Session 9300

Linux Performance Update

Jens Osterkamp (Jens.Osterkamp@de.ibm.com)
IBM Lab Boeblingen

SHARE, February 22-27, 2004 | Longbeach, CA
Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.
Enterprise Storage Server
ESCON*
FICON
FICON Express
HiperSockets
IBM*
IBM logo*
IBM eServer
Netfinity*
S/390*
VM/ESA*
WebSphere*
z/VM
zSeries
* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.
Intel is a trademark of the Intel Corporation in the United States and other countries.
Java and all Java-related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries.
Lotus, Notes, and Domino are trademarks or registered trademarks of Lotus Development Corporation.
Linux is a registered trademark of Linus Torvalds.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
Penguin (Tux) compliments of Larry Ewing.
SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.
UNIX is a registered trademark of The Open Group in the United States and other countries.
* All other products may be trademarks or registered trademarks of their respective companies.
Agenda

- Relative System Capacity
- zSeries Hardware
- Scalability
- Networking
- Disk I/O
  - Parallel Access Volume (PAV)
  - ESS Architecture
Relative System Capacity

- A system provides different types of resources
- Capacity for each resource type may be different
- The ideal machine provides enough capacity of each type
- Don't forget additional Resources (Network, Skilled staff, Money, availability of software, reliability, time ...)

The ideal platform requires a mix of resources in right quantity
Resource Profiles

Each application has its specific requirements
- CPU intensive
- I/O intensive
- Memory

Applications can often be tuned to change the resource profile
- Exchange one resource for the other
- Requires knowledge about available resources

Some platforms can be extended better than others
- Not every platform runs every application well
- It's not easy to determine the resource profile of an appl.
zSeries Hardware

z800/z900

z990
z900 System structure:
Optimized for maximum external bandwidth

- 20 PU Chips @ 1.3 / 1.09 ns
- 3 SAP's, 1 spare
- up to 16 CP's
- up to 8 ICF's/IFL's
z990: Extended Multi-Node(Book)-Structures:

From z900 ...

To z990:
- 0.83ns CPU-Cycle
- Superscalar Design
- Up to 60% more UP-Performance vs 2C1
A single pool of physical resources (CPU's, memory, I/O) in modular implementation (n=1/2/3/4 nodes/'books')

Multiple Channel Subsystems (n x 256 CHPIDs)

Exploitation through virtual servers: 15, 30, 60 (SOD) LPARs ...100+... (VM)
IBM S390 and zSeries Servers – Balanced Scaling

* External I/O or STI bandwidth only (Internal Coupling Channels and HiperSockets not included)
zSeries MCM internal bandwidth is 500 GB/s. Memory bandwidth not included (not a system constraint)
Performance results
Our Hardware for Measurements

2064-216 (z900)
1.09ns (917MHz)
2 * 16 MB L2 Cache (shared)
64 GB
FICON
HiperSockets
OSA Express GbE
z/VM 4.3

2105-F20 (Shark)
384 MB NVS
16 GB Cache
128 * 36 GB disks
10.000 RPM
FCP (2 Gbps)
FICON (1 Gbps)

2084-B16 (z990)
0.83ns (1.2 GHz)
2 Books each with 8 CPUs
64 GB
FICON
HiperSockets
OSA Express GbE
z/VM 4.3

8687-3RX (8-way X440)
8-way Intel Pentium 3 Xeon
1.6 GHz
8 * 512K L2 Cache (private)
hyperthreading
summit chipset
SuSE SLES7 versus SuSE SLES8

- From Kernel version 2.4.7 / 2.4.17 to version 2.4.19
- From glibc version 2.2.4-31 to version 2.2.5-84
- From gcc version 2.95.3 to version 3.2-31
- Huge number of United Linux patches
- 1.3 MLOC (including x,p,i changes)
- New Linux scheduler
- Async I/O
- SLES8 SP2 available
Scalability - z900 vs z990, ext2, 31 Bit

- z990 takes advantage of higher memory bandwidth
Scalability - z990 vs Intel, ext2, 31/32Bit

- z990 shows good scaling behavior
- x440 shows best throughput with 4 CPU, strong throughput degradation with more than 4 CPUs
Kernel – Context Switches

- Context Switches much faster on zSeries because of large shared caches
Networking

- IBM internal benchmark Netmark 2
- Available as “IBM Application Workload Modeler”
- Simulates network traffic
- Adjustable parameters
  - runtime
  - packet size
  - number of connections
  - ...
- Huge results file with much statistical information
- Numbers measured on z900 and z990
HiperSockets MTU 32K – LPAR

Stream workload

Transactions per sec

Throughput in MB/sec

CRR workload

Transactions per sec
GuestLAN type HiperSockets MTU 32K – z/VM guests

RR 200x32k workload

# of connections

Transaction per sec

CPU load (q time) RR 200x32k workload

1cl = 1 connection client side (sv=server)
Gigabit Ethernet MTU 1500 – z/VM guests

Stream workload

Throughput in MB/sec

<table>
<thead>
<tr>
<th>strg_1</th>
<th>strg_10</th>
<th>strg_50</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>70</td>
<td>80</td>
</tr>
</tbody>
</table>

RR workload

Transactions per sec

<table>
<thead>
<tr>
<th>1</th>
<th>10</th>
<th>50</th>
<th>RR 1/1</th>
<th>1</th>
<th>10</th>
<th>50</th>
<th>RR 200/1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>2500</td>
<td>5000</td>
<td>7500</td>
<td>10000</td>
<td>12500</td>
<td>15000</td>
<td>17500</td>
<td>20000</td>
</tr>
</tbody>
</table>
Parallel Access Volume (PAV)
A Lab experiment

Linux cannot enable PAV on the ESS but can use it under VM
Base and Aliases (PAV Cont.)

- **IOCDS changes**
  
  ```
  IODEVICE ADDRESS=(5680,024),UNITADD=00,CUNUMBR=(5680), STADET=Y,UNIT=3390B
  IODEVICE ADDRESS=(5698,040),UNITADD=18,CUNUMBR=(5680), STADET=Y,UNIT=3390A
  ```

- **ATTACH Base and Aliases to the guest**

- **QUERY PAV shows base and alias addresses**

  ```
  cat /proc/dasd/devices
  ```

  ```
  5794(ECKD) at ( 94:  0) is dasda : active at blocksize: 4096, 1803060 blocks, 7043 MB
  5593(ECKD) at ( 94:  4) is dasdb : active at blocksize: 4096, 601020 blocks, 2347 MB
  5680(ECKD) at ( 94:  8) is dasdc : active at blocksize: 4096, 1803060 blocks, 7043 MB
  56bf(ECKD) at ( 94: 12) is dasdd : active at blocksize: 4096, 1803060 blocks, 7043 MB
  ```

  ```
  cat /proc/subchannels | grep "5680|56BF"
  ```

  ```
  5680 0030 3390/0C 3990/E9 yes FC FC FF C6C7C8CA CBC90000
  56BF 0031 3390/0C 3990/E9 yes FC FC FF C6C7C8CA CBC90000
  ```

This works only with z/VM
LVM commands (PAV Cont.)

- `vgscan`: create configuration data
  - scans all discs for volume groups
- `pvcreate /dev/dasdc1`
  - has to be done for each physical volume
- `vgcreate vg_kb /dev/dasdc1`
  - creates the volume group vg_kb
- `vgdisplay`
**vgdisplay**

```
vgdisplay -v vg_kb
--- Volume group ---
VG Name                 vg_kb
VG Access              read/write
VG Status              available/resizable
VG #                   0
MAX LV                 256
Cur LV                 0
Open LV                0
MAX LV Size            255.99 GB
Max PV                 256
Cur PV                 1
Act PV                 1
VG Size                6.87 GB
PE Size                4 MB
Total PE               1759
Alloc PE / Size        0 / 0
Free PE / Size         1759 / 6.87 GB
VG UUID                3nwJYn-SxWl-gKym-OvZs-TYIf-CrHP-inO5Yp
--- No logical volumes defined in "vg_kb" ---
```
More LVM commands

```
LVcreate --name lv_kb --extents 1759 vg_kb

lvscan
```

```
LVscan -- ACTIVE            "/dev/vg_kb/lv_kb" [6.87 GB]
LVscan -- 1 logical volumes with 6.87 GB total in 1 volume group
LVscan -- 1 active logical volumes
```
**Enable Paths**

*pvpath*-change or query path attributes of a physical multipathed volume

**pvpath** -qa

Physical volume /dev/dasdc1 of vg_kb has 2 paths:

<table>
<thead>
<tr>
<th>Device</th>
<th>Weight</th>
<th>Failed</th>
<th>Pending</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td># 0:</td>
<td>94:9</td>
<td>0</td>
<td>0</td>
<td>0 enabled</td>
</tr>
<tr>
<td># 1:</td>
<td>94:13</td>
<td>0</td>
<td>0</td>
<td>disabled</td>
</tr>
</tbody>
</table>

The second path can be enabled:

**pvpath** -p1 -ey /dev/dasdc1

vg_kb: setting state of path #1 of PV#1 to enabled

**pvpath** -qa

Physical volume /dev/dasdc1 of vg_kb has 2 paths:

<table>
<thead>
<tr>
<th>Device</th>
<th>Weight</th>
<th>Failed</th>
<th>Pending</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td># 0:</td>
<td>94:9</td>
<td>0</td>
<td>0</td>
<td>0 enabled</td>
</tr>
<tr>
<td># 1:</td>
<td>94:13</td>
<td>0</td>
<td>0</td>
<td>enabled</td>
</tr>
</tbody>
</table>

Now LVM is ready to use both paths to the volume
Results

iozone sequential write/read 1 disk

<table>
<thead>
<tr>
<th>Paths</th>
<th>Write (MB/s)</th>
<th>Read (MB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14.9</td>
<td>27.0</td>
</tr>
<tr>
<td>2</td>
<td>18.7</td>
<td>46.4</td>
</tr>
<tr>
<td>3</td>
<td>22.4</td>
<td>65.9</td>
</tr>
<tr>
<td>4</td>
<td>23.4</td>
<td>81.4</td>
</tr>
<tr>
<td>5</td>
<td>23.2</td>
<td>96.9</td>
</tr>
<tr>
<td>6</td>
<td>22.6</td>
<td>106.7</td>
</tr>
<tr>
<td>7</td>
<td>21.2</td>
<td>106.7</td>
</tr>
<tr>
<td>8</td>
<td>21.1</td>
<td>119.0</td>
</tr>
</tbody>
</table>

These are preliminary results in a controlled environment.
PAV is not yet officially supported with Linux on zSeries!
ESS – Disk I/O

- Don't treat ESS as a black box, understand its structure
- The default is close to worst case:
  - You ask for 16 disks and your SysAdmin gives you
  - addresses 5100-510F
- What's wrong with that?
Let's have a deeper look to the elements of the scenario:

- **CHPIDs**
- **Host Adapter (HA) supporting FCP (FCP port)**
  - 16 Host Adapters, organized in 4 bays, 4 ports each
- **Device Adapter Pairs (DA)**
  - each one supports two loops
- **Disks are organized in ranks**
  - each rank (8 physical disks) implements one RAID 5 array (with logical disks)
ESS Architecture

Scenarios: single disk, single rank

- **FCP Switch 2109**
- **FCP CHPID**
- **FCP CHPID**
- **FCP CHPID**
- **FCP CHPID**

- **z900 2064**

- **CHPIs**

- **Host Adapter (HA) supporting FCP (FCP port)**
  - 16 Host Adapters, organized in 4 bays, 4 ports each

- **Device Adapter Pairs (DA)**
  - each one supports two loops

- **Disks are organized in ranks**
  - each rank (8 physical disks) implements one RAID 5 array (with logical disks)
ESS Architecture

Scenario: single host adapter

ansson: single host adapter

- FCP Switch 2109
- FCP CHPID
- z900 2064

- CHPIDs
- Host Adapter (HA) supporting FCP (FCP port)
  - 16 Host Adapters, organized in 4 bays, 4 ports each
- Device Adapter Pairs (DA)
  - each one supports two loops
- Disks are organized in ranks
  - each rank (8 physical disks) implements one RAID 5 array (with logical disks)
ESS Architecture

Scenario: single CHPID

- **FCP Switch 2109**
- **FCP CHPID**
- **FCP CHPID**
- **FCP CHPID**
- **z900 2064**

- **CHPID**
- **DA**
- **HA Bay 1**
- **HA Bay 2**
- **HA Bay 3**
- **HA Bay 4**

- **Cluster Processor Complex**
  - 4 way SMP RISC system

- **Device Adapter Pairs (DA)**
  - each one supports two loops

- **Disks are organized in ranks**
  - each rank (8 physical disks) implements one RAID 5 array (with logical disks)
Scenario: two CHPIDs

- **FCP Switch 2109**

  - **FCP CHPID**
  - **FCP CHPID**
  - **FCP CHPID**
  - **FCP CHPID**

- **z900 2064**

- **FCP Switch 2109**

  - **HA Bay 1**
  - **HA Bay 2**
  - **HA Bay 3**
  - **HA Bay 4**

- **CPI (Common Parts Interconnect)**

- **Cluster Processor Complex** - 4-way SMP RISC system
  - **DA**
  - **DA**
  - **DA**
  - **DA**

- **Device Adapter Pairs (DA)**
  - each one supports two loops

- **Disks are organized in ranks**
  - each rank (8 physical disks) implements one RAID 5 array (with logical disks)

- **Cloud (Common Parts Interconnect)**

- **Loop B**
  - **Loop A**

- **ESS 2105**

- **CHPIs**

- **Host Adapter (HA) supporting FCP (FCP port)**
  - 16 Host Adapters, organized in 4 bays, 4 ports each
ESS Architecture

**Scenario: four CHPIDs (4C4H4R ESS 2105)**

- **FCP Switch 2109**
- **z900 2064**

- **CHPIIDs**

- **Host Adapter (HA) supporting FCP (FCP port)**
  - 16 Host Adapters, organized in 4 bays, 4 ports each

- **Device Adapter Pairs (DA)**
  - each one supports two loops

- **Disks are organized in ranks**
  - each rank (8 physical disks) implements one RAID 5 array (with logical disks)
### Summary of the Scenarios:

<table>
<thead>
<tr>
<th>Scenario</th>
<th>used resources</th>
<th>limiting resource</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CHPIDs</td>
<td>HA</td>
</tr>
<tr>
<td>single Disk</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>single Rank</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>single Host Adapter</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>single CHPID</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>two CHPIDs</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>maximum available = 4C4H4R ESS 2105</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

**Benchmark used for measuring:** [Iozone](http://www.iozone.org)

- multi process sequential file system I/O
- each process writes and reads a 350 MB file on a separate disk
- System: LPAR, 4 CPUs, 128 MB main memory, Linux 2.4.17 with hz timer off
- scaling was: 1, 2, 4, 8, 16 processes
- the maximum throughput values were taken as result
- 1 HA limits to 40MB/s write and 65 MB/s read, regardless of the number of ranks
- 4 HA are limiting to 125 MB/s write and 240 MB/s read, but 4 CHPIDs are required to make use of it
- 31 bit and 64 bit difference is small
- it is expected that the values further increase using more ranks, HA, CHPIDs
General Rules

- this makes it **slow**:
  - when all disks are from one rank and accessed via the same path

- this makes it **fast**:
  - use many host adapters
  - spread the host adapters used across all host adapter bays
  - use as much CHPIDs as possible and access each disk through all CHPIDs, if possible (FICON, LVM1-mp)
  - spread the disks used over all ranks equally

- this applies to FCP and FICON
Visit us!

- Linux for zSeries Performance Website:

- Linux-VM Performance Website:

- Performance Redbook:
  - SG24-6926-00
Questions