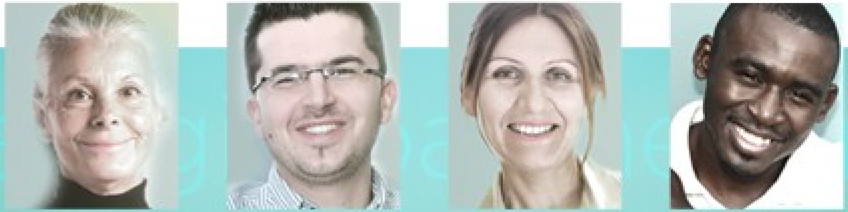
SHARE
Technology - Connections - Results

Linux on System z Performance Update - Part 3 I/O options

Mario Held
IBM Research & Development, Germany

August 28, 2009
Session Number 2193



Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

DB2*
DB2 Connect
DB2 Universal Database
e-business logo
GDPS*
Geographically Dispersed Parallel Sysplex
HyperSwap
IBM*
IBM eServer
IBM logo*
Parallel Sysplex*
System z
Tivoli*
VM/ESA*
WebSphere*
z/OS*
z/VM*
zSeries*

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Intel is a registered trademark of the Intel Corporation in the United States, other countries or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Introductory Remark

- Problem report statistics in level 3 service
  - Approximately 50% are system performance related
  - Thereof 70% are related to disk I/O
  - Good news: most can be solved easily

Areas for tuning

- **Linux**
  - **File system**
  - **I/O options**
  - **I/O scheduler**
  - **Logical volumes**
  - **Multipathing**
- Host
  - FICON/ECKD and FCP/SCSI specialties
  - Channels
- Storage server
  - Disk configuration
- Read ahead setup

Linux file system

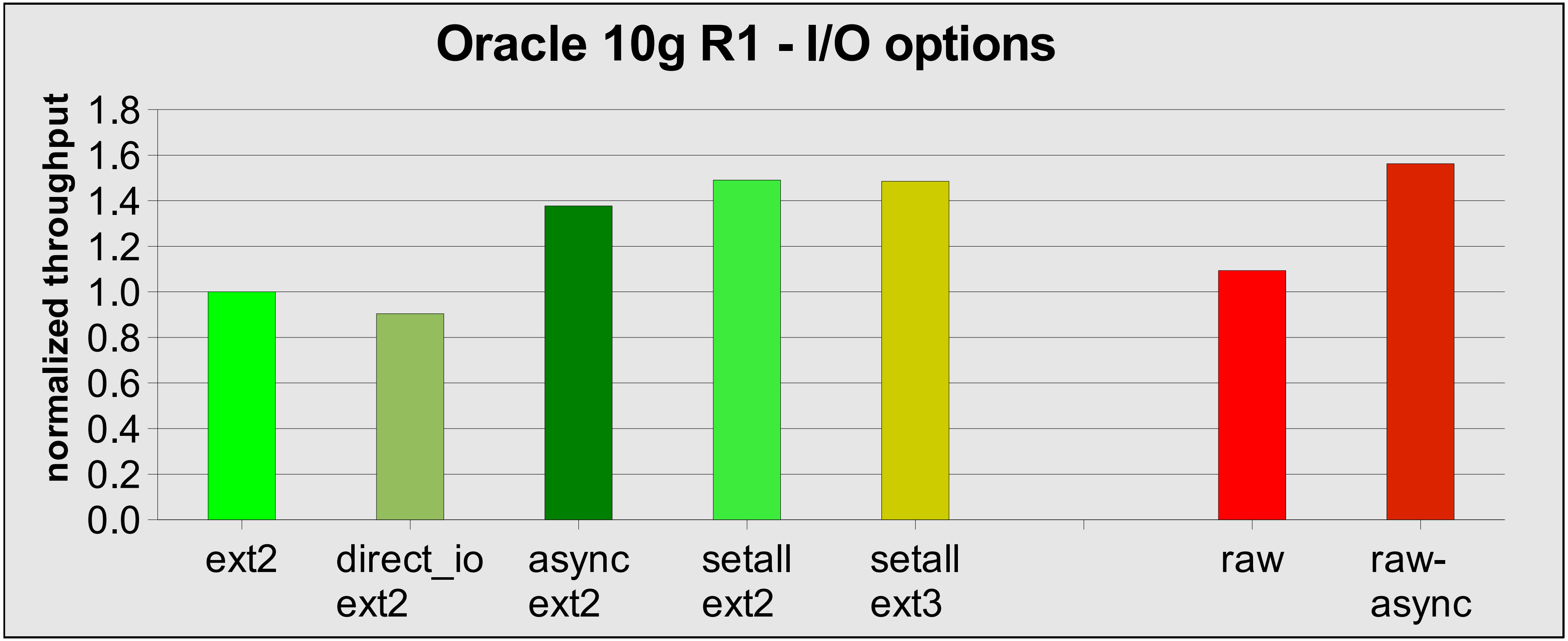
- Use ext3 instead of reiserfs
- Tune your ext3 file system
  - Select the appropriate journaling mode (journal, ordered, writeback)
  - Consider to turn off atime
  - Enabling directory indexing
- Temporary files
  - Don't place them on journaling file systems
  - Consider a ram disk instead of a disk device
- If possible, use direct I/O to avoid double buffering of files in memory
- If possible, use async I/O to continue program execution while data is fetched
  - Important for read operations
  - Applications usually are not depending on write completions

Linux 2.6 Disk I/O Options

- Direct I/O (DIO)
  - Transfer the data directly from the application buffers to the device driver, avoids copying the data to the page cache
  - Advantages:
    - Saves page cache memory and avoids caching the same data twice
    - Allows larger buffer pools for databases
  - Disadvantage:
    - Make sure that no utility is working through the file system (page cache) --> danger of data corruption
- Asynchronous I/O (AIO)
  - Application is not blocked for the time of the I/O operation
  - Resumes its processing and gets notified when the I/O is completed.
  - Advantage
    - the issuer of a read/write operation is no longer waiting until the request finishes.
    - reduces the number of I/O processes (saves memory and CPU)
- Recommendation: use both if possible with your application

Linux 2.6 Disk I/O Options - Results

- The combination of direct I/O and async I/O (setall) shows best results when using the Linux file system. Best throughput however was seen with raw I/O and async I/O.
- ext2 and ext3 lead to identical throughput



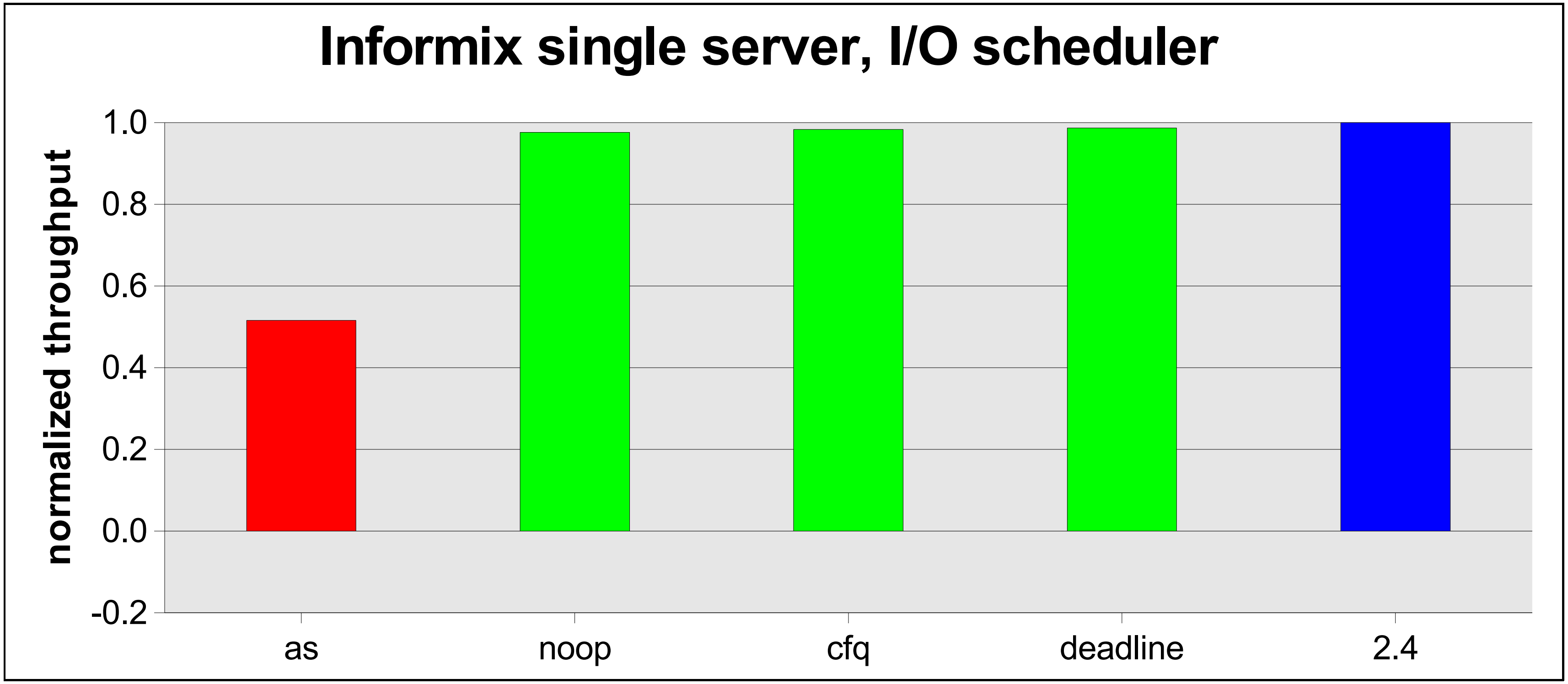
Linux 2.6 I/O Schedulers

- Four different I/O schedulers are available
  - noop scheduler
    only request merging
  - deadline scheduler
    avoids read request starvation
  - Anticipatory (as) scheduler
    designed for the usage with physical disks, not intended for storage subsystems
  - complete fair queuing (cfq) scheduler
    all users of a particular drive would be able to execute about the same number of I/O requests over a given time.
- deadline is the default in the current distributions

```
h05lp39:~ # dmesg | grep scheduler
io scheduler noop registered
io scheduler anticipatory registered
io scheduler deadline registered (default)
io scheduler cfq registered
```

Linux 2.6 I/O Schedulers - Results

- “as” scheduler is not a good choice for System z
- All other schedulers show similar results as the kernel 2.4 scheduling



Logical volumes

- Linear logical volumes allow an easy extension of the file system
- Striped logical volumes
  - Provide the capability to perform simultaneous I/O on different stripes
  - Allow load balancing
  - Are also extendable
- Don't use logical volumes for /root or /usr
  - If the logical volume gets corrupted your system is gone
- Logical volumes require more CPU cycles than physical disks
  - Increases with the number of physical disks used for the logical volume

Multipathing

- Multipathing is used to exploit multiple paths to disk devices for high availability or load balancing reasons
  - Multipathing with failover (high availability)
    - Should one path fail, the operating system can route I/O through one of the remaining paths
    - Path switching is transparent to the applications
  - Multipathing with multibus (load balancing)
    - Uses all paths in a priority group in a round-robin manner, e.g. path is switched after each 1000 I/O requests
    - Connects a device over multiple independent paths to provide load balancing.

Setting up multipathing (failover/multibus)

- To activate multipathing you need:
  - Multipath-tools package installed
  - dm-multipath module loaded
  - More than one path to your SCSI disks
  - Check/modify your /`etc/multipath.conf`

```
path_grouping_policy  failover // Multipathing with failover
path_grouping_policy  multibus // Multipathing with multibus
```

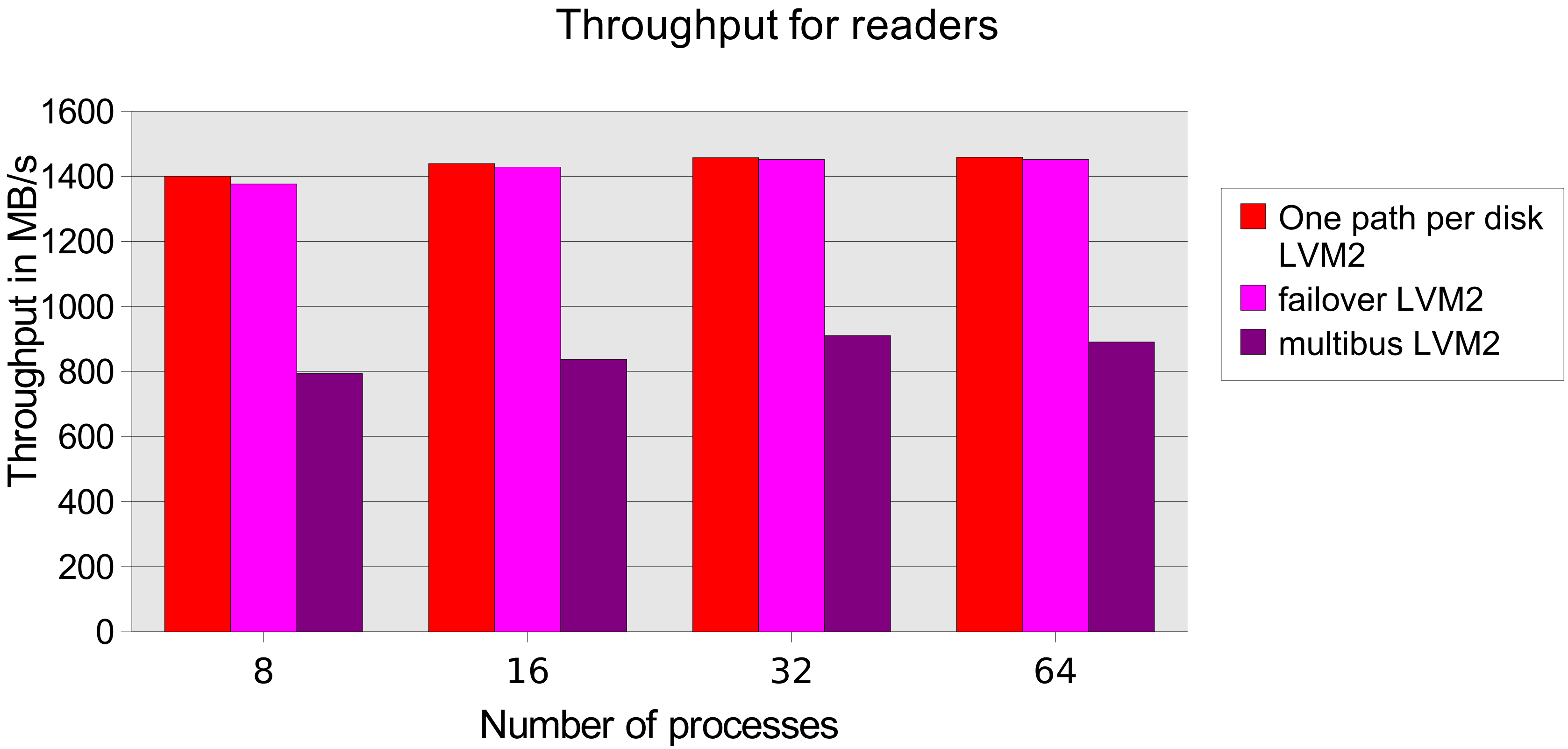
  - Multibus only:

```
rr_min_io 1000 // the number of I/O to route to a path
 before switching
```

  - Multipath daemon started
- For further information please visit our HOWTO pages at http://www.ibm.com/developerworks/linux/linux390/perf/tuning_how_dasd_multipath.html

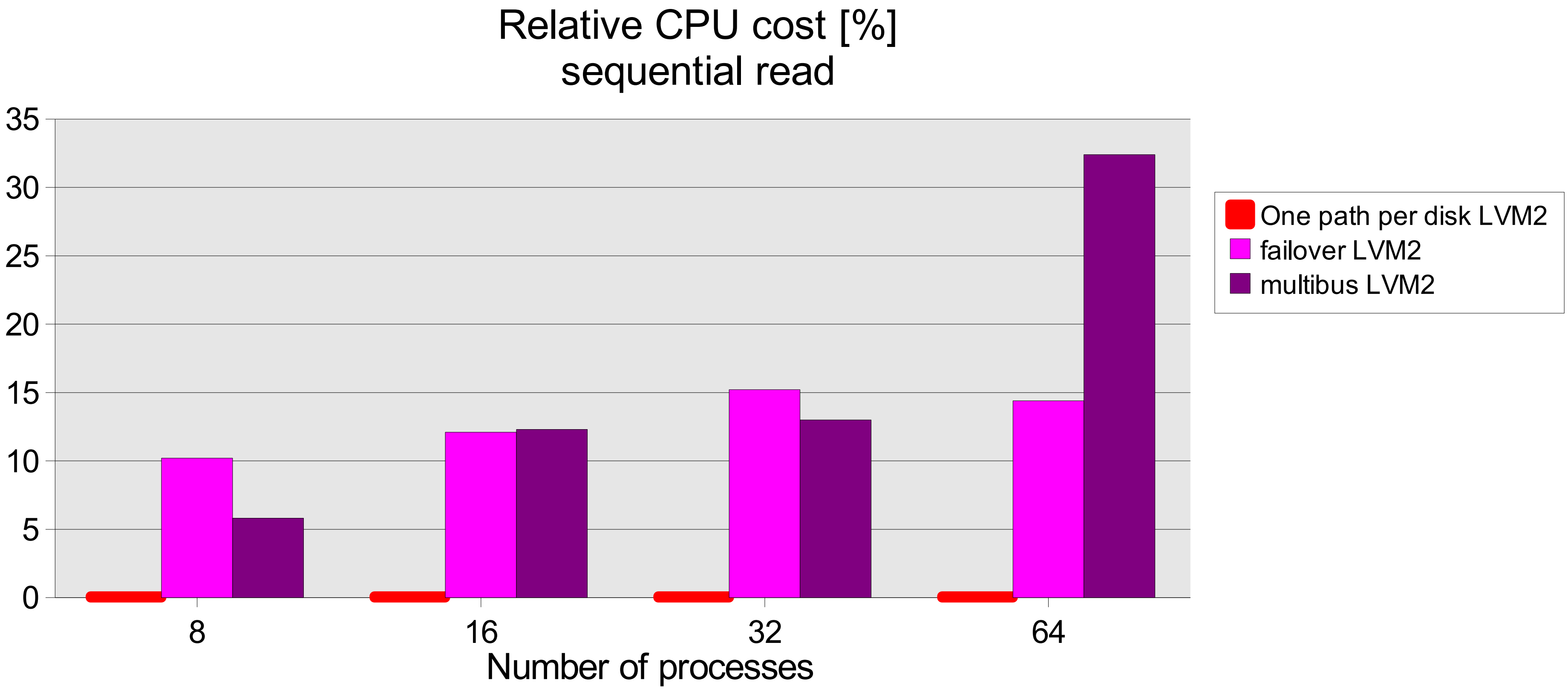
FCP/SCSI single path versus multipath

- Logical volume with 16 disks connected with and w/o multipathing
- Use failover instead of multibus

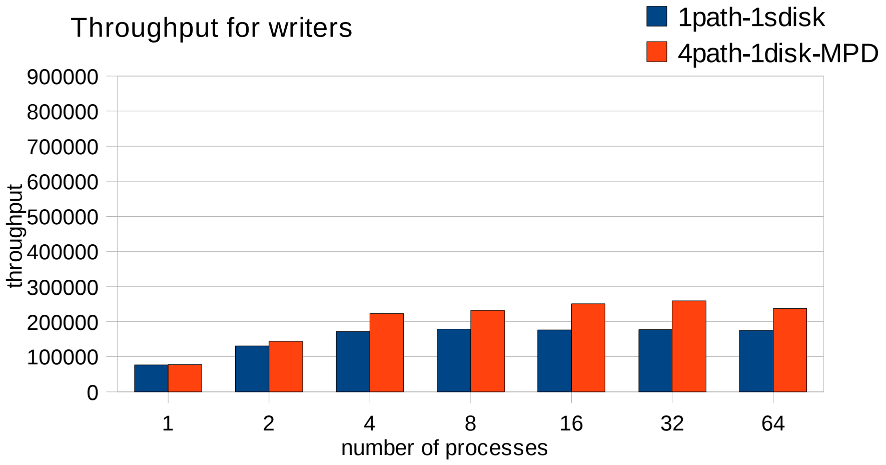


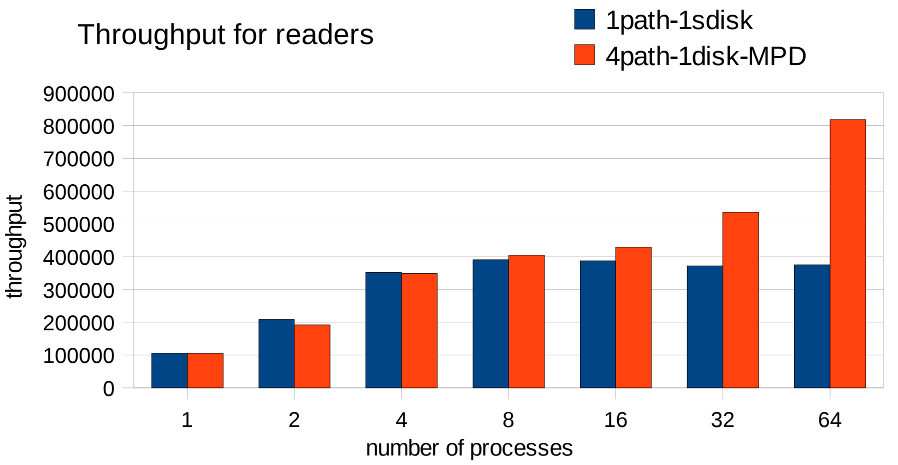
FCP/SCSI single path versus multipath(2)

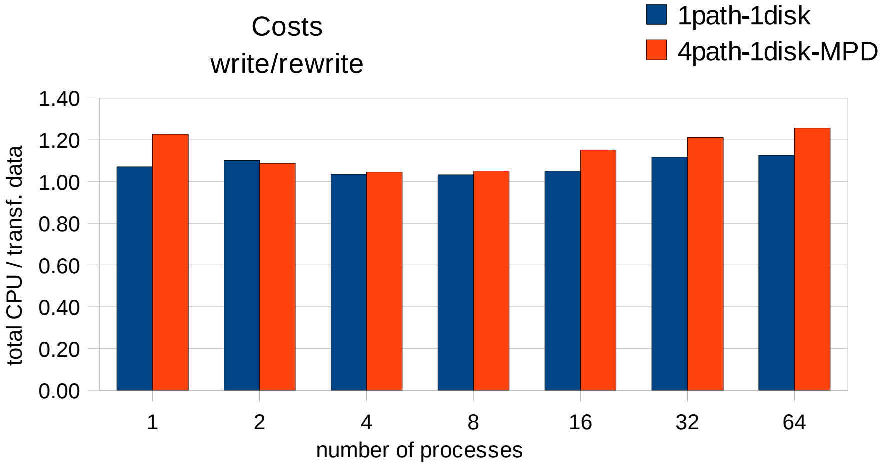
- 10% to 15% more CPU utilization with failover compared to one path
- CPU utilization for multibus is difficult to predict

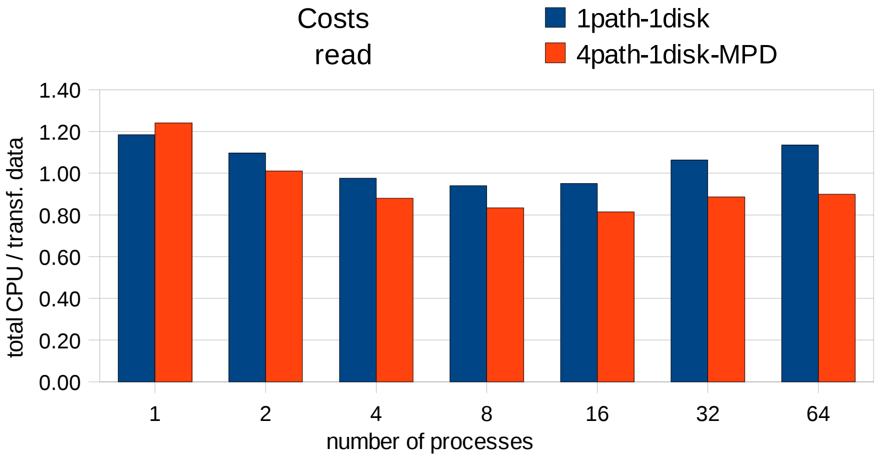


Single path vs. multipath/multibus – Sequential I/O DIO









Conclusion

- When you are using Linux on System z, and you want to establish fail-save connections to the storage server, we recommend to use multipathing.

- Multipath with failover has an performance advantage over multipath with multibus when **multiple** SCSI disks are attached via LVM.

- Multipath with multibus is the solution if you access a **single** SCSI disk and need maximum possible throughput. This increases the bandwidth.

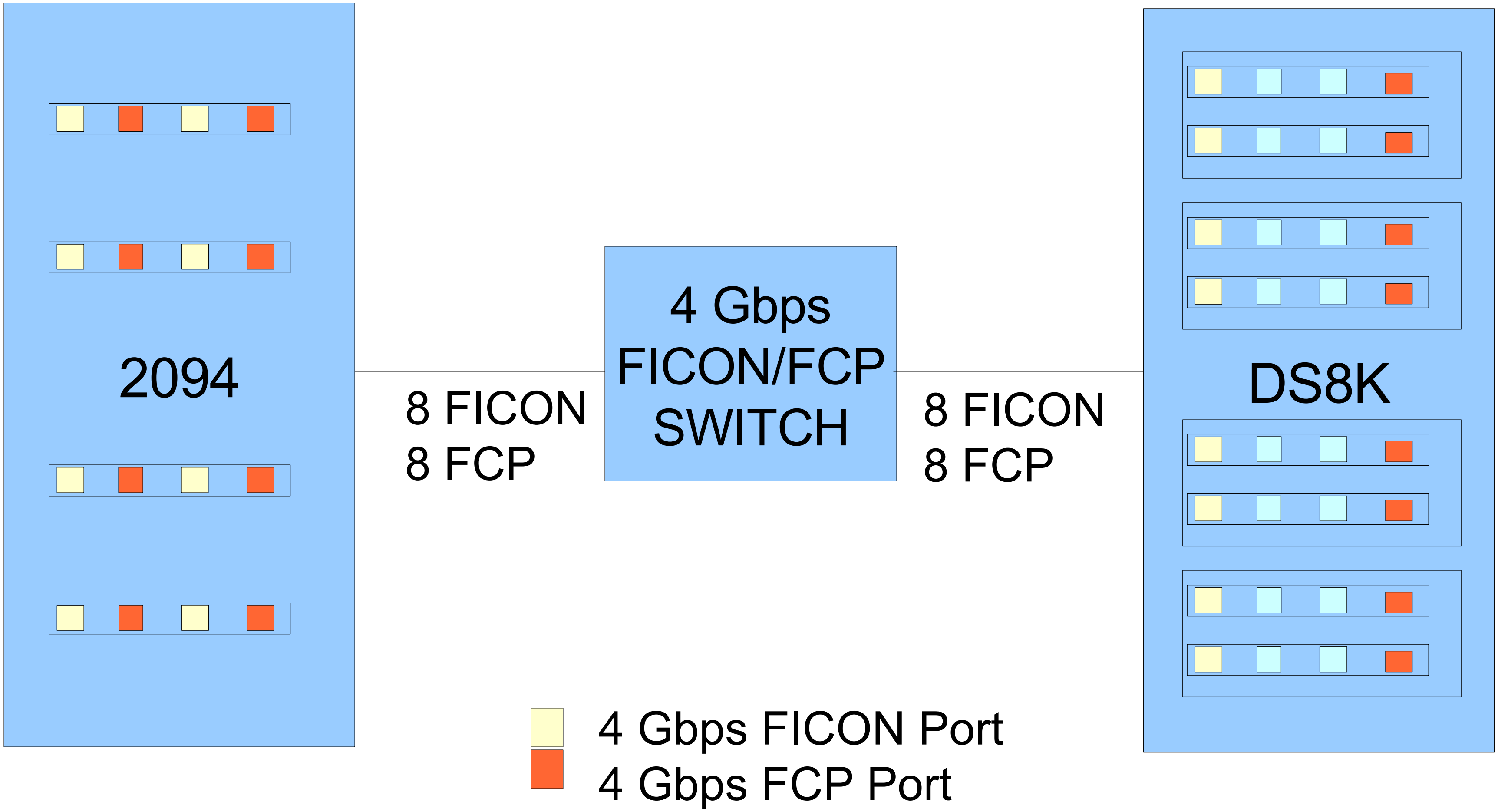
Areas for tuning

- Linux
  - File system
  - I/O options
  - I/O scheduler
  - Logical volumes
  - Multipathing
- **Host**
  - **FICON/ECKD and FCP/SCSI specialties**
  - **Channels**
- Storage server
  - Disk configuration
- Read ahead setup

I/O processing characteristics

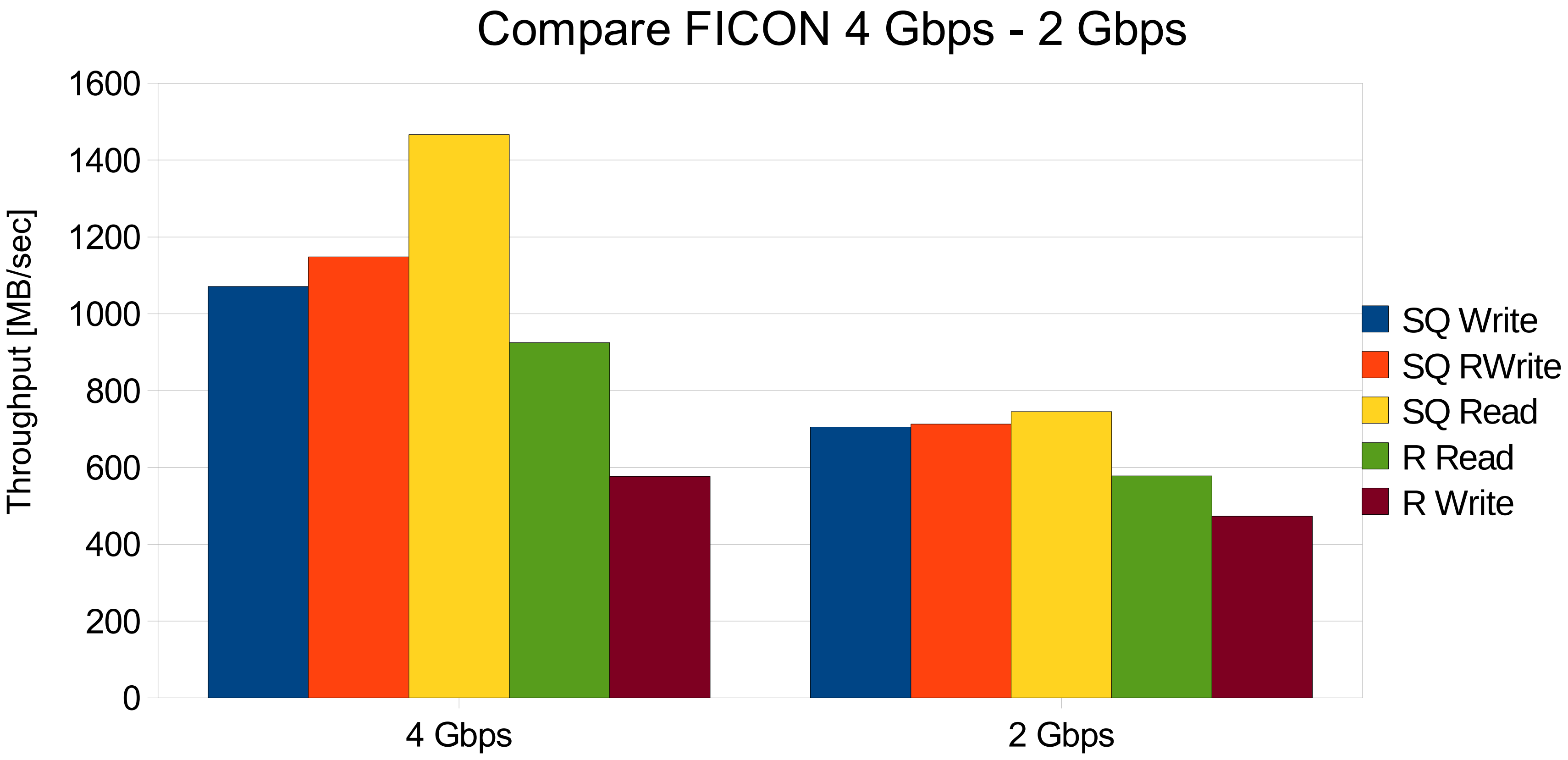
- FICON/ECKD:
  - 1:1 mapping host subchannel:dasd
  - Serialization of I/Os per subchannel
  - I/O request queue in Linux
  - Disk blocks are 4KB
  - High availability by FICON path groups
  - Load balancing by FICON path groups and Parallel Access Volumes
- FCP/SCSI
  - Several I/Os can be issued against a LUN immediately
  - Queuing in the FICON Express card and/or in the storage server
  - Additional I/O request queue in Linux
  - Disk blocks are 512 bytes
  - High availability by Linux multipathing, type failover
  - Load balancing by Linux multipathing, type multibus

Configuration 4Gbps disk I/O measurements



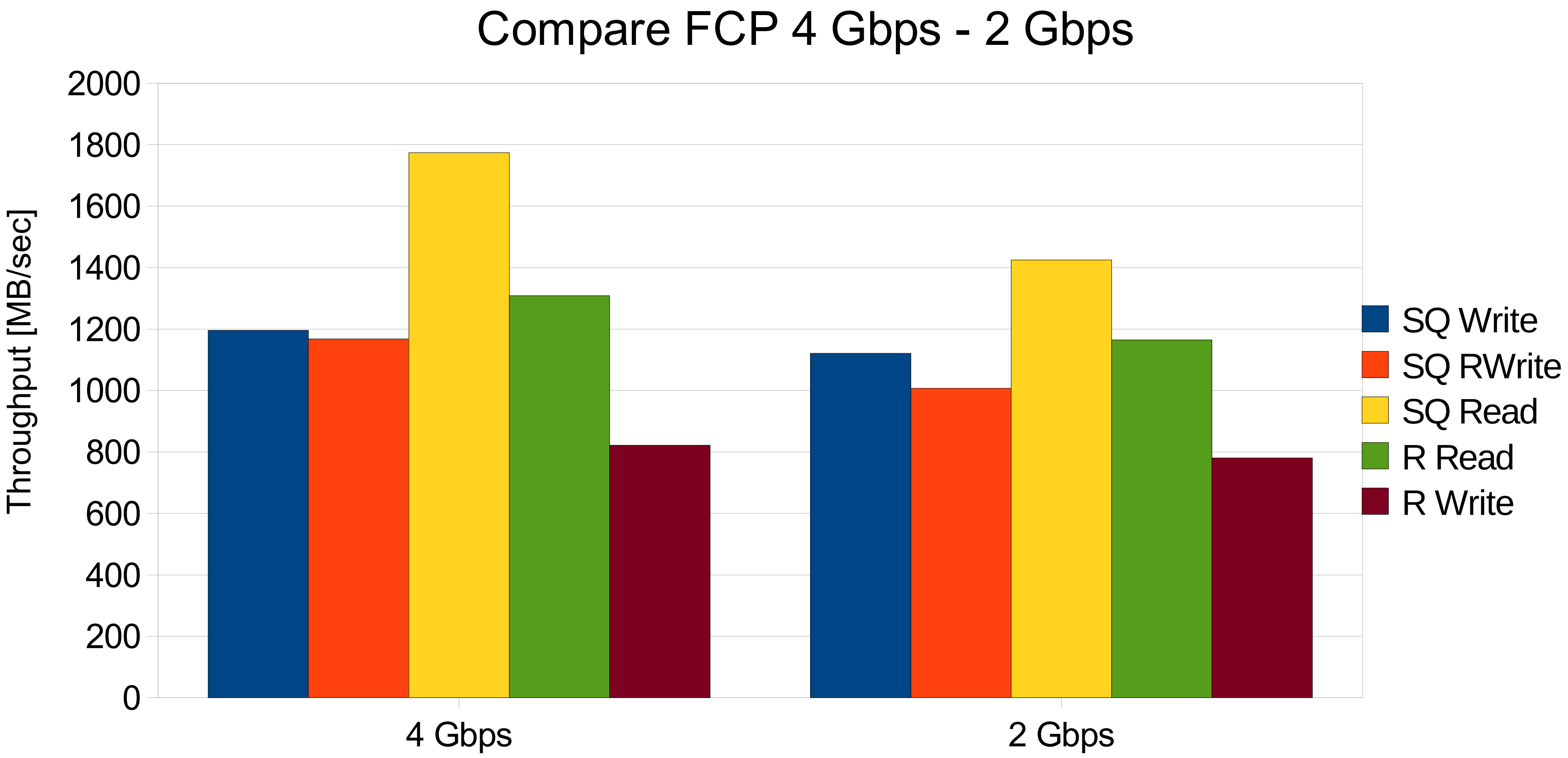
Disk I/O performance with 4Gbps links - FICON

- Strong throughput increase (average x 1.6)
- Doubles with sequential read



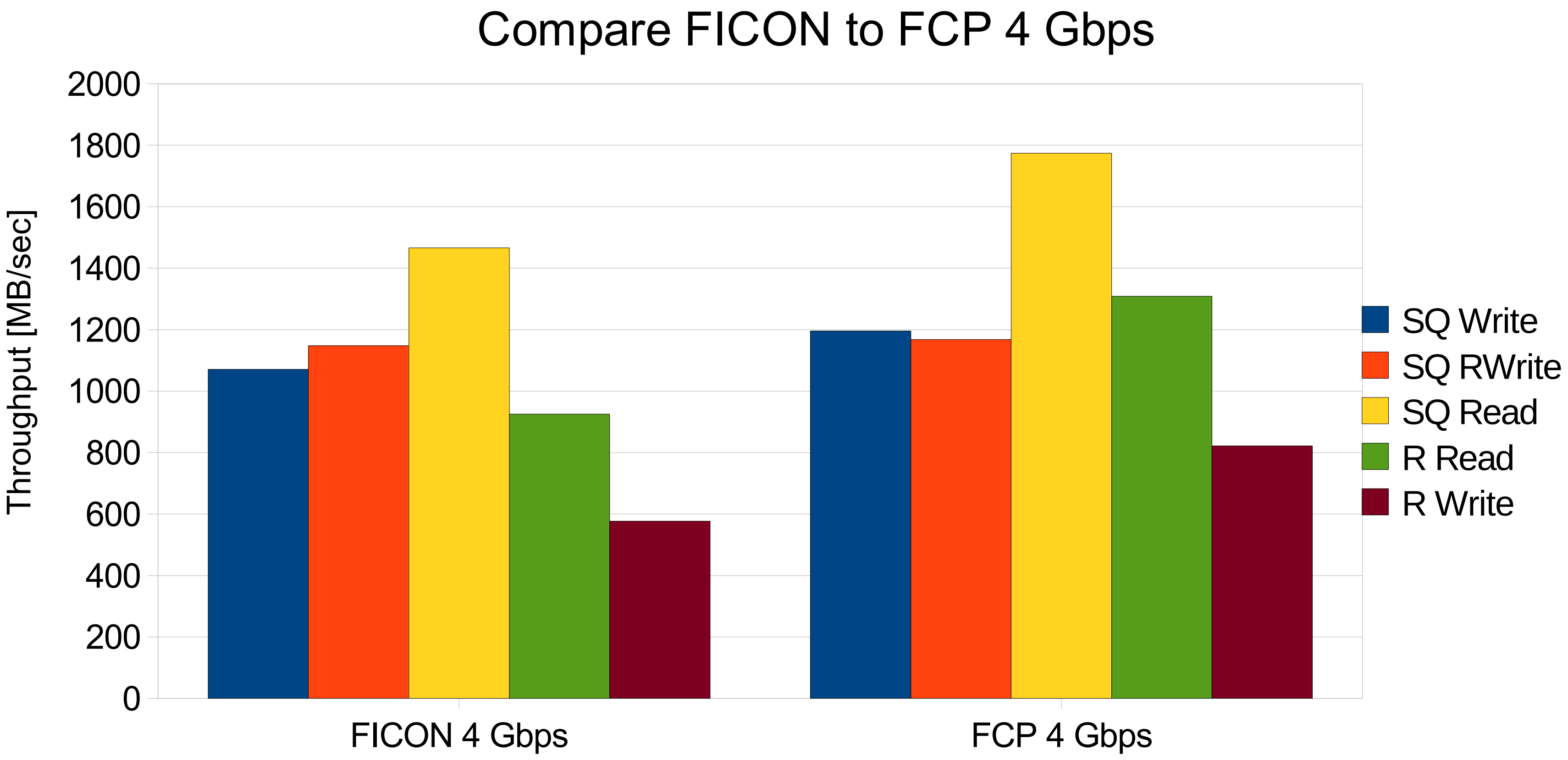
Disk I/O performance with 4Gbps links - FCP

- Moderate throughput increase
- Highest increase with sequential read at x 1.25



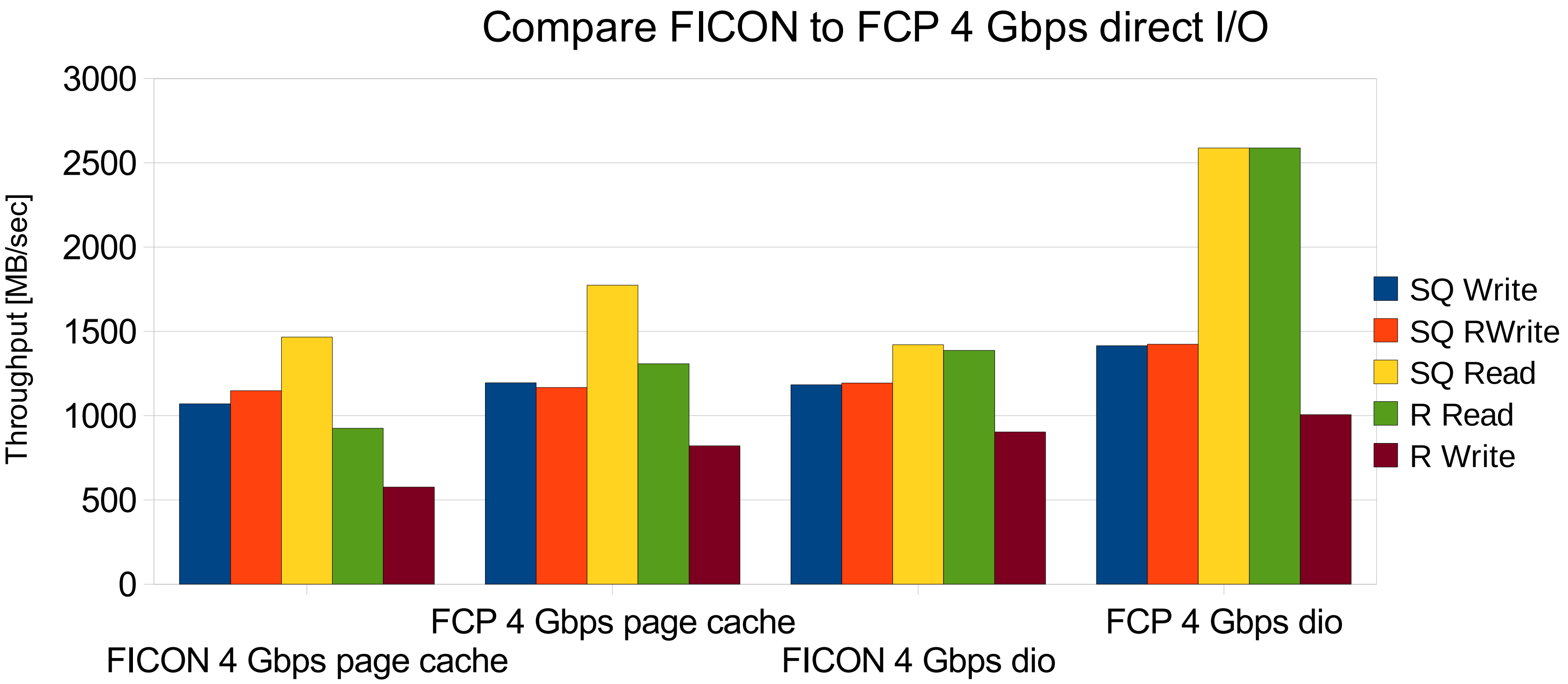
Disk I/O performance with 4Gbps links – FICON versus FCP

- Throughput for sequential write is similar
- FCP throughput for random I/O is 40% higher



Disk I/O performance with 4Gbps links – FICON versus FCP / direct I/O

- Bypassing the Linux page cache improves throughput for FCP up to 2x, for FICON up to 1.6x.
- Read operations are much faster on FCP



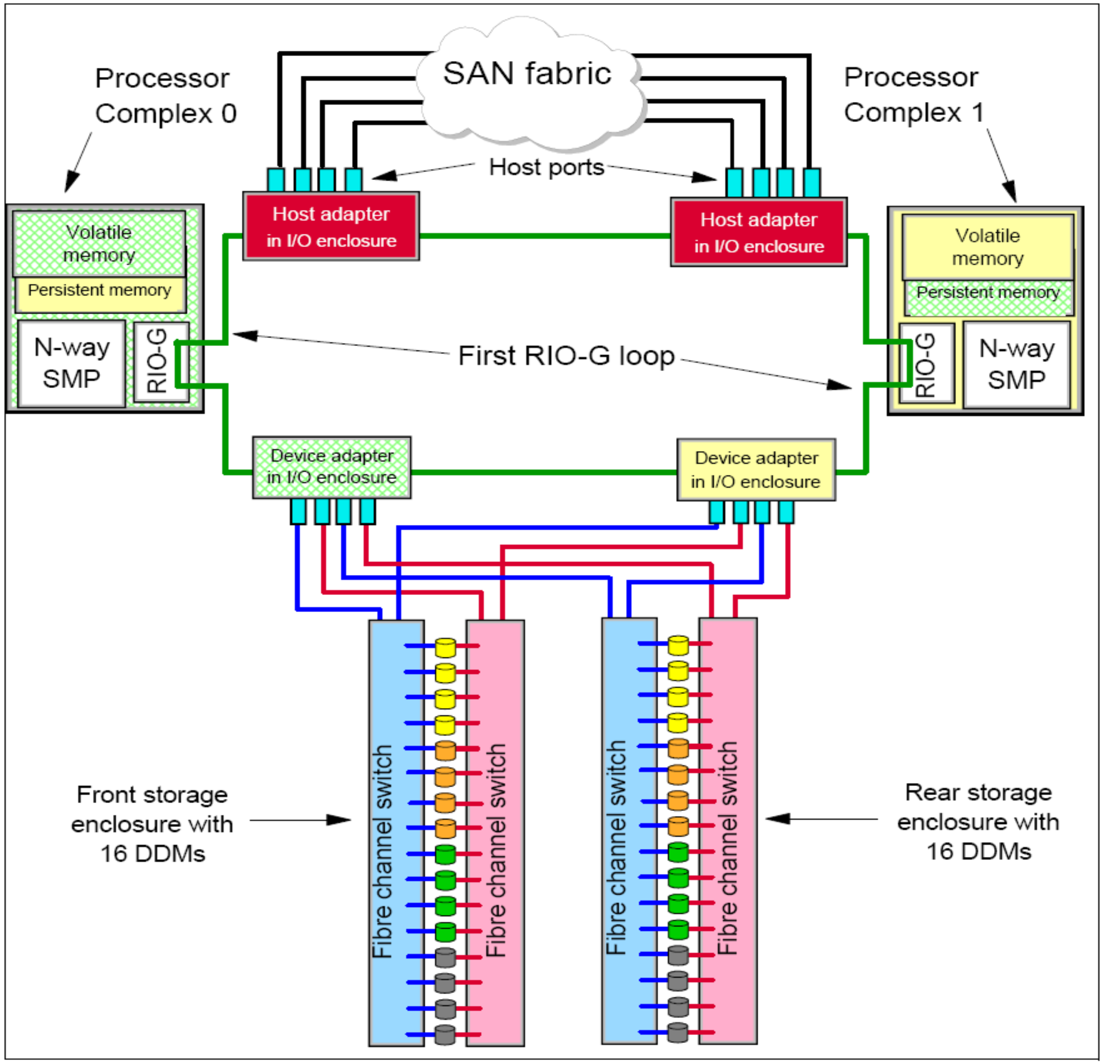
Areas for tuning

- Linux
  - File system
  - I/O options
  - I/O scheduler
  - Logical volumes
  - Multipathing
- Host
  - FICON/ECKD and FCP/SCSI specialties
  - Channels
- Storage server
  - Disk configuration
- Read ahead setup

(DS8000) Disk setup

- Don't treat a storage server as a black box, understand its structure
- Enable storage pool striping if available
- Principles apply to other storage vendor products as well
- You ask for 16 disks and your system administrator gives you addresses 5100-510F
    - From a performance perspective this is close to the worst case
- So - what's wrong with that?

DS8000 Architecture



- **structure** is complex
  - disks are connected via two internal FCP switches for higher bandwidth
- the DS8000 is still divided into two parts named **processor complex** or just **server**
  - caches are organized per server
- one **device adapter pair** addresses 4 array sites
- one **array site** is build from 8 disks
  - disks are distributed over the front and rear storage enclosures
  - have the same color in the chart
- one **RAID array** is defined using one array site
- one **rank** is built using one RAID array
- ranks are assigned to an **extent pool**
- extent pools are assigned to **one of the servers**
  - this assigns also the caches
- - one disk range resides in one extent pool

Rules for selecting disks

- **Goal is to get a balanced load on all paths and physical disks**
- Use as many paths as possible (CHPID -> host adapter)
  - ECKD switching of the paths is done automatically
  - FCP needs a fixed relation between disk and path
    - Establish a fixed mapping between path and rank in our environment
    - Taking a disk from another rank will then use another path
- Switch the rank for each new disk in a logical volume
- Switch the ranks used between servers and device adapters
- Select disks from as many ranks as possible!
- Avoid reusing the same resource (path, server, device adapter, and disk) as long as possible

Disk Setup

SHARE
Technology · Connections · Results

	FICON/ECKD															
Ficon channels	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
StorageServer ports	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
Server#	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
DA pair#	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
DA#	0	3	4	7	1	2	5	6	0	3	4	7	1	2	5	6
Rank#	0	5	8	13	1	4	9	12	2	7	10	15	3	6	11	14
Disk#	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
	FCP/SCSI															
FCP channel#	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
WWPN#	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Server#	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
DA pair#	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
DA#	0	2	4	6	1	3	5	7	0	2	4	6	1	3	5	7
Rank#	16	20	24	28	18	22	26	30	16	20	24	28	18	22	26	31
Disk#	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63

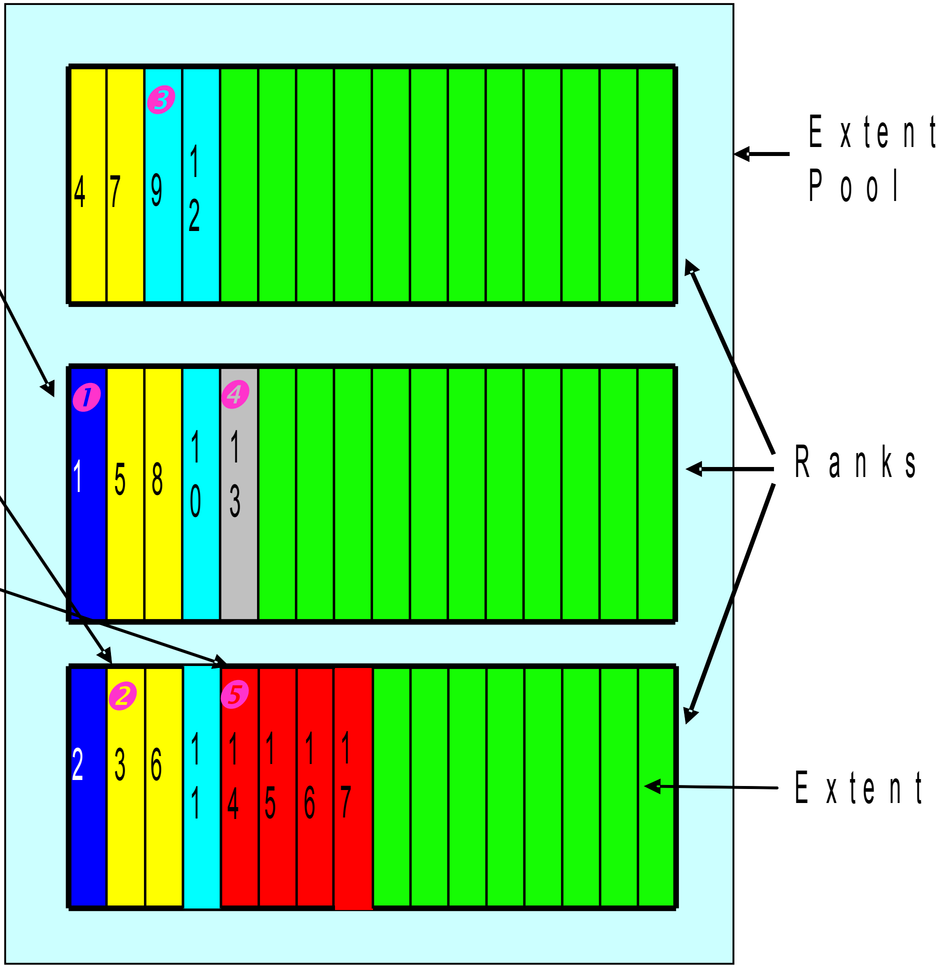
Logical Volume and Storage Pool Striping

LUN / CKD volume is created

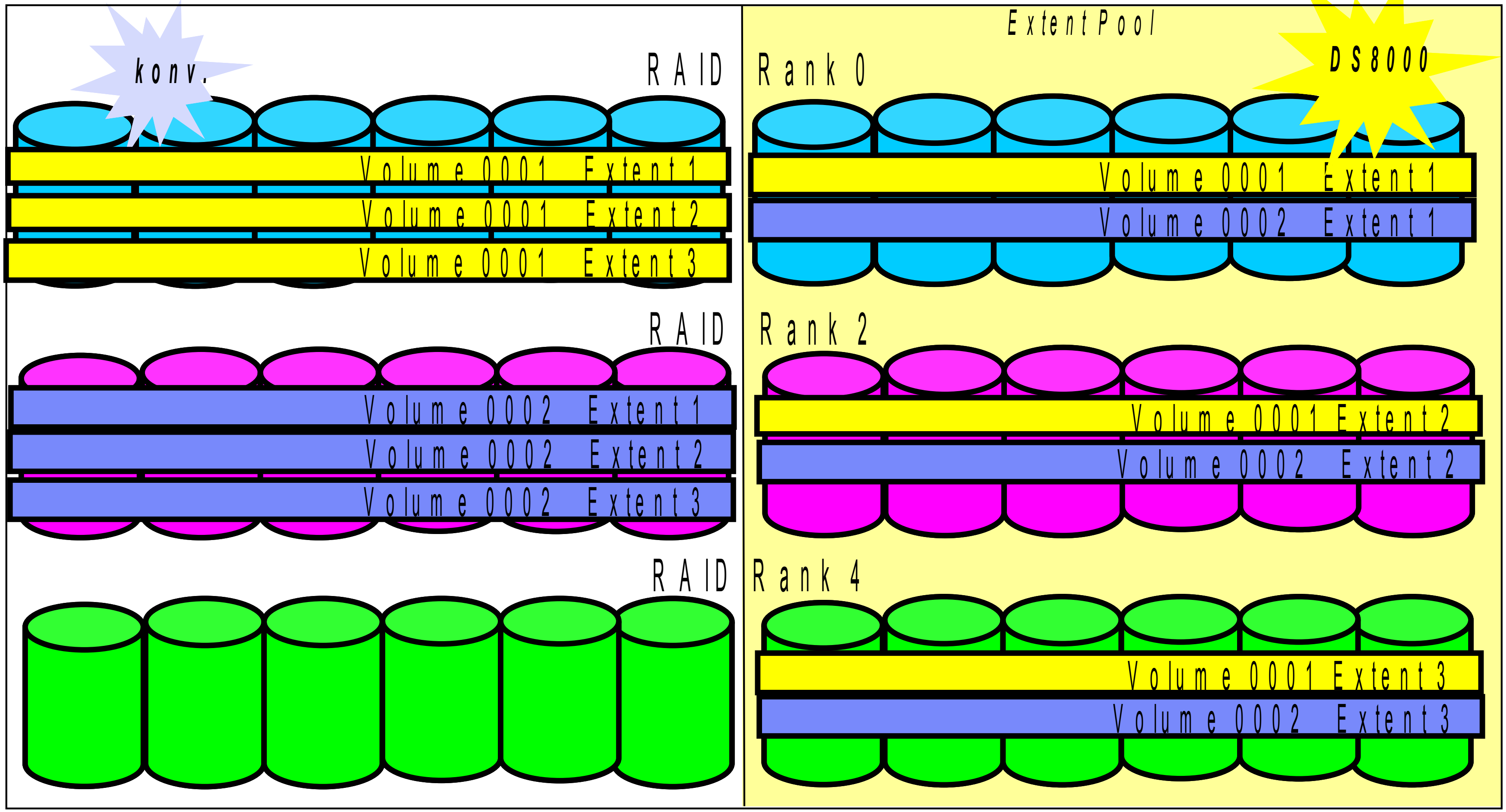
1 Striped volume with two Extents created

2 Next striped volume starts at next rank (six extents in this example)

5 Non-striped volume created Starts at next rank



Storage Pool Striping



Striped Volumes

	LVM striping	DS8000 storage pool striping
striping is done in	Linux	storage server
effort to construct the volume	take care of picking subsequent disks from different ranks	configure storage server
administrating disks within Linux	can be challenging, e.g. several hundred for a database	simple
volume extendable ?	yes	no
maximum I/O request size	stripe size (e.g. 64KB)	maximum provided by the device driver (e.g. 512KB)
multipathing	SCSI: assign paths round robin to disks, multipath failover ECKD: path group	SCSI: multipath multibus, ECKD: path group
usual disk sizes	LV = many disks SCSI 10GB to 20GB, ECKD mod9 or mod27	Volume = 1 disk, SCSI unlimited, e.g. 300GB, ECKD max. mod54
extent pool	1 rank	multiple ranks
maximum number of ranks for the constructed volume	total number of ranks	Total number of one server side (50%)

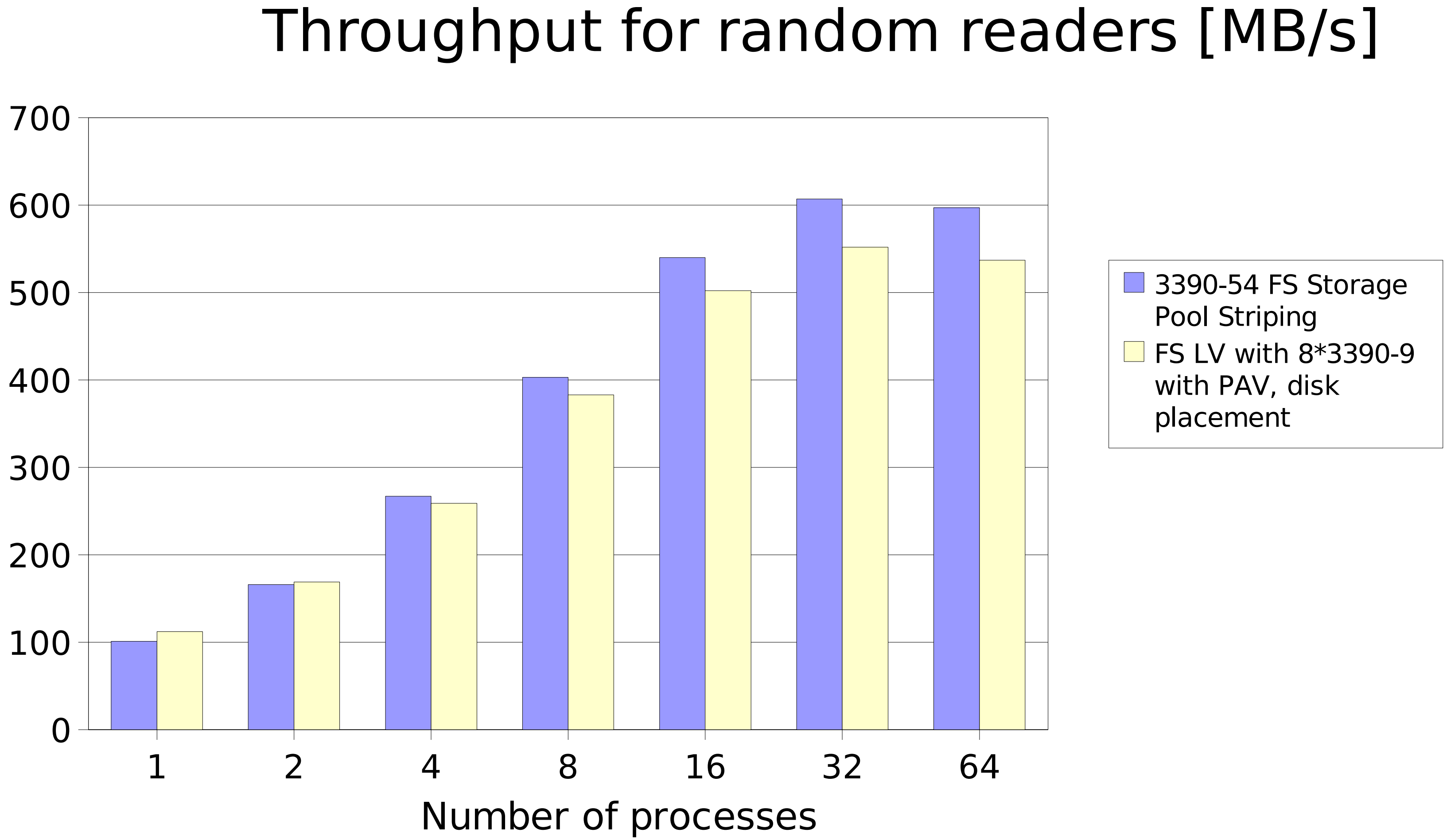
Impact of DS8000 Storage Pool Striping

- License Machine Code 5.30xx.xx
- Stripe the extents of a DS8000 Logical Volume across multiple RAID arrays
- Will improve throughput for some workloads
- 1 GB granularity, random workloads will generally benefit more than sequential ones
- Cannot span servers
- Can be combined with LVM striping or DB2 database container striping
- Risk: Losing one rank means loosing one extent pool
  - Tip: not more than 4 to 8 ranks par extent pool
- → Assumption: dedicated disk placement no longer necessary

Measurement Setup

- System z9 LPAR, 8 CPUs, 256 MB
- DS8000
  - Server 0
    - 1 Extent-Pool over 8 ranks
    - 1 3390-54 Volume defined
  - Server 1
    - 8 Extent-Pools (1 per rank)
    - 8 x 3390-9 Volumes defined
    - LV with dedicated disk placement
- 4 x 4 Gb/s Ficon
- Internal Driver
- HyperPAV, 63 Aliases per Server

Iozone Test


Throughput for random readers [MB/s]
700
600
500
400
300
200
100
0
1
2
4
8
16
32
64
Number of processes
3390-54 FS Storage Pool Striping
FS LV with 8*3390-9 with PAV, disk placement

Striped volumes
– results and recommendations (1)

- General pros / cons
  - Storage pool striped volumes are as simple to set up and to administrate as a few large disks
  - Striping on the storage device lowers CPU consumption (LVM) on the Linux side
  - Stripe size is 1 GB
  - Rank failure will hit all disks

Striped volumes
– results and recommendations (2)

- Results with ECKD disks
  - Combination with HyperPAV reaches nearly the same performance as Linux solution
  - Without HyperPAV only one I/O outstanding per DASD is possible, which limits the performance
  - FICON path groups doing the load balancing
- Results with SCSI disks
  - Linux striped logical volumes are faster but the logical volume manager takes more CPU cycles than e.g. the multipath daemon
  - For random workloads the multipath daemon used to distribute workload to the FCP channels needs improvements
- If you don't use striping in Linux today, consider to enable it at least in the storage server – your performance won't become worse

General recommendations

- FICON/ECKD
  - Storage pool striped disks (no disk placement)
  - HyperPAV (SLES11)
  - Large volume (future distributions)
- FCP/SCSI
  - Linux LV with striping (disk placement)
  - Multipath with failover

Areas for tuning

- Linux
  - File system
  - I/O options
  - I/O scheduler
  - Logical volumes
  - Multipathing
- Host
  - FICON/ECKD and FCP/SCSI specialties
  - Channels
- Storage server
  - Disk configuration
- Read ahead setup

Read ahead setup

- Read ahead in to many stages
  - Applications like DBMS
    - With low hit workload it could be disadvantage
    - If large columns are accessed can be advantage
  - LVM
    - Caches per default 1024 pages
    - Change setting for read ahead to <n pages> with command

      ```
      lvchange -r <n pages> /dev/<volume group>/<logical volume>
      ```
  - Linux block device layer
    -
    - Set the value to 0 using the blockdev command

      ```
      blockdev --setra 0 /dev/sda
      ```
  - Storage subsystem caching
    - Set the appropriate modes
    - Use the normal caching mode which adopts to the workload
- Plan your read ahead setup
  - Makes often no sense in all stages in parallel
  - Prefer read ahead if done by the application

Visit us !

- Linux on System z: Tuning Hints & Tips
  http://www.ibm.com/developerworks/linux/linux390/perf/

- Linux-VM Performance Website:
  http://www.vm.ibm.com/perf/tips/linuxper.html

- IBM Redbooks
  http://www.redbooks.ibm.com/

- IBM Techdocs
  http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/Techdocs

Questions