Hello. Welcome to the second volume of VM performance case studies. If you missed the first volume you can find it at http://www.vm.ibm.com/devpages/bitner/presentations/case99.pdf. This is a collection of real situations that I worked on in the past year or so. They illustrate various performance techniques and tools.
I will show various examples of reports and data in this presentation. Many of the reports have been slightly edited to allow them to fit on the page and to highlight the important information.
I added this chart to try to simplify a discussion of the tools available to the VM performance analyst. As you can see there are a number of tools available from IBM and vendors. Most rely on the architected monitor data, but others use diagnose x'04' to view CP control blocks for additional data.

Different products are used for different purposes: real time monitoring, history and trend analysis, or statistical analysis.
Case 1: The Case of Crowded Storage

I got an e-mail from Erich Amrehn about a customer...

I have a customer question about CMS performance looks like the 16MB (line) is a problem for him and he is looking for some help to identify the problem and possible ways to fix it. Can you help??

Erich and I exchanged a couple notes with my last response being...

I would recommend looking at the CMS Storage Utilities: STORMAP, SUBPMAP, and STDEBUG. I'm not familiar with stairs, is it a long running application? Server like? If so, they might want to use the EXTSET option to allow the utilities to collect data while the program is running.

This first case is interesting. Erich Amrehn, who currently works at the ITSO in Poughkeepsie, was contacted by a customer that knew him from his work with IBM in Germany. Erich asked if I could help. You see Erich's request and my reply. After my last response I expected to have a series of notes and data being sent back and forth.
Case 1: No News is Good News?

- No response for a month.
- Then a thank you note from the customer.
- Problem solved through Storage Utilities and web page hints.
- The LE segment and Pipelines segments somehow were overlaying each other.
- LE would try to load segment, but could not.
- LE then proceeded to load run time below 16MB storage. (LE is not small)

- After a month, I still had not heard from the customer. Then I got a thank you note. (I like getting them). The odd thing was I wasn't sure at first why they were thanking me.
- It turns out they were able to solve the problem through the tools I pointed them to and the information on the performance web page (http://www.ibm.com/s390/vm/perf/tips/).
- There problem was that the segment containing the LE code and the Pipelines segments were somehow overlaying each other. LE would try to load the segment, but could not because Pipelines was already loaded. When this occurs LE will attempt to load the run time library below the 16MB line. Unfortunately, LE does not provide an error or informational message to indicate that this is happening. LE is not small and therefore caused a large decrease in available virtual...
Case 1: Conclusion

- Bit is not really needed any more, just check the VM Home Page.
- The Storage Utilities can be helpful:
  - STORMAP - map out storage in CMS
  - SUBPMAP - map out subpool storage
  - STDEBUG - track storage obtains/releases
- It is worth double checking segments, especially when LE is involved.

- This was an example of where we have tried to provide useful information to customers and other IBMers off our home page. Often your problem and solution has already been seen. Our VM support teams use these pages as well.
- If you run into virtual storage problems, I highly recommend looking at the Storage Utilities for help. They are very powerful. It is also worth your time to double check segments for overlays, particularly after upgrades or migrations to newer releases.
Case 2: The Case of the Rotten RSU

- Customer concern: After applying RSU 9904 to VM/ESA 2.3.0 response time is worse.
- Integrated Server (P390) 256MB/64MB
- Development house:
  - various 2nd level systems, VM and MVS
  - 1st level CMS work, batch, SQL/DS  DB/2
  - TCP/IP
- Discussions narrowed it down to almost everyone is affected
- Sent in monitor data

- We had a call appear on the queue from a customer who saw performance degrade sharply after applying RSU 9904. I gave the customer a call and found he was running a development house workload with several guests on an Integrated Server. I asked the typical questions about who was seeing the problem and what exactly was meant by "performance" problem. It came down to just about everyone was seeing response time that was at least an order of magnitude worse than it had been prior to the RSU. I was not aware of any APARs that would be likely problems on that RSU. The systems programmer was a contractor and fairly new to this system. I knew the Integrated Server could emulate devices, but he was not familiar enough with the system to know if they were using a lot of emulated devices. I was interested in seeing monitor data.
I walked the customer through collecting monitor data and he FTP the data to us. Reducing the monitor data with VMPRF, I looked at the PRF084 Device Configuration report. As you can see there are various DASD types defined. After describing what the real boxes would look like, it became clear that many of these were emulated. You'll also see that there is only a single channel for each device.
Looking at the VMPRF PRF002 System Summary report, we see a few more interesting pieces of information. The processor utilization is fairly low, with a max shown here of 54.1%. You also see DASD Resp Time is 0. That's because the subchannel measurement timing values are not valid for emulated I/O on the Integrated Server. Make a note of the Active User Count of 19 or 20. We will see how that is important later.
Case 2: DASD I/O?

<table>
<thead>
<tr>
<th>Case</th>
<th>PRF016 Run 12/08/1999 14:38:11</th>
<th>DASD Activity Ordered by Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>12/02/1999 10:04:36</td>
<td>VMPRF 1.2.1</td>
</tr>
<tr>
<td>To</td>
<td>12/02/1999 17:01:36</td>
<td>CPU 7490 SN 2086</td>
</tr>
<tr>
<td>For</td>
<td>25020 Secs 06:56:59</td>
<td>Bill Bitner Analysis</td>
</tr>
<tr>
<td></td>
<td>VM/ESA 2.3.0 SLU 990</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Device</th>
<th>Volume</th>
<th>Control</th>
<th>Size</th>
<th>Count</th>
<th>Rate</th>
<th>Busy</th>
<th>Pend</th>
<th>Disc</th>
<th>Conn</th>
<th>Serv</th>
<th>Resp</th>
<th>Read</th>
<th>Hit</th>
<th>Miss</th>
</tr>
</thead>
<tbody>
<tr>
<td>012C</td>
<td>SDSMV4</td>
<td>3380-K</td>
<td>3880-23</td>
<td>0MB</td>
<td>84184</td>
<td>3.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0822</td>
<td>SCPMV5</td>
<td>3380-E</td>
<td>3880-23</td>
<td>0MB</td>
<td>55190</td>
<td>2.2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>062D</td>
<td>V81001</td>
<td>3380-E</td>
<td>3880-23</td>
<td>0MB</td>
<td>18111</td>
<td>0.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>062A</td>
<td>230CP0</td>
<td>3380-K</td>
<td>3880-23</td>
<td>0MB</td>
<td>17610</td>
<td>0.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.9</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0413</td>
<td>E22W02</td>
<td>3380-J</td>
<td>3880-23</td>
<td>0MB</td>
<td>14206</td>
<td>0.6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>065A</td>
<td>BLS35A</td>
<td>3380-J</td>
<td>3880-23</td>
<td>0MB</td>
<td>11731</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.9</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0627</td>
<td>BLS627</td>
<td>3380-E</td>
<td>3880-23</td>
<td>0MB</td>
<td>11600</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- While we can not see the timings for I/Os, we can get the I/O rate for the DASD by looking at VMPRF PRF016 Cache Dasd by Activity report. You see here that the I/O rates are fairly low. You will also notice that while cache control units are emulated, the cache statistics are not. Therefore, the counters associated with cache efficiency are all zero.
- In any case, the I/O rates are low enough and we do not see queuing on the devices, that we probably need to look else where for the problem.
The PRF072 System Configuration report is one I am learning to pay more attention. This report describes a lot of the tuning parameters. As I go down the list, I do not see anything too strange until I get to the DSPBUF settings. Few people turn this tuning knob. It controls how many users of the different transaction classes are allowed to run (allowed into the dispatch list). Since most of the guests are second level or service machines, Q2 and Q3 values are of particular interest. I stopped at this point and called the customer back.
Case 2: Conclusion

- Not clear who changed various scheduler settings.
- Went back to defaults.
- Things seem much better.
- The third value for DSPBUF was limiting only 10 users to be dispatchable at one time.

SET SRM DSPBUF:
Turn with Care!!

It was not clear who had changed this setting (the default is over 32000 for each class). However, after setting it back to the default the system ran much better. Please be careful if you turn this knob. I have never seen it used effectively, except in very processor constrained environments.
Case 3: The case of a Needle in the Haystack

- Customer with complex processing involving
  - IBM Products (Callup)
  - Other Vendor Products
  - SFS, Spool
  - Nightly processing
- 9672-R86 partition with 5 logical processors
- 2GB/4GB
- User data transformation
  - using SFS
  - using Callup Product for directory services

- This next problem involved a complex system with IBM products, vendor code, and customer applications. Basically, it involved a process that transformed data in various formats. A large amount of processing was performed each night. From a hardware perspective, a lot of resources were available. The Shared File System (SFS) was involved in holding some of the data. Also a product for directory services from IBM was being used. I have always called the product Callup, but I believe the official name is CDS.
Case 3: SFS still on my mind

After getting monitor data, I first wanted to look at SFS for no other reason than the last problem I had looked at was SFS related. It was still fresh in my mind. This VMPRF SFS by Time report shows that things look good from an SFS perspective. The time per file pool request is under 1 millisecond. The server utilization is all processor time with not page read, checkpoint, or QSAM delays. The active and held agent rates are low, indicating nice short units of work. Also, note that there are no rollbacks due to deadlocks.

- We need to look else where.
Looking at the User Resource Utilization report I see one user, named SILLY, standing out. It consumes a huge amount of processor time particularly in relation to the DASD I/O rate in the far right column. You see the SFS machine and other machines (CHANGE1, TRANSFOR) that are involved in the transformation process and that they do not use near the amount of resources that SILLY does.
Case 3: Why so much Silly User?

- Processing logs showed approximately 283 transformations during that time frame
  - 10834 seconds virtual CPU
  - 60 seconds CP CPU
  - \(\frac{10834}{283} = 38.3\) seconds CPU
  - Roughly 1.9 Billion instructions per transaction!
    - Little CP activity
    - Little activity from the other users
- Something wrong in the SILLY user!

- The customer was able to provide me with logs from their transformation processing that showed me the rate of work. Using data from the logs, I was able to compute how much processor time and roughly how many instructions were involved in a single transaction on the average.
- Something was definitely wrong with the SILLY user and I needed to talk to the customer to understand what.
Case 3: Tracking Silly User

- Silly user runs large Execs
- TRACEXEC tool (from Kent Fiala of SAS)
  - See workshop tool tape pages
- A few CP Q TIME inserted in the key exec
- Narrowed it down to a routine named CALLDBI which is interface to Callup to get directory record layouts.
  - Note: we could have used the profiling capabilities of TRACEXEC to narrow down the problem to this level.

► It turned out the SILLY ran some rather large REXX execs that involved directory lookups. Using a tool from Kent Fiala called TRACEXEC and a few strategically placed CP Q TIME commands, we were able to narrow it down to a routine named CALDBI which is an interface to the Callup product to get the directory record layout. Now we were making some progress.
Case 3: Recreate the Crime

- Initial attempts to recreate unsuccessful
- Customer found that CALLDBI seemed sensitive to the number of Rexx variables that exist when you invoke CALLDBI
- At this point I was able to recreate
  - TRACEXEC
  - STARS (System Trace Analysis Reports) - currently internal use only

Since we use Callup internally, I thought it would be easy to recreate the problem on my own system. However, I was unable to recreate the problem until the customer noticed that CALLDBI performance was sensitive to the number of Rexx variables that existed. At that point, I used TRACEXEC and an internal tool named STARS (System Trace Analysis Reports) to dig further.
Case 3: RXCALLV is Ugly

<table>
<thead>
<tr>
<th>Stem Variables</th>
<th>1000</th>
<th>10000</th>
<th>100000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front matter</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>calldr</td>
<td>0.07</td>
<td>0.06</td>
<td>0.06</td>
</tr>
<tr>
<td>middle 1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>rxcallv get</td>
<td>0.06</td>
<td>0.56</td>
<td>5.63</td>
</tr>
<tr>
<td>middle 2</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>rxcallv set</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>globalv</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>parse select</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>about to exit</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
</tr>
</tbody>
</table>

RXCALLV GET basically passes REXX variables between routines.

- Using a test program where I could vary the number of stem variables defined, I measured several different cases of CALLDBI and saw that a routine named RXCALLV varied a great deal with the number of variables when called with the GET option. This routine basically passes Rexx variables between routines.
Case 3: STARS narrows in

<table>
<thead>
<tr>
<th></th>
<th>1 Var</th>
<th>10 Vars</th>
<th>100 Vars</th>
<th>1000 Vars</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL</td>
<td>1.00</td>
<td>1.29</td>
<td>4.23</td>
<td>33.55</td>
</tr>
<tr>
<td>DMSITS</td>
<td>1.00</td>
<td>1.33</td>
<td>4.64</td>
<td>37.69</td>
</tr>
<tr>
<td>RXCALLV</td>
<td>1.00</td>
<td>1.35</td>
<td>4.90</td>
<td>40.37</td>
</tr>
<tr>
<td>DMSITSX</td>
<td>1.00</td>
<td>1.43</td>
<td>5.70</td>
<td>48.45</td>
</tr>
<tr>
<td>DMSREX</td>
<td>1.00</td>
<td>1.41</td>
<td>5.48</td>
<td>46.22</td>
</tr>
<tr>
<td>IXXRVA</td>
<td>1.00</td>
<td>1.39</td>
<td>5.24</td>
<td>43.80</td>
</tr>
<tr>
<td>DMSFRG</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>DMSFRR</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>DMSGLU</td>
<td>1.00</td>
<td>1.36</td>
<td>4.96</td>
<td>40.96</td>
</tr>
<tr>
<td>DMSFRE</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Number of instructions normalized to 1 variable

- The STARS tool shows me the code involved in this large systems effect. When we consider that the customer exec would read entire directories (10s of thousands of entries) into stem variables, you can see the potential for problems and why it may not have been noticeable with smaller test cases.
Case 3: Conclusion

- Short Term: in transformation processing, only use CALLDBI to get the directory layout once when processing starts instead of for each transformation.
- Long Term: Bit needs to come up with alternative for RXCALLV and get the Callup Product owners to accept it.

- The problem was further exasperated by using CALLDBI for each transaction. A simple change was to check the record layout once with CALLDBI when processing starts instead of for each transaction.
- A long term solution would be for the Callup product to find a better way of getting access for the Rexx variables. I have not made time to pursue this much at this point.
Case 4: The Case of "You think you got enough storage there?"

- 9672-Z17 G6 Turbo
- 1994MB cstore / 29GB xstore
- 1150 CMS Users
- A few server machines
- Wanted to know why they saw paging activity with so much storage?
  - 18 dedicate page volumes
  - Paging not on any of them

- You might be saying, "I have should have such a problem." This was actually an internal test system trying to get some high end measurements with a new CMS interactive workload. Their question was "why do I see paging activity when I have so much storage?". And further, why is the paging not to paging space?
- I asked them to send me monitor data and we could take a look.
Case 4: Paging Reports

PRF088 Run 02/01/2000 13:44:02
DASD_SYSTEM.Areas
DASD System Areas by Type: Paging and Spooling Activity
From 02/01/2000 13:26:26       VMPRF 1.2.1
To    02/01/2000 13:40:26
For    840 Secs 00:14:00       9672 G6 Turbo CMS Run

 benefici  <---------Slots--------> <----------------Rate-------------->
Num- Volume         Avail-  Page Spool  Page  Page Spool Spool         SSCH   Time   Used
ber Serial   Type    able InUse InUse  Read Write  Read Write  Total +RSCH  /Page  Alloca
0E0A LSP3VM   Spool  40680     0  15.1 190.9     0  22.2  19.0  232.1  53.3      0     100
F5C0 SPOL01   Spool 400500     0   0.1     0     0   2.9   3.0    5.9   5.9    1.1      25
F5C1 SPOL02   Spool 400500     0   4.3     0     0   5.6   5.5  11.1  11.1    1.2      45
F5C2 SPOL03   Spool 400500     0   0.0     0     0   1.7   1.9    3.6   3.7    1.0      20
Sum/Mean      Spool 310545     0   1.9  47.7     0  8.1  7.3  63.1  18.5    0.8      47
0E08 LSP1VM   -PgSp  35820   1.5  17.7 321.5   8.4  28.6 13.5  372.1  81.4    0.4      60
Sum/Mean      -PgSp  35820   1.5  17.7 321.5   8.4  28.6 13.5  372.1  81.4    0.4      60

Why page to spool areas?

The VMPRF DASD System Areas report shows the page and spool space and the activity to these volumes. One less than optimal item is that there is a volume with mixed page and spool space. This probably is not a major problem when you have 29GB of xstore, but I mention it for completeness. What else is interesting is the the "paging activity" to the LSP3VM volume which is a spool only volume. Also note that it is only for reads, not writes.

Any ideas why we might page to a spool area?
Case 4: Which Segments?

- We page the initial read of a segment in from spool.

```plaintext
<table>
<thead>
<tr>
<th>Name of Seg</th>
<th>NSS or DCSS</th>
<th>Spool Number</th>
<th>Creation-Date</th>
<th>NSS or DCSS</th>
<th>Spool Number</th>
<th>Creation-Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMSINST</td>
<td>DCSS</td>
<td>127</td>
<td>09/02/1999 14:08:14</td>
<td>1153</td>
<td>09/02/1999 14:08:14</td>
<td></td>
</tr>
<tr>
<td>CMSPIPES</td>
<td>DCSS</td>
<td>120</td>
<td>09/02/1999 13:02:40</td>
<td>1151</td>
<td>09/02/1999 13:02:40</td>
<td></td>
</tr>
<tr>
<td>CMSVMLIB</td>
<td>DCSS</td>
<td>121</td>
<td>09/02/1999 13:18:38</td>
<td>1150</td>
<td>09/02/1999 13:18:38</td>
<td></td>
</tr>
<tr>
<td>FORTRAN</td>
<td>DCSS</td>
<td>100</td>
<td>12/18/1991 09:00:37</td>
<td>0120</td>
<td>12/18/1991 09:00:37</td>
<td></td>
</tr>
<tr>
<td>GCSXA</td>
<td>DCSS</td>
<td>123</td>
<td>09/29/1999 09:03:04</td>
<td>1173</td>
<td>09/29/1999 09:03:04</td>
<td></td>
</tr>
<tr>
<td>GOODHELP</td>
<td>DCSS</td>
<td>125</td>
<td>09/29/1999 09:54:09</td>
<td>1293</td>
<td>09/29/1999 09:54:09</td>
<td></td>
</tr>
<tr>
<td>MONDCSS</td>
<td>DCSS</td>
<td>95</td>
<td>01/18/1999 12:18:06</td>
<td>1280</td>
<td>01/18/1999 12:18:06</td>
<td></td>
</tr>
<tr>
<td>VTAMXA</td>
<td>DCSS</td>
<td>97</td>
<td>08/31/1993 08:34:01</td>
<td>128</td>
<td>08/31/1993 08:34:01</td>
<td></td>
</tr>
</tbody>
</table>
```

Difficult to tell due to packed and logical segments.

- Some of you probably guessed that it was due to segment activity. And you would be right. The next step is to determine which segments. After the first user connects to a segment, we read it in and then page it in and out to paging areas until, as users drop the segment, there is no one connected to it. CP then releases the structures associated with the segment. The next time a user loads it, we again would read it from the spool area. There are a couple of candidates from this VMPRF report that could be getting loaded and dropped a lot. There are also some segments we can rule out, such as CMS, CMSINST, etc. that have a high Users count.
Case 4: The answer is in there...

- **MONITOR LIST1403** describes layout for monitor records.
- **Monitor Domain 3 Record 16 (MRSTOSHD):** 
  - NSS/DCSS/SSP Removed From Storage
  - Storage Domain Event Record
  - Saved Segment Name at offset 20 for 8
  - Spool id at offset 40 for 2
- **Read raw monitor data with:**
  - Using Monview (from samples disk)
  - Pipelines STARMON stage

- The answer is currently hidden in the monitor data. If we look at the **MONITOR LIST1403** file, we can get the record layouts for monitor records and see that Domain 3 Record 16 is created when a segment is removed from storage (last user connected to segment, releases it). From that record, we can get the segment name and spool id at offsets 20 and 40.
- To view the monitor records, we can either process existing monitor data files from MONWRITE with the Monview tool (on the VM samples disk) or with the Pipelines STARMON stage.
### Case 4: Getting the data out...

```c
/* Example with Pipelines Starmon to get D3/R16 spool name and spid */
'Segment Load MONDCSS'
'PIPE (endchar ?) STARMON MONDCSS SHARE',
'| locate 5 x03 | locate 8 x10',
'| spec 21.8 2 41.2 c2x 11',
'| a: fanout ',
'| sort count',
'| cons',
'|? a: | drop 100 | pipestop'
```

**MONVIEW** raw_monitor out_file (DR 3 16
Pipe < out_file | monvu t20.8 40.2 (nohdr | sort count | cons

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>632</td>
<td>ASMAPSEG</td>
<td>0065</td>
</tr>
<tr>
<td>834</td>
<td>FORTRAN</td>
<td>0064</td>
</tr>
</tbody>
</table>

- Here you see examples of how to use Pipelines or the MONVIEW tool. In the Pipelines case, we do a pair of locates to get Domain 3 and Record 16. The spec stage gets the two fields we are interested in (note that offsets here are plus one since it is a byte count). The fanout to a pipestop is just my crude way of stopping after I get 100 records.
- The second box shows how to do this with MONVIEW and friends. Here the offsets are just offsets. The sort count stages get the count of each segment dropped.
- In our case, the two big segments were the Assembler and the Fortran segments.
Case 4: Conclusion

- Preload the Assembler and Fortran segments.
- Get rid of the mixed page/spool volume.
- Get a bigger workload to use that storage.

► Now what? Well, if we know the segments that get used a lot, but only for short periods of time, we can get an improvement by preloading them. Have an autologged user load the segments and just stay there disconnected.
► I also recommended that we remove the mixed page and spool situation.
► I am looking forward to increasing the workload to use all of that storage.
Case 5: The Case of Tired Iron

- Eleventh hour type of migration for Year 2000.
- HPO5 workload running second level on VM/ESA 2.3.0.
- Migrate HPO5 system to VM/ESA, again second level.
  - V=R Guest
  - CMS level stays at CMS 5.
- Performance becomes horrible!

I did get involved in a few last minute migrations in 1999 where people must have forgot about the Year 2000 thing. This example was one of them. A customer had an HPO 5 system running second level on VM/ESA 2.3.0. They migrated the HPO 5 system to VM/ESA and continued to run this as a V=R guest of the VM/ESA 2.3.0 system. CMS was kept at Level 5. Performance was horrible. I was asked to get involved.
Case 5: E before S before J

- Customer mentions it is a 3090
- VM/ESA uses SIE to dispatch guests
- HPO uses LPSW to dispatch guests
- On new configuration we have SIE on top of SIE
- To run efficiently it needs Interpreted SIE assist.
- What model of 3090?

Speaking to the customer on the phone, I hear that they are running on a 3090 (I do not hear that very often anymore). That ends up being important. While CMS had stayed the same, there are key differences in CP between HPO 5 and VM/ESA 2.3.0. In particular, VM/ESA uses SIE (start interpretive execution) to run guests, while HPO used LPSW. If you run VM/ESA on VM/ESA, as was done here, you have two levels of SIE. To avoid virtualization of SIE you need hardware assists. This stretched my memory as to when those assists had been made available.
Case 5: If we had RTM

From D GENERAL or D USER or D ULOG:

<table>
<thead>
<tr>
<th>USERID</th>
<th>%CPU</th>
<th>%CP</th>
<th>%EM</th>
<th>ISEC</th>
<th>PAG</th>
<th>WSS</th>
<th>RES</th>
<th>UR</th>
<th>PGES</th>
<th>SHARE</th>
<th>VMSIZE</th>
<th>TYP,CHR,STAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BITMAN</td>
<td>98</td>
<td>34</td>
<td>64</td>
<td>0.00</td>
<td>0.00</td>
<td>9995</td>
<td>8198</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>256M</td>
<td>VUS,VSI,SIMW</td>
</tr>
<tr>
<td>KARLAC</td>
<td>40</td>
<td>2.0</td>
<td>38</td>
<td>27</td>
<td>0.00</td>
<td>1824</td>
<td>1829</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>64M</td>
<td>VUX,---,IDLE</td>
</tr>
<tr>
<td>HOLDER</td>
<td>2.0</td>
<td>.28</td>
<td>1.7</td>
<td>15</td>
<td>0.00</td>
<td>2814</td>
<td>2814</td>
<td>0</td>
<td>523</td>
<td>100</td>
<td>64M</td>
<td>VUC,IAB,IDLE</td>
</tr>
</tbody>
</table>

From D PRIVOPS (last page):

<table>
<thead>
<tr>
<th>CNTRNAME</th>
<th>INTLVCNT</th>
<th>NSEC</th>
<th>TOTALCNT</th>
<th>NSEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>STOSM</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TB</td>
<td>0</td>
<td>0</td>
<td>133037</td>
<td>7</td>
</tr>
<tr>
<td>V/SIE</td>
<td>82521</td>
<td>2750</td>
<td>2.50E6</td>
<td>145</td>
</tr>
</tbody>
</table>

I did not have access to RTM for this customer. But here are some things to look for if you do. The user screen will have "VSI" under the characteristics field if Virtual SIE is currently being used. More telling, would be the V/SIE count from the last page of the Privops display which gives the rate of virtualization of SIE instructions.
It turns out this processor was an "E", which did not have the assist. Only the 3090Js have the assist. I believe you can still get the assist via an RPQ for "S" models. However, it would be cheaper to just get a new processor. We recommended that they move this work to the first level VM/ESA and things ran much better. I should note that all current 9672s support the assist.

That's it for this volume of case studies. Stay tuned for Volume 3.