Connect the Dots: A z13 and z/OS Dispatching Update

What I hope to cover......

- What are dispatchable units of work on z/OS
- How WLM manages dispatchable units of work
- The role of HiperDispatch and Warning Track
- Dispatching work to zIIP engines
- z13 Simultaneous Multithreading (SMT)
There are different types of Dispatchable Units (DU's) in z/OS

- Preemptible Task (TCB)
- Non Preemptible Service Request (SRB)
- Preemptible Enclave Service Request (enclave SRB)
  - Independent - a new transaction
  - Dependent – extend existing address space
  - Work-dependent – extend existing independent enclave
z/OS dispatching work

WLM policy adjustment algorithm
**WLM dispatching priority usage**

<table>
<thead>
<tr>
<th>Priority</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>255</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>254</td>
<td>SYSSTC</td>
</tr>
<tr>
<td>253</td>
<td>Small Consumer</td>
</tr>
<tr>
<td>252</td>
<td>Priorities for dynamic policy adjustment</td>
</tr>
<tr>
<td>208</td>
<td>Not used</td>
</tr>
<tr>
<td>201</td>
<td>Discretionary work Mean Time to w1t algorithm</td>
</tr>
<tr>
<td>192</td>
<td></td>
</tr>
</tbody>
</table>

**Dispatching in an LPAR environment**
**HiperDispatch mode**

- **PR/SM**
  - Supplies topology information/updates to z/OS
  - Ties *high priority* logicals to physicals (gives 100% share)
  - Distributes remaining share to *medium priority* logicals
  - Distributes any additional service to unparked *low priority* logicals

- **z/OS**
  - Ties tasks to small subsets of logical processors
  - Dispatches work to *high priority* subset of logicals
  - Parks *low priority* processors that are not needed or will not get service

- **Hardware cache optimization occurs when a given unit of work is consistently dispatched on the same physical CPU**

**HiperDispatch: z/OS part**

- **z/OS obtains the logical to physical processor mapping in HiperDispatch mode**
  - Whether a logical processor has high, medium or low share
  - On which book and chip the logical processor is located

- **z/OS creates dispatch nodes**
  - The idea is to have 4 high share CPs in one node
  - Each node has TCBs and SRBs assigned to the node
  - Optimizes the execution of work units on z/OS
**RMF CPU activity report**

### HiperDispatch and LPAR

#### PARTITION DATA REPORT

<table>
<thead>
<tr>
<th>Name</th>
<th>S</th>
<th>WGT</th>
<th>DEF</th>
<th>ACT</th>
<th>Def</th>
<th>WLM%</th>
<th>Num</th>
<th>Type</th>
<th>Effective</th>
<th>Total</th>
<th>Effective</th>
<th>Total</th>
<th>LPAR MGMT</th>
<th>Effective Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPAR1</td>
<td>A</td>
<td>494</td>
<td>0</td>
<td>582</td>
<td>NO</td>
<td>0.0</td>
<td>32.0</td>
<td>CP</td>
<td>02.17.24.319</td>
<td>02.20.44.154</td>
<td>28.63</td>
<td>29.32</td>
<td>0.44</td>
<td>17.96</td>
</tr>
<tr>
<td>LPAR2</td>
<td>A</td>
<td>446</td>
<td>0</td>
<td>762</td>
<td>NO</td>
<td>0.0</td>
<td>32.0</td>
<td>CP</td>
<td>03.01.28.607</td>
<td>03.04.05.167</td>
<td>37.81</td>
<td>38.35</td>
<td>0.34</td>
<td>23.72</td>
</tr>
<tr>
<td>LPAR3</td>
<td>A</td>
<td>59</td>
<td>0</td>
<td>0</td>
<td>NO</td>
<td>0.0</td>
<td>1.0</td>
<td>CP</td>
<td>00.00.00.000</td>
<td>00.00.00.000</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>LPAR5</td>
<td>A</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>NO</td>
<td>0.0</td>
<td>1.0</td>
<td>CP</td>
<td>00.00.00.000</td>
<td>00.00.00.000</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**TOTAL AVERAGE**

<table>
<thead>
<tr>
<th>Name</th>
<th>S</th>
<th>WGT</th>
<th>DEF</th>
<th>ACT</th>
<th>Def</th>
<th>WLM%</th>
<th>Num</th>
<th>Type</th>
<th>Effective</th>
<th>Total</th>
<th>Effective</th>
<th>Total</th>
<th>LPAR MGMT</th>
<th>Effective Total</th>
</tr>
</thead>
</table>

### CPC CPU Activity

<table>
<thead>
<tr>
<th>CPU</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
<th>Effective</th>
<th>Total</th>
<th>Effective</th>
<th>Total</th>
<th>LPAR MGMT</th>
<th>Effective Total</th>
</tr>
</thead>
</table>

**TOTAL AVERAGE**

Total LPAR weight = 1000

LPAR1 = 494/1000 = 49.4% at 53 CPs = 26.18 CPs

LPAR2 = 446/1000 = 44.6% at 53 CPs = 23.84 CPs

LPAR1 = 25 VH and 2 VM at 59% share (27 logicals unparked)

LPAR2 = 23 VH and 1 VM at 64% share (24 logicals unparked)

51 logicals unparked
Dispatching in an LPAR environment

Warning track

- In a PR/SM™ environment the LPAR hypervisor assigns physical engines to logical engines accordingly to the weighting factors of the partitions.
- Once the time slice for a logical engine is expired the currently executing work is suspended until a physical engine is assigned to the logical engine again.
- The Warning Track Interruption Facility notifies the operating system that PR/SM™ will undispach a certain logical processor within the next 50 microseconds (grace period).
- z/OS is now able to save status for the running unit of work and re-dispatch the work unit on a different logical processor within the grace period.
- z/OS now signals to PR/SM via Diagnose x'9C' that the logical processor can be un-dispatched.
- Warning Track processing is only supported in HyperDispatch=YES environments.
- A high benefit can be achieved for Low Share processors which might be parked by WLM.
Warning track

Latent demand: LPAR Busy vs MVS Busy

<table>
<thead>
<tr>
<th>CPU</th>
<th>2097</th>
<th>CPC CAPACITY 1451</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>CP</td>
<td>100.00</td>
</tr>
<tr>
<td>1</td>
<td>CP</td>
<td>100.00</td>
</tr>
<tr>
<td>2</td>
<td>CP</td>
<td>100.00</td>
</tr>
<tr>
<td>3</td>
<td>CP</td>
<td>100.00</td>
</tr>
<tr>
<td>4</td>
<td>CP</td>
<td>100.00</td>
</tr>
<tr>
<td>5</td>
<td>CP</td>
<td>100.00</td>
</tr>
<tr>
<td>6</td>
<td>CP</td>
<td>100.00</td>
</tr>
<tr>
<td>7</td>
<td>CP</td>
<td>100.00</td>
</tr>
<tr>
<td>8</td>
<td>CP</td>
<td>100.00</td>
</tr>
<tr>
<td>9</td>
<td>CP</td>
<td>100.00</td>
</tr>
<tr>
<td>A</td>
<td>CP</td>
<td>100.00</td>
</tr>
<tr>
<td>B</td>
<td>CP</td>
<td>100.00</td>
</tr>
<tr>
<td>C</td>
<td>CP</td>
<td>100.00</td>
</tr>
<tr>
<td>D</td>
<td>CP</td>
<td>100.00</td>
</tr>
<tr>
<td>E</td>
<td>CP</td>
<td>100.00</td>
</tr>
<tr>
<td>F</td>
<td>CP</td>
<td>100.00</td>
</tr>
<tr>
<td>10</td>
<td>CP</td>
<td>100.00</td>
</tr>
<tr>
<td>TOTAL/AVERAGE</td>
<td>42.47</td>
<td>91.45</td>
</tr>
</tbody>
</table>
**Warning track statistics**

- RMF keeps track of the number of times PR/SM issued a warning-track interruption to a logical processor and z/OS was able/unable to return the logical processor within the grace period.
- RMF measures the amount of time in microseconds that a processor was yielded to PR/SM due to Warning-track processing.

<table>
<thead>
<tr>
<th>SMF record type 70 subtype 1 (CPU Activity) – CPU data section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>80</td>
</tr>
<tr>
<td>84</td>
</tr>
<tr>
<td>88</td>
</tr>
</tbody>
</table>

**RMF Postprocessor Overview Conditions**

<table>
<thead>
<tr>
<th>Name</th>
<th>Qualifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WTRKCP (WTR(KA)AP1) (WTRKSP)</td>
<td>cpu-id</td>
<td>The percentage of times PR/SM issued a warning track interruption to a processor and z/OS was able to return it to PR/SM within the grace period.</td>
</tr>
<tr>
<td>WTRKCP (WTR(KT)AP1) (WTR(KTIP))</td>
<td>cpu-id</td>
<td>Time in microseconds that a purpose processor was yielded to PR/SM due to Warning Track processing.</td>
</tr>
</tbody>
</table>

---

**WLM Topology Report Tool**

- New *as-is* tool available for download from the WLM homepage
- Visualizes mapping of HiperDispatch affinity nodes to physical structure
- Supports IBM zEC10 and later

To use:
1. Download from above location
2. Run installer
3. Collect SMF99.14 records
4. Upload Host code to a z/OS system

Sample output (z/13):

```
Topology for 01-30-2015-14-38-32, System: SYSD
```

Sample output (z/13):
```
```

(c) 2014 IBM Corporation
Dispatching zIIP eligible work

IBM z Integrated Information Processor (zIIP) on the z13

- The IBM z13 continues to support the z Integrated Information Processor (zIIP) which can take advantage of the optional simultaneous multithreading (SMT) technology capability. SMT allows up to two active instruction streams per core, each dynamically sharing the core’s execution resources.
  - With the multithreading function enabled, the performance capacity of the zIIP processor is expected to be up to 1.4 times the capacity of these processors on the zEC12

- The rule for the CP to zIIP purchase ratio is that for every CP purchased, up to two zIIPs may be purchased

- zAAP eligible workloads such as Java and XML, can run on zIIPs using zAAP on zIIP processing

- zAAPs are no longer supported on the z13
Current IBM exploitation of zAAPs and zIIPs

<table>
<thead>
<tr>
<th>Specialty CP</th>
<th>Eligible</th>
<th>Major Users</th>
</tr>
</thead>
</table>
| zAAP or zIIP on z13 | Any Java Execution | WebSphere  
                     CICS  
                     Native apps  
                     XMLSS |
| zIIP          | Enclave SRBs                  | DRDA over TCP/IP  
                     DB2 Parallel Query  
                     DB2 Utilities Load, Reorg, Rebuild  
                     DB2 V9 z/OS remote native SQL procedures  
                     TCP/IP - IPSEC  
                     XMLSS  
                     zIIP Assisted HiperSockets Multiple Write  
                     Virtual Tape Facility Mainframe (VTFM) Software  
                     z/OS Global Mirror (XRC), System Data Mover (SDM)  
                     z/OS CIM Server  
                     RMF Mon III  
                     OMEGAMON on z/OS and DB2  
                     IMS Ver 8 |

What is "Needs Help?"

- Determination zIIP or zAAP work is being delayed and additional resources should help process the work
  - Requires xxPHONORPRIORITY=YES to be set

- If help is required:
  - The zxxP CP signals waiting zxxP to help
  - When all zxxP CPs are busy the zxxP asks for help from the GCP
    - All available speciality engines (of the same type) must be busy before help is asked of the GCPs
      - IF the zxxPs needs help and all zxxPs are busy help is obtained from 1 GCP
      - IF zxxPs continue to need help additional CPs may be asked to help
  - Help is always provided in dispatch priority order
Specialty CP work running in a WLM service class

REPORT BY: POLICY=WLMPOL  WORKLOAD=BAT_WKL  SERVICE CLASS=BATSPEC  RESOURCE GROUP=BATMAXRG

TRANSACTIONS  TRAN-S-TIME HH.MM.SS.TTT  --DASD I/O--  ---SERVICE----  SERVICE TIMES  ---APPL %---
AVG 0.98 ACTUAL 6.520 SSCHRT 11.5 IOC 8326 CPU 24.7 CP 0.97
MPL 0.98 EXECUTION 6.128 RESP 7.0 CPU 662386 SRB 0.0 AAPCP 0.01
ENDED 10 QUEUED 391 CONN 6.9 NSO 0 RCT 0.0 IIPCP 0.00
END/S 0.17 R/S AFFIN 0 DISC 0.0 SRB 0 IIT 0.0
$SWAPS 0 INELIGIBLE 0 Q+PEND 0.1 TOT 671677 HST 0.0 AAP 40.27
.EXCTD 0 CONVERSION 0 IOSQ 0.0 /SEC 11195 AAP 24.2 IIP 0.00
AVG ENC 0.00 STD DEV

GOAL: EXECUTION VELOCITY 35.0%  VELOCITY MIGRATION: I/O MGMT 99.2%  INIT MGMT 92.2%

RESPONSE TIME EX  PERF  AVG ------ USING% ------ ------------ EXECUTION DELAYS % -------

SYSTEM VEL% INDX ADRSP CPU AAP IIP I/O TOT CPU
SYSD --N/A-- 99.2 0.4 1.0 0.8 45.9 0.0 3.9 0.4 0.4

RMF report is at 1 minute interval

DB2 parallel query, enclave SRBs and zIIPs

Have been independent enclave SRBs to be zIIP eligible. Beginning in z/OS R11 the child tasks are now work-dependent enclaves.

Portions of complex query arrive on participant systems, classified under "DB2" rules, and run in enclave SRBs, so zIIP eligible

(c) 2014 IBM Corporation
Implement a new type of enclave named "Work-Dependent" as an extension of an Independent Enclave. A Work-Dependent enclave becomes part of the Independent Enclave's transaction but allows to have its own set of attributes (including zIIP offload percentage).

In cases where DRDA applications create extended duration work threads in DB2, for example through extensive use of held cursors, the zIIP utilization levels can become more variable. DB2 and DDF now may use work-dependent enclaves in this situation to control this variability. See APAR PM28626.
Work-dependent enclaves in SDSF

zIIP processors and simultaneous multithreading

zIIP eligible work units

zIIP logical cores with 1 or 2 threads

zIIP logical cores with 1 or 2 threads

zIIP physical cores with 1 or 2 threads
z13 - Simultaneous Multithreading (SMT)

- “Simultaneous multithreading (SMT) permits multiple independent threads of execution to better utilize the resources provided by modern processor architectures.”

- With z13, SMT allows up to two instructions per core to run simultaneously to get better overall throughput.

- SMT is designed to make better use of processors.

- On z/OS, SMT is available for zIIP processing:
  - Two concurrent threads are available per core and can be turned on or off.
  - Capacity (throughput) usually increases.
  - Performance may in some cases be superior using single threading.

*Wikipedia®

z13 - SMT Exploitation

- Generally focuses on increasing core throughput predictably and repeatability.
- PR/SM supports SMT for SMT aware OS like z/OS via core dispatching.
- z/OS controls and manages whole core (all threads) to:
  - Maximize core throughput (fill running cores, spill to waiting cores).
  - Maximize core availability (meet goals using fewest cores).
- Limits SMT variability to a single z/OS workload.
- Makes capacity, accounting, latency, response time more predictable and repeatable.
What is new with multithreading support?

LOADxx: PROCVIEW CPU
- This the existing mode that you are used to
- z/OS does only know processors (CPUs)

LOADxx: PROCVIEW CORE
- z/OS is core-aware and maps CPUs to cores.
- Core syntax for CPU/core related commands.

MT1 Mode
IEAOPTxx:
MT_ZIIP_MODE = 1
MT2 Mode
IEAOPTxx:
MT_ZIIP_MODE = 2

SET OPT=yy

Possible on any supported hardware
z13 only
MT1 performance & capacity
MT2 performance & cap.
HiperDispatch=No possible
HiperDispatch=Yes enforced

New terminology for SMT…

- z/OS logical processor (CPU) ➔ Thread
  - A thread implements (most of) the System z processor architecture
  - z/OS dispatches work units on threads
  - In MT mode two threads are mapped to a logical core

- Processor core ➔ Core
  - PR/SM dispatches logical core on a physical core
    - Thread density 1 (TD1) when only a single thread runs on a core
    - Thread density 2 (TD2) when both threads run on a core

- MT1 Equivalent Time (MT1ET)
  - z/OS CPU times are normalized to the time it would have taken to run same work in MT-1 mode
    - ASCB, ASSB, ..., SMF30, SMF32, SMF7x, ...
  - You will usually not see the term MT1ET because it is implied

- Several new metrics to describe how efficiently core resources could be utilized…

* Statements regarding IBM future direction and intent are subject to change or withdrawal, and represent goals and objectives only.
...and several new metrics for SMT...

- **WLM/RMF:** Capacity Factor (CF), Maximum Capacity Factor (mCF)
- **RMF:** Average Thread Density, Productivity (PROD)

**How are the new metrics derived?**
- Hardware provides metrics (counters) describing the efficiency of processor (cache use/misses, number instructions when one or two threads were active…)
- LPAR level counters are made available to the OS
- MVS HIS component and supervisor collect LPAR level counters. HIS provides HISMT API to compute average metrics between “previous” HISMT invocation and “now” (current HISMT invocation)
- System components (WLM/SRM, monitors such as RMF) retrieve metrics for management and reporting

* Statements regarding IBM future direction and intent are subject to change or withdrawal, and represent goals and objectives only.

---

**Transitioning into MT2 mode: WLM considerations (1)**

- **Less overflow from zIIP to CPs** may occur because
  - zIIP capacity increases, and
  - number of zIIP CPUs double

- CPU time and CPU service **variability may increase**, because
  - Threads which are running on a core at the same time influence each other
  - Threads may be dispatched at TD1 or TD2

- Sysplex workload routing: routing recommendation may change because
  - zIIP capacity will be adjusted with the mCF to reflect MT2 capacity
  - mCF may change as workload or workload mix changes

* Statements regarding IBM future direction and intent are subject to change or withdrawal, and represent goals and objectives only.
Transitioning into MT2 mode: WLM Considerations (2)

- Goals should be verified for zIIP-intensive work, because
  - The number of zIIP CPUs double and the achieved velocity may change
    - “Chatty” (frequent dispatches) workloads may profit because there is a chance of more timely dispatching
    - More capacity is available
    - Any single thread will effectively run at a reduced speed and the achieved velocity will be lower.
    Affects processor speed bound work, such as single threaded Java batch

* Statements regarding IBM future direction and intent are subject to change or withdrawal, and represent goals and objectives only.

What I hope I covered.....

- What are dispatchable units of work on z/OS
- How WLM manages dispatchable units of work
- The role of HiperDispatch and Warning Track
- Dispatching work to zIIP engines
- z13 and Simultaneous Multithreading (SMT)
IBM Notice Regarding Specialty Engines

Any information contained in this document regarding Specialty Engines ("SEs") and SE eligible workloads provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g., zILPs, zAAPs, and IFLs). IBM authorizes customers to use IBM SEs only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the “Authorized Use Table for IBM Machines” provided at:

No other workload processing is authorized for execution on an SE.

IBM offers SEs at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.