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**

6-Way Processor

- CP 0
- CP 1
- CP 2
- CP 3
- CP 4
- CP 5

- Job A
- Job B
- Job C
- Job E
- Job K
- Job M

Address Space

- In Ready

**WLM policy adjustment algorithm**

- Select Receiver
- Determine Bottleneck
- Fix Bottleneck for selected receiver
  1. Select Donor, one or multiple using the resource
  2. Assess changes on receiver and donors
  3. Adjust access and usage of resources if assessment shows a positive result
- Has a change been made?
- Are there other bottlenecks?
- Are there other receivers?
- End
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 wit algorithm</td>
</tr>
<tr>
<td>192</td>
<td></td>
</tr>
</tbody>
</table>

Dispatching in an LPAR environment

[Diagram showing LPAR1 and LPAR2 connections to Physical Processors and PR/SM Dispatcher.]
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 high share CPs in each node
  - Each node has TCBs and SRBs assigned to the node
  - Optimizes the execution of work units on z/OS
RMF CPU activity report

<table>
<thead>
<tr>
<th>CPU ACTIVITY</th>
<th>CPU ACTIVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
<td><strong>CPU</strong></td>
</tr>
<tr>
<td>Type</td>
<td>Type</td>
</tr>
<tr>
<td>Process</td>
<td>Process</td>
</tr>
<tr>
<td>Online</td>
<td>Online</td>
</tr>
<tr>
<td>LPAR Busy</td>
<td>LPAR Busy</td>
</tr>
<tr>
<td>MVS Busy</td>
<td>MVS Busy</td>
</tr>
<tr>
<td>Pinned</td>
<td>Pinned</td>
</tr>
<tr>
<td>Share %</td>
<td>Share %</td>
</tr>
<tr>
<td>Log Proc</td>
<td>Log Proc</td>
</tr>
<tr>
<td>% VIA INT</td>
<td>% VIA INT</td>
</tr>
</tbody>
</table>

---

**CPU ACTIVITY**

RMF CPU activity report

<table>
<thead>
<tr>
<th>CPU ACTIVITY</th>
<th>CPU ACTIVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
<td><strong>CPU</strong></td>
</tr>
<tr>
<td>Type</td>
<td>Type</td>
</tr>
<tr>
<td>Process</td>
<td>Process</td>
</tr>
<tr>
<td>Online</td>
<td>Online</td>
</tr>
<tr>
<td>LPAR Busy</td>
<td>LPAR Busy</td>
</tr>
<tr>
<td>MVS Busy</td>
<td>MVS Busy</td>
</tr>
<tr>
<td>Pinned</td>
<td>Pinned</td>
</tr>
<tr>
<td>Share %</td>
<td>Share %</td>
</tr>
<tr>
<td>Log Proc</td>
<td>Log Proc</td>
</tr>
<tr>
<td>% VIA INT</td>
<td>% VIA INT</td>
</tr>
</tbody>
</table>

---

**CPU ACTIVITY**

RMF CPU activity report

<table>
<thead>
<tr>
<th>CPU ACTIVITY</th>
<th>CPU ACTIVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
<td><strong>CPU</strong></td>
</tr>
<tr>
<td>Type</td>
<td>Type</td>
</tr>
<tr>
<td>Process</td>
<td>Process</td>
</tr>
<tr>
<td>Online</td>
<td>Online</td>
</tr>
<tr>
<td>LPAR Busy</td>
<td>LPAR Busy</td>
</tr>
<tr>
<td>MVS Busy</td>
<td>MVS Busy</td>
</tr>
<tr>
<td>Pinned</td>
<td>Pinned</td>
</tr>
<tr>
<td>Share %</td>
<td>Share %</td>
</tr>
<tr>
<td>Log Proc</td>
<td>Log Proc</td>
</tr>
<tr>
<td>% VIA INT</td>
<td>% VIA INT</td>
</tr>
</tbody>
</table>

---

**CPU ACTIVITY**

RMF CPU activity report

<table>
<thead>
<tr>
<th>CPU ACTIVITY</th>
<th>CPU ACTIVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
<td><strong>CPU</strong></td>
</tr>
<tr>
<td>Type</td>
<td>Type</td>
</tr>
<tr>
<td>Process</td>
<td>Process</td>
</tr>
<tr>
<td>Online</td>
<td>Online</td>
</tr>
<tr>
<td>LPAR Busy</td>
<td>LPAR Busy</td>
</tr>
<tr>
<td>MVS Busy</td>
<td>MVS Busy</td>
</tr>
<tr>
<td>Pinned</td>
<td>Pinned</td>
</tr>
<tr>
<td>Share %</td>
<td>Share %</td>
</tr>
<tr>
<td>Log Proc</td>
<td>Log Proc</td>
</tr>
<tr>
<td>% VIA INT</td>
<td>% VIA INT</td>
</tr>
</tbody>
</table>

---

**CPU ACTIVITY**

RMF CPU activity report

<table>
<thead>
<tr>
<th>CPU ACTIVITY</th>
<th>CPU ACTIVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
<td><strong>CPU</strong></td>
</tr>
<tr>
<td>Type</td>
<td>Type</td>
</tr>
<tr>
<td>Process</td>
<td>Process</td>
</tr>
<tr>
<td>Online</td>
<td>Online</td>
</tr>
<tr>
<td>LPAR Busy</td>
<td>LPAR Busy</td>
</tr>
<tr>
<td>MVS Busy</td>
<td>MVS Busy</td>
</tr>
<tr>
<td>Pinned</td>
<td>Pinned</td>
</tr>
<tr>
<td>Share %</td>
<td>Share %</td>
</tr>
<tr>
<td>Log Proc</td>
<td>Log Proc</td>
</tr>
<tr>
<td>% VIA INT</td>
<td>% VIA INT</td>
</tr>
</tbody>
</table>

---

**CPU ACTIVITY**

RMF CPU activity report

<table>
<thead>
<tr>
<th>CPU ACTIVITY</th>
<th>CPU ACTIVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
<td><strong>CPU</strong></td>
</tr>
<tr>
<td>Type</td>
<td>Type</td>
</tr>
<tr>
<td>Process</td>
<td>Process</td>
</tr>
<tr>
<td>Online</td>
<td>Online</td>
</tr>
<tr>
<td>LPAR Busy</td>
<td>LPAR Busy</td>
</tr>
<tr>
<td>MVS Busy</td>
<td>MVS Busy</td>
</tr>
<tr>
<td>Pinned</td>
<td>Pinned</td>
</tr>
<tr>
<td>Share %</td>
<td>Share %</td>
</tr>
<tr>
<td>Log Proc</td>
<td>Log Proc</td>
</tr>
<tr>
<td>% VIA INT</td>
<td>% VIA INT</td>
</tr>
</tbody>
</table>

---

**CPU ACTIVITY**
HiperDispatch and LPAR

### PARTITION DATA REPORT

- **z/OS Version**: z/OS V1R10
- **System ID**: LPAR1
- **Date**: 04/29/2011
- **Interval**: 14.59.998
- **Time**: 19.28.00
- **Cycle**: 1,100 seconds

#### MVS Partition Name: LPAR1
- **Number of Physical Processors**: 55
- **Group Name**: N/A
- **Image Capacity**: 3165 CP
- **Limit**: N/A
- **Number of Configured Partitions**: 4
- **Available**: N/A
- **Wait Completion**: No
- **Dispatch Interval**: Dynamic

#### Partition Data

<table>
<thead>
<tr>
<th>Name</th>
<th>S</th>
<th>WGT</th>
<th>Def</th>
<th>Act</th>
<th>Type</th>
<th>Effective</th>
<th>Total</th>
<th>Effective</th>
<th>Total</th>
<th>LPAR MGMT</th>
<th>Effective</th>
<th>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>
</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>
</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>0.0</td>
<td>CP</td>
<td>00.00.00.000</td>
<td>00.00.00.000</td>
<td>0.00</td>
<td>0.00</td>
</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>0.0</td>
<td>CP</td>
<td>00.00.00.000</td>
<td>00.00.00.000</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**Physical**

- **Effective**: 00.10.58.833
- **Total**: 05.18.52.927
- **Effective Total**: 2.21
- **Total**: 41.68
- **LPAR1**: 17.96
- **LPAR2**: 24.06

#### Total LPAR weight = 1000
- **LPAR1**: 494/1000 = .494 * 53 CPs = 26.18 CPs
- **LPAR2**: 446/1000 = .446 * 53 CPs = 23.64 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

53 physicals

---

**Dispatching in an LPAR environment**

[Diagram showing LPAR1 and LPAR2, with work units, logical processors, z/OS dispatcher, and PR/SM dispatcher]
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 undispatch a certain logical processor within the next 50 microseconds (grace period).

z/OS 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
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>Offset</th>
<th>Name</th>
<th>Length</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>SMF70STS</td>
<td>4</td>
<td>Binary</td>
<td>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.</td>
</tr>
<tr>
<td>84</td>
<td>SMF70STU</td>
<td>4</td>
<td>Binary</td>
<td>The number of times PR/SM issued a warning-track interruption to a logical processor and z/OS was unable to return the logical processor within the grace period.</td>
</tr>
<tr>
<td>88</td>
<td>SMF70STI</td>
<td>4</td>
<td>Binary</td>
<td>Amount of time in microseconds that a logical processor was yielded to PR/SM due to Warning Track processing.</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 (WTRKAP) (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/unable to return it to PR/SM within the grace period.</td>
</tr>
<tr>
<td>WTRKCTP (WTRKTAAP) (WTRKTP)</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/390):

Sample output
### 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>
<tbody>
<tr>
<td>zAAP or zIIP on z13</td>
<td>Any Java Execution</td>
<td>Websphere, CICS, Native apps, XMLSS</td>
</tr>
<tr>
<td>zIIP</td>
<td>Enclave SRBs</td>
<td>DRDA over TCPIP, DB2 Parallel Query, DB2 Utilities Load, Reorg, Rebuild, DB2 V9 z/OS remote native SQL procedures, TCPIP - 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, SDSF (V2.2)</td>
</tr>
</tbody>
</table>

Work-Dependent enclaves

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)
**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.

**DDS and work-dependent enclaves**

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 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.

*Note: Speed limit signs for illustration only*  

---

**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.
Several new metrics for SMT...

New metrics:

- 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.

z13 – z/OS SMT Metrics

- Capacity Factor (CF)
  - How much work core actually completes for a given workload mix at current utilization - relative to single thread
  - MT-1 Capacity Factor is 1.0 (100%)
  - MT-2 Capacity Factor is workload dependent
- Maximum Capacity Factor (mCF)
  - How much work a core can complete for a given workload mix at most
- Core Busy Time
  - Time any thread on the core is executing instructions when core is dispatched to physical core
- Average Thread Density
  - Average number of executing threads during Core Busy Time (Range: 1.0 - 2.0)
- Productivity
  - Core Busy Time Utilization (percentage of used capacity) for a given workload mix
  - Productivity represents capacity in use (CF) relative to capacity total (mCF) during Core Busy Time.
- Core Utilization
  - Capacity in use relative to capacity total over some time interval
  - Calculated as Core Busy Time x Productivity
z13 – SMT: Postprocessor CPU Activity Report

- PP CPU activity report provides new metrics when SMT is active
  - MT Productivity and Utilization of each logical core
  - MT Multi-Threading Analysis section displays MT Mode, MT Capacity Factors and average Thread Density
- One data line in PP CPU activity report represents one thread (CPU)
  - CPU NUM designates the logical core
- Some metrics like TIME % ONLINE and LPAR BUSY provided at core granularity only

<table>
<thead>
<tr>
<th>NUM</th>
<th>TYPE</th>
<th>ONLINE</th>
<th>LPAR BUSY</th>
<th>MVS BUSY</th>
<th>PARKED</th>
<th>PROD</th>
<th>UTIL</th>
<th>SHARE %</th>
<th>RATE</th>
<th>% VIA TPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CP</td>
<td>100.00</td>
<td>68.07</td>
<td>67.94</td>
<td>0.00</td>
<td>100.00</td>
<td>68.07</td>
<td>100.00</td>
<td>370.1</td>
<td>13.90</td>
</tr>
<tr>
<td>1</td>
<td>CP</td>
<td>100.00</td>
<td>46.78</td>
<td>46.78</td>
<td>0.00</td>
<td>100.00</td>
<td>46.78</td>
<td>52.9</td>
<td>5.29</td>
<td>16.93</td>
</tr>
</tbody>
</table>

TOTAL/AVERAGE

A   IIP 100.00 48.15 41.70 32.81 85.84 41.33 100.0 35.66 0.00
B   IIP 100.00 38.50 32.81 26.47 85.94 33.09 100.0 26.47 0.00

TOTAL/AVERAGE

29.48 23.23 152.9 375.3 13.95

MULTI-THREADING ANALYSIS

- CPU TYPE: CP
  - CPU NUM: 0
  - CPU TYPE: IIP
  - CPU NUM: 1

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 Simultaneous Multithreading (SMT)