If you are not making maximum use of your zIIP processor to run DB2 for z/OS work the amount of money you are wasting could cause you embarrassment. According to industry calculations, hardware plus software costs for a zIIP processor is $150 to $200 per MIPS compared with $2,200 to $3,400 for a general purpose processor.
In addition different software vendors will make claims about how much of their code is run on the zIIP engine, and as a customer of these vendors it's necessary to validate their claims.
In this instance I will use the DB2 for z/OS utilities as a case study, specifically LOAD,REORG and Rebuild Index to calculate what percent of the total CPU time used by each utility is offloaded to the zIIP processor.
Most DB2 monitors will report on accumulated CPU seconds for the 4 DB2 for z/OS started tasks of:
MSTR , DBM1 ,IRLM ,DIST
And for each task the following categories of accumulated CPU seconds will be recorded:
TCB CPU , SRB CPU , Pre-emptible SRB , pre-emptible ZIIP
The presentation will outline a method using Chorus the DB2 DBA role to bench mark what percentage of CPU seconds have been offloaded to the zIIP processor, using a dedicated DB2 for z/OS subsystem that is not executing any work apart from the utility.

Key Bullet Points from the Presentation.

- The Cost Benefits of using the zIIP processor.
- Task Control Blocks, Service Request Blocks and….
  - Enclave Service Request Blocks managed by Work Load Manager that are pre-emptible.
- **The Problem 1** – Why were our monitors saying only 5 or 6% of the IBM's DB2 for z/OS utilities workload is offloaded to zIIP?
- **The Problem 2** – IBM's DB2 for z/OS Utilities, this amount of zIIP use did not ring true, how much of the workload is really offloaded to zIIP?
- Chorus and the DB2 DBA role, how we used it to solve these 2 problems.
- Review of zIIP use and eligible DB2 for z/OS utility workloads.

First to provide some estimation of the costs savings that can be achieved by moving DB2 for z/OS workloads to the zIIP processor. Next to simply explain Task Control Blocks, Service Request Blocks and work load manager, because a basic understanding of these concepts is required to understand how work is allocated to run on the zIIP processor. We had a specific issue to measure how much work IBM DB2 for z/OS utilities is offloaded to the zIIP. Initial findings indicated relative low amount of work offloaded around 5 or 6%. Did not ring true so we needed to dig deeper to find out hopefully the complete picture of zIIP use. Will give some back ground about Chorus and the DB2 for z/OS DBA role and how this product helped us to solve these 2 problems. Finally some general information about zIIP use and DB2 workloads that are eligible to run on the zIIP.

Comparing the Costs.

- According to industry calculations, hardware plus software costs for a zIIP processor is **$150 to $200** per MIPS.
- Compared with **$2,200 to $3,400** for a general purpose processor.
- It is a necessity to first measure zIIP utilization, while planning to further exploit the zIIP in the future.
- So failure to check your zIIP will cause embarrassment in an age where we need to do more with less!

No notes.

Claims by Software Vendors.

- There is some debate around how to best exploit the zIIP.
- It's an opportunity for some 'wild' claims about zIIP exploitation.
- As a customer before purchasing software it is necessary to validate these claims.
- How much DB2 for z/OS workload can be offloaded to the zIIP depends to a great extend on the data structure and usage.

No notes

How to Check your sites zIIP Installation

- Console Command: **D M=CPU**

```
IEE174I 10.30.19 DISPLAY M 470
PROCESSOR STATUS
ID  CPU          SERIAL
00  +            0C4D072827
01  +            0C4D072827
02  +            0C4D072827
03  +            0C4D072827
04  +            0C4D072827
05  +            0C4D072827
06  +I           0C4D072827
07  +I           0C4D072827
08  +I           0C4D072827

I      INTEGRATED INFORMATION PROCESSOR (zIIP)
```

No notes.

Task Control Blocks .

- **Task control block** (TCB) - represents a unit of work or task, such as an application program, that runs in an address space, and is pre-emptible. It runs at the dispatching priority of the address space.
- Pre-emption – is the act of temporarily interrupting a task without requiring its cooperation, with the intention of resuming the task at a later time.
- Typical decision for a TCB is to swap it out of execution due to a lack of or need to wait for a system resource.

In effect the TCB is doing the 'real' work of the application its processing data according to the application design. The z/OS system needs to manage a complex work load and part of this process involves interrupting or pre-empting the task.

Service Request Blocks.

- **Service request block** (SRB) - represents a request for a system service and is non pre-emptible. It runs at supervisor priority.
- SRBs are typically created when one address space detects an event that affects a different address space
- SRBs provide a mechanism for communication between address spaces.
- A pure 'old style' SRB cannot be pre-empted, simply it was a system service request, so it got executed without any interruptions before execution.

However the TCB cannot run in isolation, it needs to talk to the other parts of the z/OS system. When a TCB needs information from say another address space it will send a Service Request Block or SRB. Before zIIPs and work load manager came along the SRB could not be pre-empted it was simply a service request so it got executed at a relatively high priority.

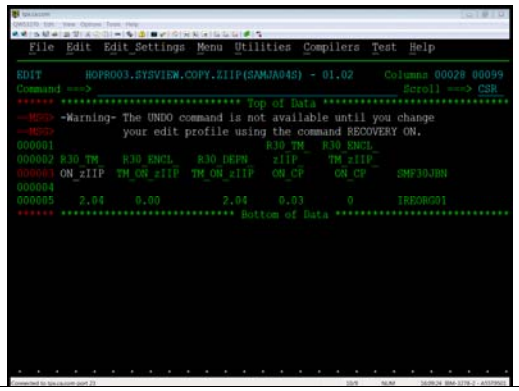
Enclave SRBs that are pre-emptible.

- Work Load Manager (WLM) manages the performance goals of a unit of work that can span a number of address spaces, called an enclave.
- The zIIP processor, due to restrictions from IBM, can only run SRB workloads and ……
- these **SRB workloads must be a WLM enclave**.
- Enclave SRB requests are pre-emptible so WLM can make the decision to either run the work on the zIIP or not.
- Enclave SRB runs at the dispatching priority of the enclave.

Now we have work load manager and zIIP processors so the old SRB was changed. The SRB can now be pre-empted and is run at a priority determined by the policies coded in WLM. The SRB is defined as an 'unit of work' or enclave. Now we have an SRB enclave so the work is now eligible to run on the zIIP processor.

Problem 1.

- We know IBM DB2 for z/OS utilities exploit the zIIP processor.
- Analyzed **SMF30_TIME_ON_ZIIP** SMF records to determine how many zIIP CPU seconds were used by the utilities batch job.



No notes.

Problem 1 contd.

- Results said around **5% to 6 % zIIP exploitation**.
- We know IBM push some of the DB2 for z/OS utilities workload through the DB2 for z/OS DBM1 address space.
- We made the assumption that IBM further exploited the zIIP by directing some work load to the DBM1 address space.

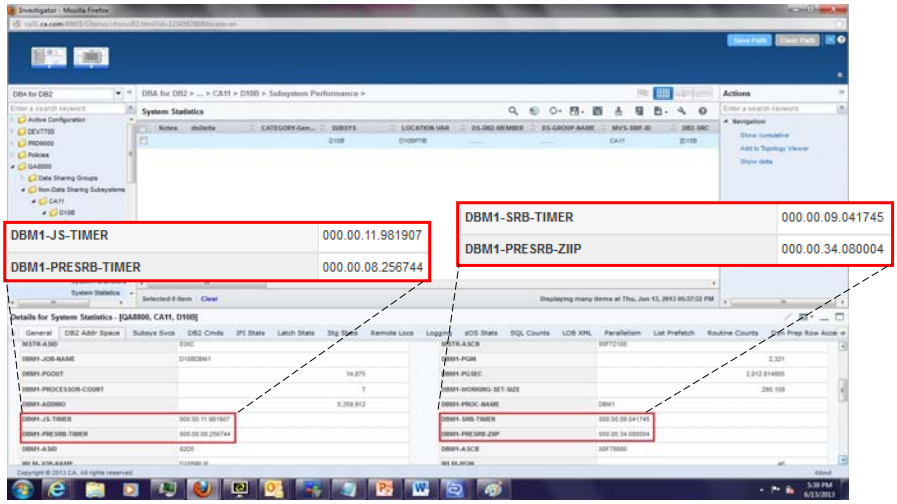
Initial findings indicated very low zIIP use, however we know from previous experience that IBM direct some of the DB2 for z/OS utilities workload to the DBM1 address space, and if this was the case the zIIP seconds would be recorded against the DBM1 address space.

Problem 1 contd.

- Chorus the DBA role monitors CPU seconds for the 4 DB2 for z/OS address spaces, that is zIIP and General Purpose time.
- From my observations MSTR and IRLM use zero zIIP seconds.
- Generally DIST will exploit the zIIP for distributed workloads.
- The DBM1 address space may be used to run DB2 for z/OS utilities workloads.
- Isolated a DB2 sub system so it only runs a DB2 for z/OS utilities workload and measure the before and after CPU seconds.
- Confirmation zIIP is exploited via the DBM1 address space!

No notes.

Chorus DBA role displaying zIIP CPU seconds.



This shows a screen shot from Chorus and the monitoring of the DBM1 address space:

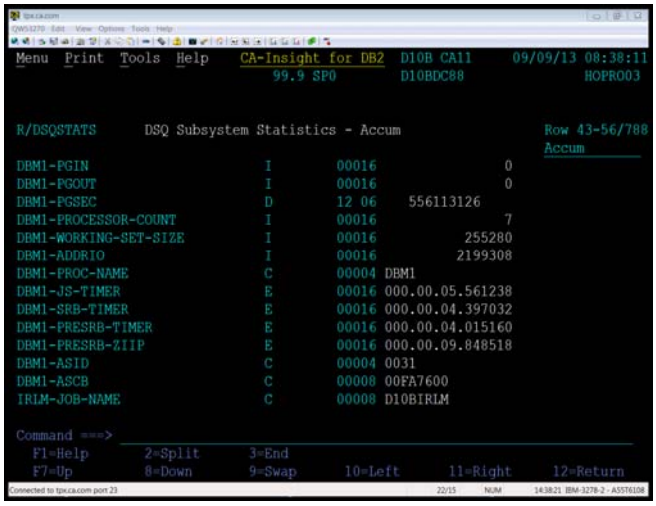
JS-Timer- this the CPU seconds for the TCB of the DBM1 address space.

SRB-Timer- this is the accumulated SRB timer value for the address space **EXCLUDING ZIIP** time.

PRESRB-Timer- this is the amount of pre-emptible SRB time for the DBM1 address space **EXCLUDING ZIIP** time.

PRESRB-zIIP- this is the amount of pre-emptible SBB time run on the zIIP processor. On this DB2 sub system IBM DB2 utilities have been executed, this is why we see relatively large zIIP count of 34s compared to the other CPU values

Insight Provides the Data to Chorus



Screen shot showing raw Insight data in a 3270 display.

Problem 2.

- Now we want to measure the % of the total DB2 for z/OS utilities workload run on the zIIP processor.
- Again isolate a DB2 for z/OS subsystem, no other users.
- Execute the IBM DB2 for z/OS utilities batch job and note number of CPU seconds (zIIP and general purpose) consumed by the job.
- Monitor the 4 DB2 for z/OS address spaces noting the increase in CPU seconds for General Purpose and zIIP seconds.
- Add up these CPU seconds.

No notes.

Problem 2 contd.

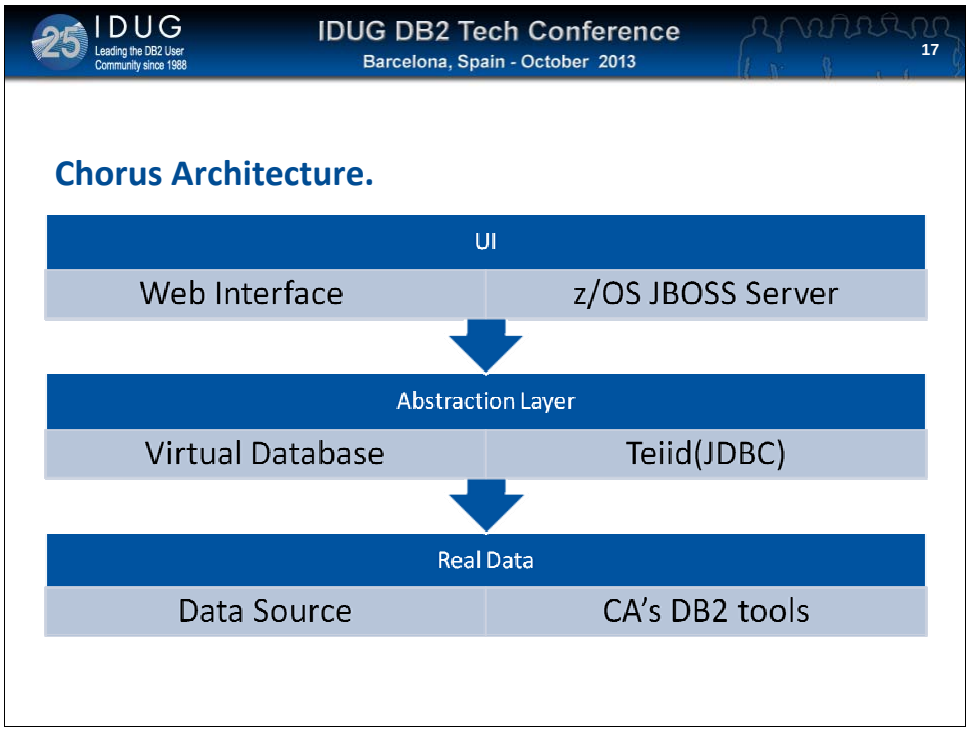
- Now we have total number of CPU seconds used to complete this IBM DB2 for z/OS utility.
- We know the number of zIIP seconds consumed by the DBM1 address space and the utilities batch job.
- With some simple mathematics we now know what % of the total workload was run on the zIIP.
- zIIP seconds / Total CPU seconds * 100

No Notes.

Results from our tests…but please run your own!

- Table - Partition by Range 16 partitions, 37m rows and 2 Index.
- **V9 Build Index 14%** offloaded to the zIIP.
- **V10 Build Index 27%** offloaded to the zIIP.
- **V9 Load 7.5%** offloaded to the zIIP.
- **V10 Load 10%** offloaded to the zIIP.
- Table – Partition by Growth 2 partitions, 1.1m rows and 1 Index
- **V9 Reorg 14%** offloaded to the zIIP
- **V10 Reorg 36%** offloaded to the zIIP

Did some very basic tests and these are the results BUT use of zIIP is very much dependent on data structure, results are by no means conclusive but a guide or ball park figure. Maybe next year I will propose a presentation on data structures that make extensive use of the zIIP?



The web interface to Chorus is provided by a JBOSS Server running under z/OS. The JBOSS server accesses the DB2 performance data via a virtual database Teiid, this allows data to be retrieved with standard SQL statements. CA's DB2 for z/OS tools provides the actual data to be displayed in the UI, for example Insight provides the CPU data we required.

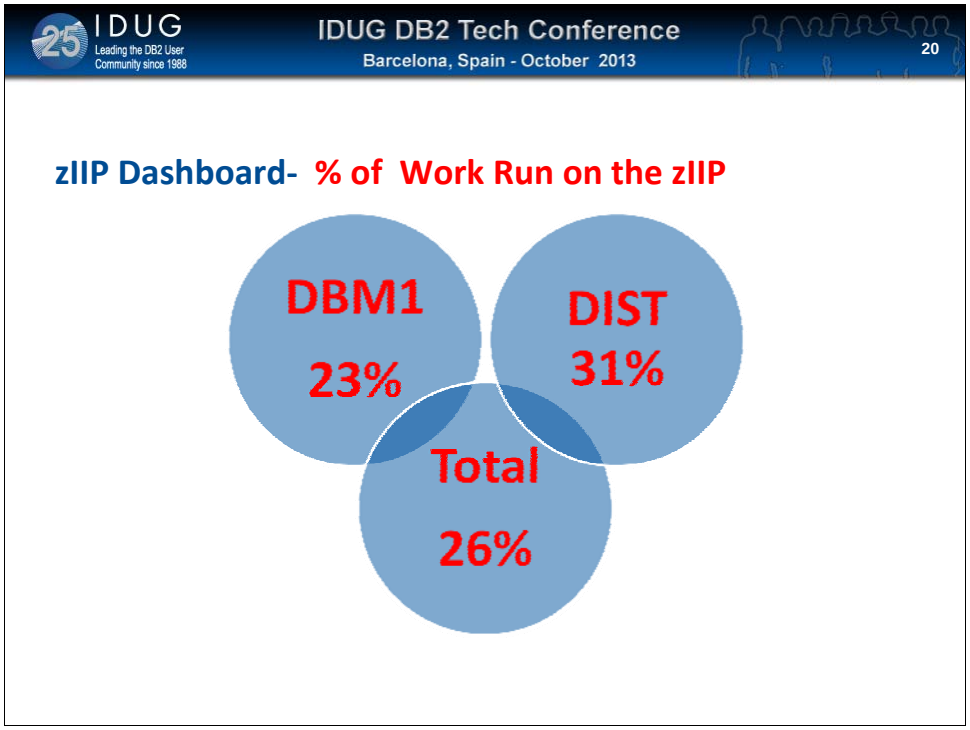
How Chorus Architecture Helped us! Better by Design!

- Chorus uses a Virtual Database called Teiid which provides in built flexibility.
- Using a JDBC connector it's possible to create a customized query for zIIP monitoring purposes.
- Simple SQL:
- SELECT 'CPU seconds' FROM Table …
- Created a simple application for getting the CPU times, zIIP and General Purpose for the 4 DB2 address spaces, and write them to a file for analysis.

The Future Maybe?

- A nice zIIP dashboard displayed in our performance monitors. So we can see zIIP use at a glance.
- I am selling the idea to CA's Product Management.
- Shout out to your performance product vendors if you want it!

Its not a straight forward job to measure your zIIP use for DB2 for z/OS workloads, so I suggest we implement a zIIP dashboard so at a glance zIIP use is displayed. Generally it would be good to demystify zIIP use and measurement.



The main users of zIIP will be the DBM1 and DSIT address spaces.

Types of DB2 for z/OS Utilities Workloads Suitable for the zIIP.

- DB2 Utilities LOAD, REORG & REBUILD.
- DB2 utility functions used to maintain index structure and sort.
- Sorting for LOAD, REORG , REBUILD, & RUNSTATS.

No Notes.

High End to Low End zIIP Utilization.

Lower end of range is expected with:

- Tables with fewer Indices.
- Fewer partitions for Partition Utility.
- Compression used.

Higher end of range is expected with:

- Tables with many Indices or many partitions for Partition Utility.

No Notes.

Problems with Over Loaded zIIP processors

- If the zIIP is over loaded it can ask for help from the standard processor, if system parameter IIPHONORPRIORITY is set to yes. We sent it to zIIP now its coming back!
- This makes **performance and financial NONSENSE**
- A merry go round performance problem.
- If IIPHONORPRIORITY is set to no the workload waits for some spare zIIP capacity, another performance bottleneck.
- Rule of thumb - **Run your zIIP processors at 50% capacity**.
- Have a **one to one ratio zIIP to general processor**.

Over loaded zIIP processors are a situation to be avoided, because you lose the opportunity to use MIPS at $200 and are paying maybe $3,400. And you paid on over head to send the work to the zIIP then sent it right back. Spare capacity on zIIP may not be a wasted resource if it avoids zIIP eligible work being run on a general processor.

A Quick Review of zAPP and zAPP on zIIP.

- zAAP – Application Assist Processor is designed to run JAVA workloads.
- zAPP on zIIP – Was introduced to help manage workloads, for example if zAPP was over loaded it could use spare capacity on the zIIP, or if a customer had a small amount of JAVA workload that did not justify purchasing a zAPP processor he could direct the JAVA workload to the zIIP.
- zEC12 is planned to be the last high-end System z server to offer support for zAAP specialty engine processors.

No Notes.

The Latest News on zIIPs from IBM.

- IBM has changed the zIIP to central process (CP) ratio for both the zEC12 and the zBC12
- For these two machines, IBM has announced that the zIIP to CP ratio will be increased to 2:1

No Notes.

Robin Hopper
CA Technologies
robin.hopper@ca.com

Session Code: A06

Not Checking your zIIP can lead to Embarrassment!

INTERNATIONAL DB2 USERS GROUP
Independent • Not-for-Profit • User Run
Founded 1988