs cost far less, *so you want to run as much work on zIIPs as possible*.
  - Rather than basing zIIP upgrades on utilization, base them on the amount of zIIP-eligible work that is running on general purpose CPs at the time of your peak R4HA. Aim to have enough zIIP capacity to minimize overflow to general purpose CPs.

Compression

- We seem to be swimming in oceans of data these days, and it is only going to get worse. This is putting financial and performance pressure on I/O subsystem (channels, switches, ports, disk subsystems, HDDs) components.
- One way to reduce the impact of growing data volumes is to compress your data:
  - Compression also reduces encryption costs because there is less data to encrypt.
- However, what compression giveth, compression also taketh away – compression is CPU intensive, which pushes up your capacity requirements and (potentially) your software bills.
- But there is good news – zEDC and z13…

Compression - zEDC

- z Enterprise Data Compression is a dedicated PCIe card (and associated SW interfaces) that compresses data at high speed and with better compression ratios than traditional compression. AND, because the compression is not done by a general purpose CP, the software cost of using zEDC is minimal.
  - Compression ratios are very data-dependent, but with SMF data, typical ratios are between 7:1 and 10:1. On average. zEDC seems to achieve about 2x the compression ration of traditional compression.
  - Typically reduces elapsed time and frees up capacity compared to traditional compression.
  - If you compress CICS and DB2 SMF records, you could turn off that compression, use zEDC instead and achieve better compression ratios AND reduce CPU usage by CICS and DB2.
- zEDC is available on zEC12 and later CPCs. Customer feedback has been universally very positive.

Compression – z13

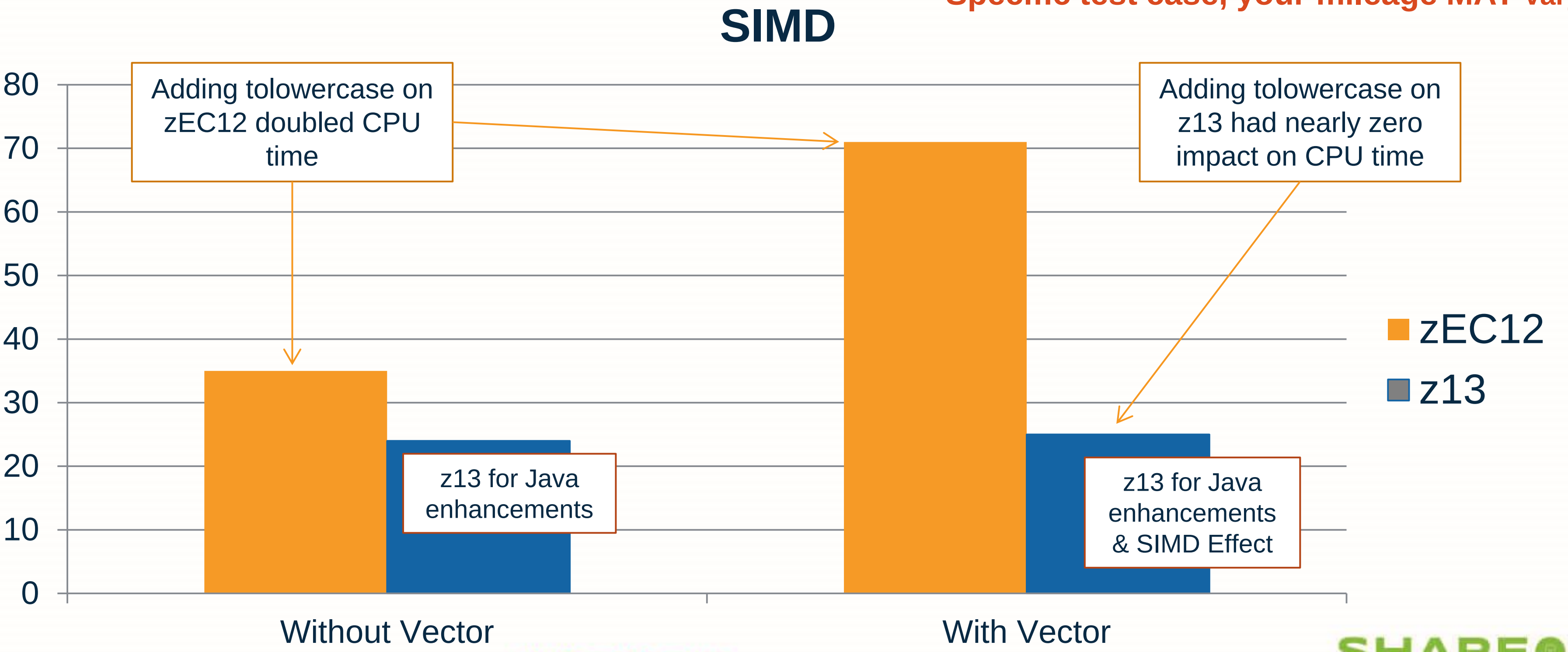
- Not all exploiters of traditional compression support zEDC, and not every site will purchase zEDC.
- For those users of traditional compression, z13 implemented changes to the hardware that performs the compression, resulting in CPU time savings of about 50% compared to zEC12.

SIMD

- Another enhancements on z13 was Single Instruction, Multiple Data (SIMD).
- SIMD lets you process multiple pieces of data with a single instruction.
- This provides the ability to more efficiently process large volumes of data – it is particularly suited to data crunching that is typical of analytics workloads, but also has other applications.
  - SIMD is exploited by programming languages to hide complexity from application developers.

SIMD

Specific test case, your mileage MAY vary !



Adding tolowercase on zEC12 doubled CPU time

Adding tolowercase on z13 had nearly zero impact on CPU time

z13 for Java enhancements

z13 for Java enhancements & SIMD Effect

The Future

- SIMD and zEDC and z13 compression changes are excellent examples of what we should expect to see in the future:
  - Because we can no longer rely on faster chip speeds to deliver across-the-board performance improvements, we should expect that more future enhancements will target specific work types or instructions.
  - Because the complexity of exploiting new hardware instructions should be transparent to applications, look for even closer synergy between the compilers and hardware design.

The Future

- Prior to COBOL V5, the programs created by the compiler could run on an early 9672 CPC (mid-1990s). This means that the compiler did not exploit any of the new CPC instructions delivered over the next 15 years.
- COBOL V5 lets you specify the target CPC type when compiling your programs.
- Typical CPU time savings are between 10% and 20% compared to COBOL V4, with up to 60% seen in some customers.
  - These are for a simple re-compile, no source changes.
- COBOL V5 also exploits SIMD for certain existing instructions:
  - Sample program without SIMD: 46.63 CPU Seconds
  - Same program compiled to use SIMD: 1.54 CPU Seconds

The Future – Recompile Considerations

- What this will mean for you:
  - Possible radical departure for strategy about recompiling programs.
  - Potential to see greater performance enhancements because *applications* (and not just the underlying middleware or operating systems) will be exploiting all available new functions.
  - Puts more responsibility on the customer to take overt action to obtain performance enhancements.
  - Cannot simply recompile and go – due to compiler changes, a program might ‘validly’ give a different result when compiled with a different level of the COBOL compiler.

Automatic Binary Optimizer (ABO)

- If you don’t want to (or are unable to) recompile your COBOL programs, an alternative is to use the Automatic Binary Optimizer.
  - This takes existing COBOL load modules, analyzes the code, and creates a new load module that exploits new instructions.
  - The benefit is expected to be less than recompiling the program, however the testing effort in expected to be reduced.
    - If the new load module does not produce identical output to the original, that is an APARable situation.
  - Unlike the COBOL Compiler, which only needs to be licensed in the LPAR where you compile your programs, ABO must be licensed in every z/OS LPAR.
    - However, IBM will work with you to find a cost model that ensures that ABO delivers cost savings.
  - ABO is not a no-brainer – need to determine if it will reduce your Peak R4HA. Also need to be careful with vendor programs – vendor might not agree to you optimizing their load modules.

Compilers

- Both COBOL V6 and ABO have a free 90-day Developer Trial.
- PL/I V4 and V5 compile code that exploits new hardware instructions – ARCH compile option lets you identify the target machine.
- C++ compiler also provides ability to target a specific architecture level.
- Java 8 has significant performance enhancements for z13 AND Java will automatically exploit the architecture level of the machine it is running on without having to recompile.
    - Despite significant performance enhancements in Java 7 and Java 8, we still see widespread use of older Java versions.
- To get the maximum value from future hardware enhancements, z/OS customers need to be more proactive about maintaining compiler currency.

Analytics

- One of the hottest topics in the IT/Business world is how to use your vast volumes of data to reduce fraud, protect against hacking, and generate more business.
- There are a number of Analytics products available today, mainly on distributed platforms.
  - The products are open source, so distributed is the natural environment for them.
  - Distributed hardware and software are less expensive than traditional z/OS hardware and software.
  - The data that they have been analyzing often originates on distributed platforms or is historical data that originated on z/OS.
- The question of how you can use that paradigm for realtime analytics with z/OS transactions has been challenging early implementers.

Analytics

- Apache Spark is an open source analytics framework and one of the fastest growing open source projects (advertised as 100x faster than Hadoop).
- In 2016, IBM announced the z/OS Platform for Apache Spark. This consists of:
  - The Spark framework. Currently Spark 2.0.2.
  - Rocket Software's MDS to retrieve data with maximum parallelism.
- This allows customers to exploit:
  - Existing Spark skills from other platforms.
  - Java programming skills.
  - Huge memory available on z Systems. This is ideal for Spark's Resilient Distributed Datasets (RDDs) and enables processing of huge volumes of data in minimal time.
  - z/OS' industry-leading I/O capability.
  - Close to zero latency between the running transaction, MDS, and Spark.

Analytics

- The business case for Spark on z/OS is difficult to ignore:
  - There is no MLC or OTC for Spark on z/OS. S&S is optional and is a fixed price per CPC, regardless of CPC size or how many LPARs run Spark.
  - The Spark infrastructure is 100% zIIP-eligible. (runs on cheap zIIP engines)
  - All user programs that run under Spark are 100 zIIP-eligible. (Ditto)
  - MDS is 88% zIIP eligible when retrieving data from DB2, and 99+% zIIP-eligible when retrieving sequential data. (Ditto)
  - IBM is offering significant discounts on the (already cheap) zIIPs used by Spark.
  - And significant discounts on memory purchased for use by Spark.
  - IBM is offering no-charge 90-day Proof of Concept projects that include loaner zIIPs and memory and access to IBM Spark experts.
  - If the data that you want to analyze originated on z/OS, if the task that needs the results of the analysis is running on z/OS, if you can reuse existing skills, if you are concerned about the security considerations of having multiple copies of sensitive data, and if the hardware and software costs are comparable, why (and how!) would you send the data and the request to another platform just so the answer can be sent back to z/OS?

Machine Learning and Watson

- We have all heard about IBM’s Watson brand of cognitive products and services.
- But did you know that IBM’s Watson Machine Learning technology is currently only available on two platforms:
  - As a cloud service (if you don’t mind sending your data outside your company).
  - **On z/OS**.
- IBM is deploying its machine learning tools on z/OS first because that is where the majority of revenue-generating operational and transaction data resides.
- The Machine Learning function is based on Spark, so all (98+%) of the work is zIIP-eligible (cheaper engines, no impact on MLC).
- For more information about IBM Machine Learning for z/OS, see Lunch and Learn Session 20815 *Machine Learning*.

Summary

- The end of Moore’s Law will bring back the need for performance management skills and activities – just because chip speeds have plateaued does *not* mean that your work volumes will stop rising.
- In order to get more work out of your hardware, and to run your work at the lowest cost, more focus will need to be placed on LPAR configuration and RNI management.
- There are many things that can be done *today*, that many companies are not doing, that can improve your performance (*and* reduce your costs) immediately.
- In the future, expect performance improvements to target specific subsets of your workloads. Work that can exploit these enhancements may see significant performance improvements. But work that does not, might see no improvement at all.
  - If you chargeback for use of your systems, this will present you with a challenge.
- Expect more focus on performance enhancements being delivered via the compilers, meaning that programs must be recompiled to see the benefits.

Summary

- Always take *all* your costs into account. For example, if you don't use compression today, turning on zEDC might slightly increase your peak R4HA – but the savings on DASD space and I/O subsystem load could be larger than the CPU cost.
- Remember that incremental MSUs always cost less than the average.
  - So growing your workload will likely *cost* less than you expect, AND because the additional MSUs are at the lower, incremental, rate, this reduces your overall average $ per MSU.
  - Conversely, shrinking will probably *save* you less than you expect, AND it has the effect of increasing the overall average $ per MSU because you have taken away the cheapest MSUs.
- Memory is cheap and zIIPs are cheap – USE THEM!

Summary

- In general, strive to stay on current releases. There is no rising tide that lifts all ships any more, meaning that more performance will come from software changes and software exploitation of new hardware capabilities.
  - With IBM's pricing changes, and the recent Multi-Version Measurements announcement, there is no financial justification for staying on old versions. In fact, staying on old versions might be costing you more because you will be missing out on enhancements that are only available in the latest versions.
- IBM are serious about getting new workloads onto z/OS – reference z/OSMF, Mobile Workload Pricing, zCAP, zWPC, Spark on z/OS, and now Machine Learning. The underlying technology on z Systems means that the mainframe should be *at least* as good a platform as any other to deploy new applications. If IBM can address the cost difference, this should lead to a resurgence in z/OS workload growth, putting pressure on you to deliver the required performance while controlling your costs. The tools (hardware, software, and pricing options) are there to help you be successful, you just need to be aware of them and to make the optimum use of them.

References

- For an informative guide to z Systems processor architecture and enhancements to deliver more capacity and performance, refer to SHARE in Atlanta Session 19840, *IBM z Systems Processor Optimization Primer*, by **Kevin Shum**.
- For information about how IBM measures its mainframes and assigns MSU values, see IBM's Large Systems Performance Reference (LSPR).
- To get more information about how RNIs are calculated using data in the SMF type 113 records, see Session 19658 (SHARE in Atlanta) *CPU MF Update* by **John Burg**.
- IBM Redpaper REDP-4727 *Setting Up and Using the IBM System z CPU Measurement Facility with z/OS* provides information about how to set up CPU MF and start collecting SMF type 113 records.
- For more information about understanding and tuning your RNI, refer to a series of articles about the CPU Measurement Facility starting with the 2016 No. 4 issue of Cheryl Watson's Tuning Letter.

References

- IBM Redbook SG24-8251, *z13 Technical Guide* provides valuable information to help you optimize your memory buying decision.
- For more information about the benefits of large memory, refer to IBM Redpaper *Benefits of Configuring More Memory in the IBM z/OS Software Stack*.
- For information about DB2 use of large pages, see:
  - *Utilizing Large Memory with DB2 12*, by **Akiko Hoshikawa**
  - *DB2 12 for z/OS: Memory Exploitation for Performance Benefits* by **Mark Rader**
  - *DB2 xx for z/OS Performance* Redbooks (www.redbooks.ibm.com)
- For information about large page exploitation and management, refer to ‘Controlling Large Pages’ in *Cheryl Watson’s Tuning Letter 2014 No. 2*.
- ‘HiperDispatch Questions and Answers’ article in *Cheryl Watson’s Tuning Letter 2015 No. 4* provides information to help you optimize your LPAR configuration.
- ‘Holistic Capacity Planning’ article in *Cheryl Watson’s Tuning Letter 2015 No. 4* provides information about one customer’s experiences with z13 processor performance/

Phew, we made it!!

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.

Mainframe Performance and You