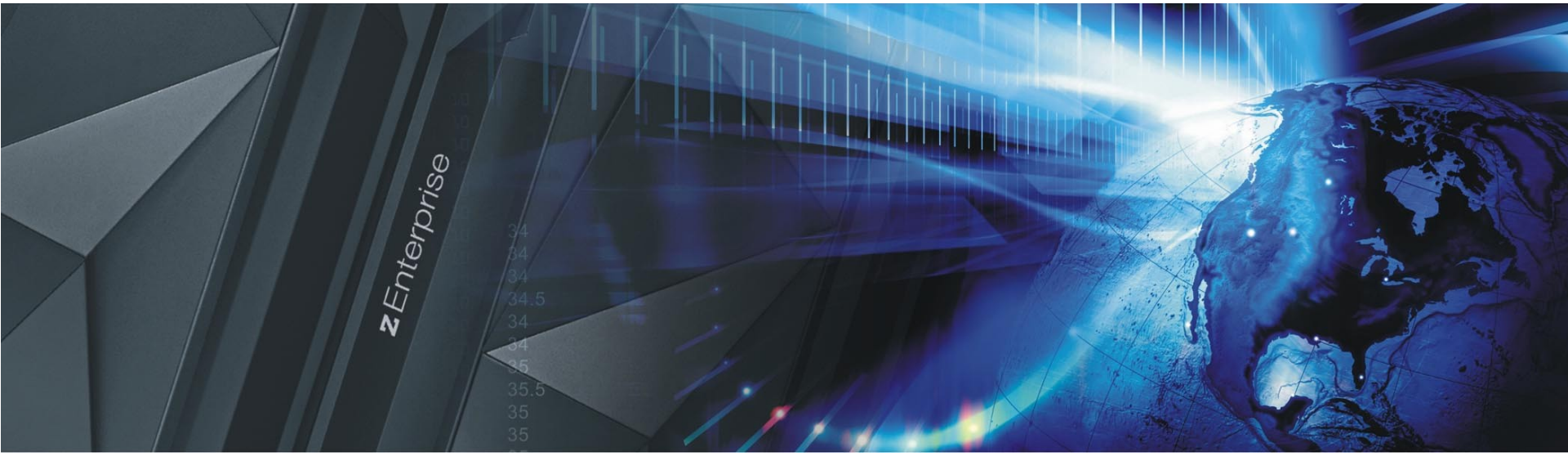
Linux on System z - Disk I/O Performance – Part 2

Mustafa Mešanović, IBM R&D Germany, System Performance Analyst



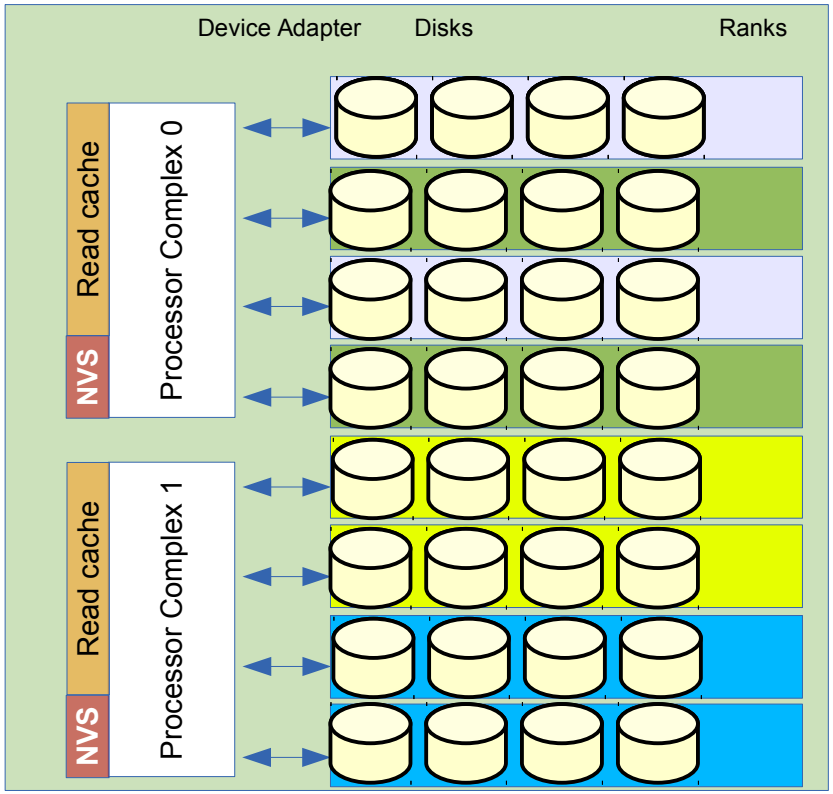
Trademarks

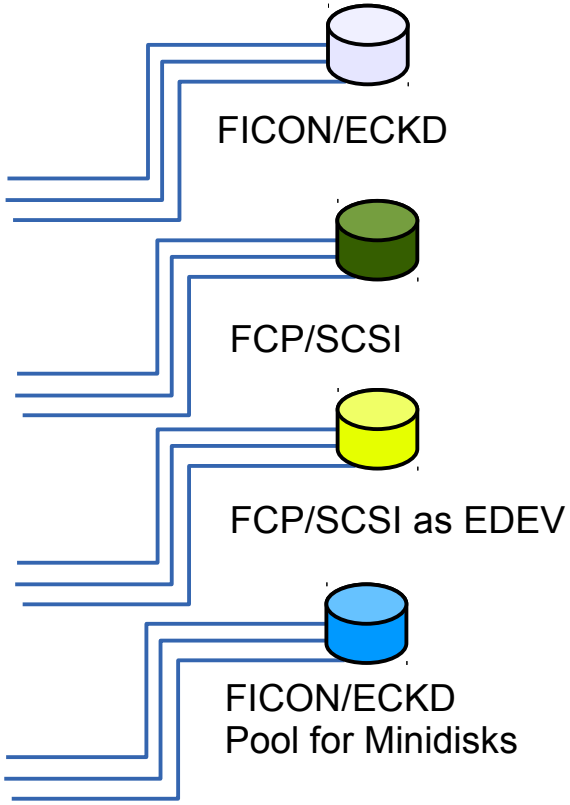
Agenda

- Storage server setup
  - common parts, storage pool striping...
- Disk I/O configurations for FICON/ECKD and FCP/SCSI
  - and its possible advantages and bottlenecks
  - a simple comparison of FICON/ECKD vs. FCP/SCSI
    in a OLTP-like workload
- Summary / Conclusion

Disk I/O – DS8000 Storage Server Family

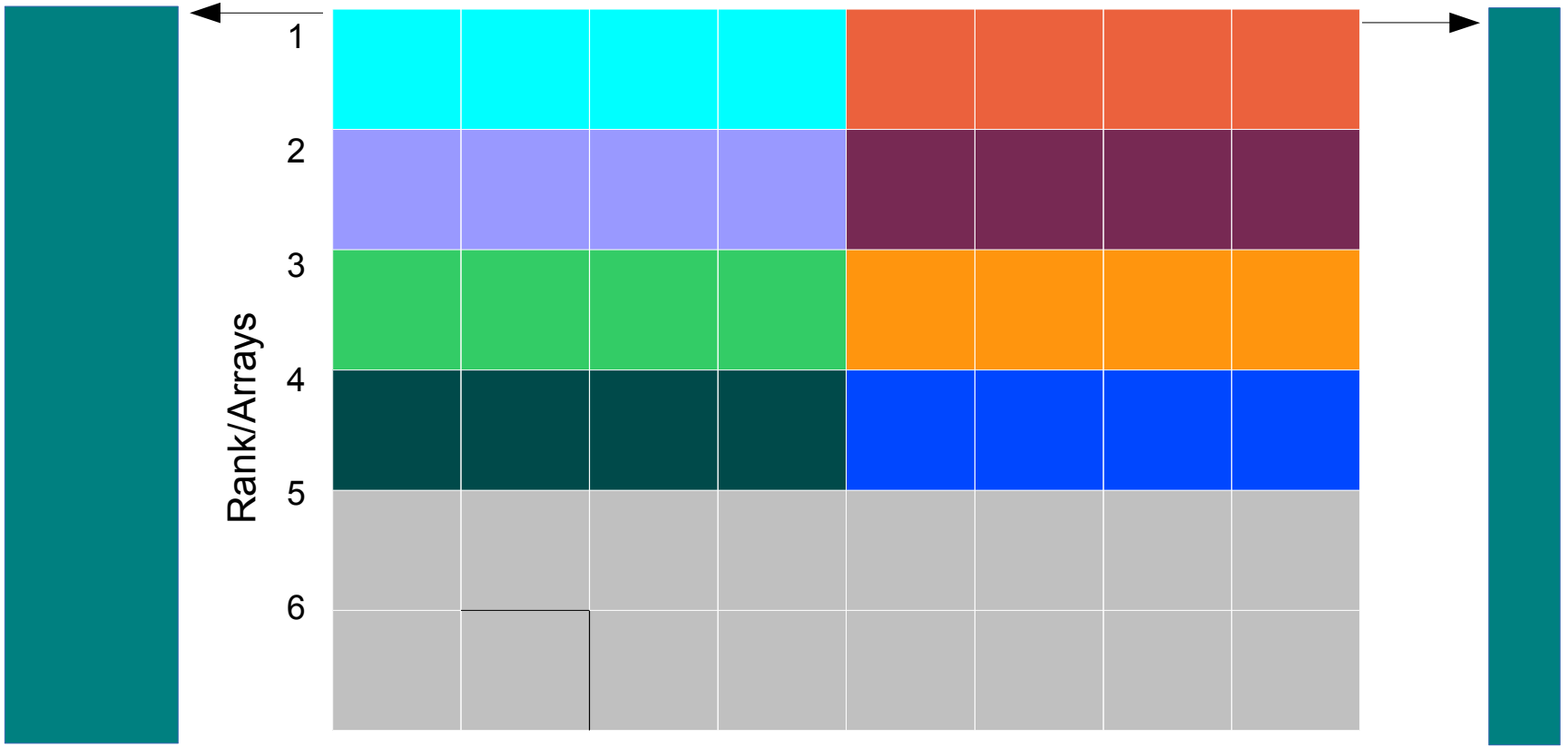
- Storage server basics – various configurations possible
  - Preferable many ranks into one extent pool with Storage Pool Striping (extents striped over all ranks within extent pool)





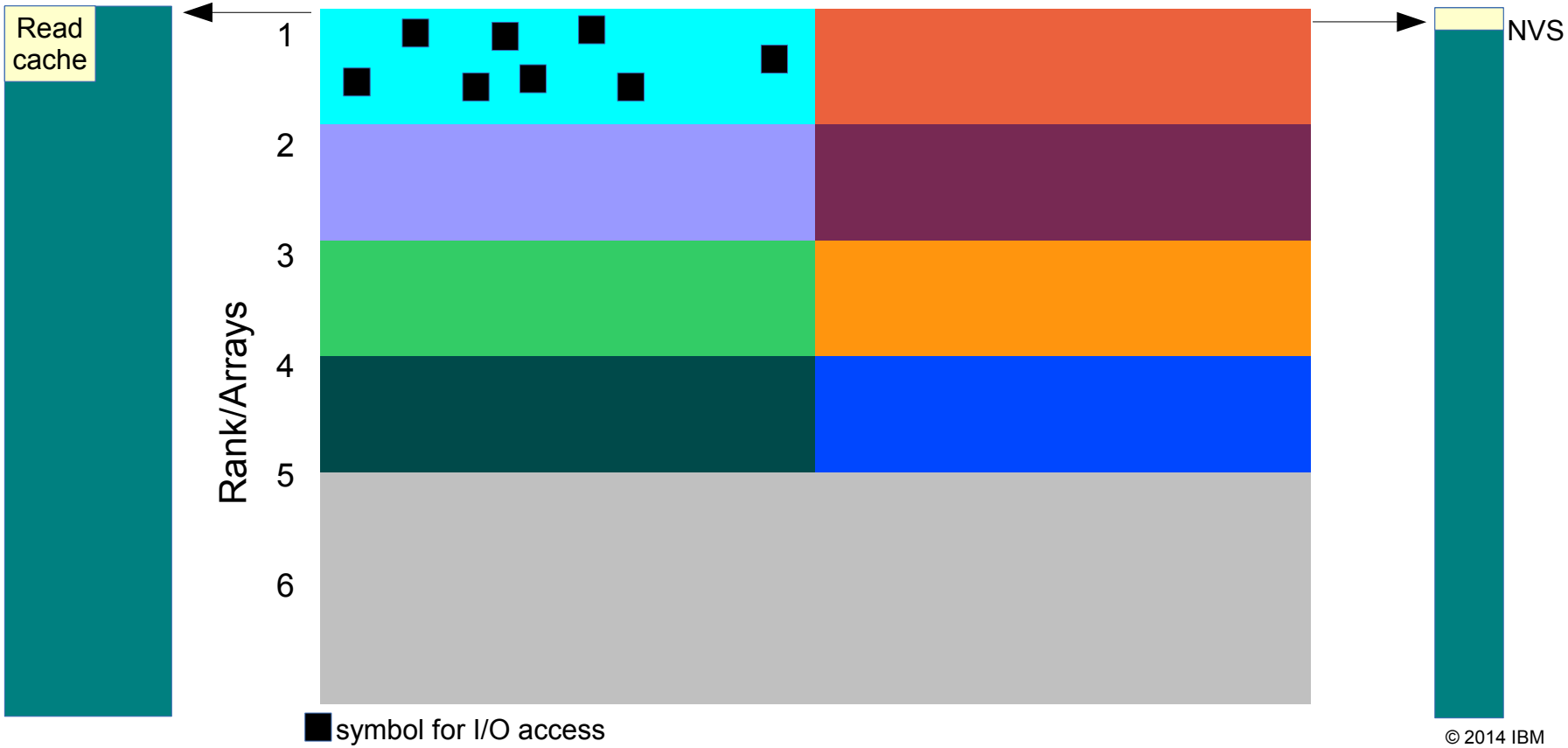
Disk I/O – Volumes

- Extent pool with 8 volumes each with 4 GiB
  - Each rank has access to an adequate portion of the read cache and non-volatile storage (NVS – write cache)
- Example: random access to one volume
  - Usable portions of read cache and NVS very limited because just one rank is involved
  - Only one Device Adapter in use



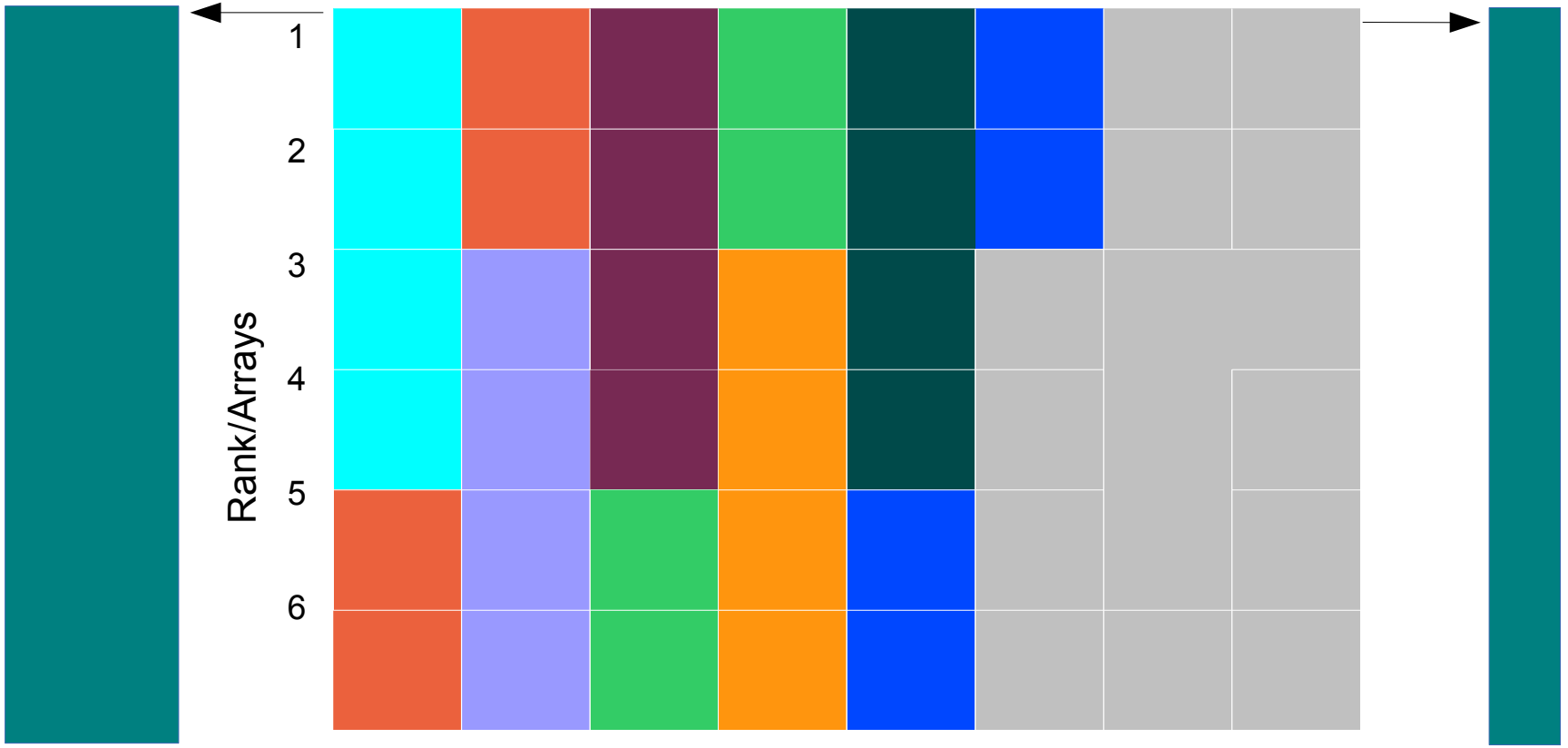
Disk I/O – Volumes

- Extent pool with 8 volumes each with 4 GiB
  - Each rank has access to an adequate portion of the read cache and non-volatile storage (NVS – write cache)
- Example: random access to one volume
  - Usable portions of read cache and NVS very limited because just one rank is involved
  - Only one Device Adapter in use



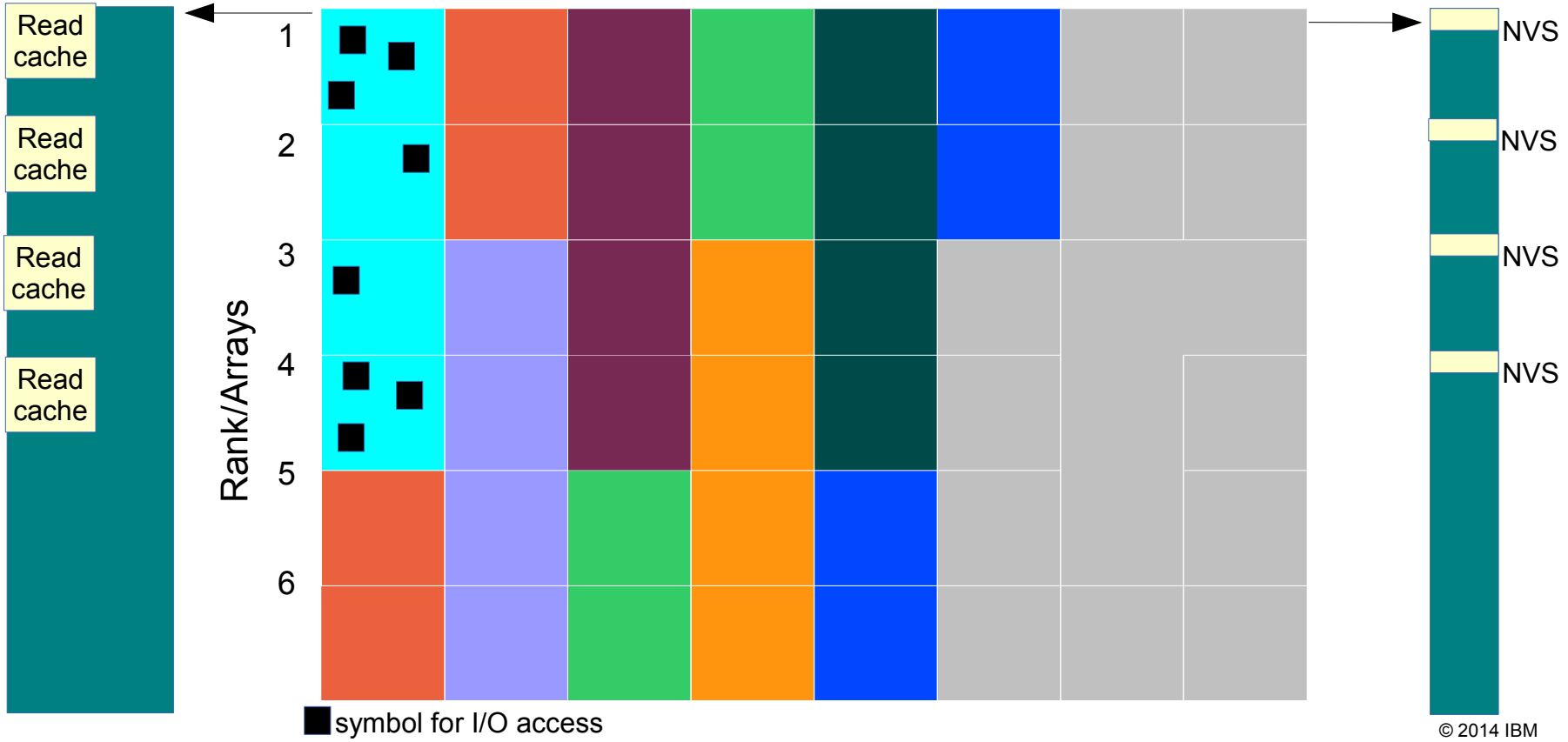
Disk I/O – Volumes with Storage Pool Striping (SPS)

- Extent pool example with 8 volumes each with 4 GiB and with Storage Pool Striping (SPS)
  - Each rank has access to an adequate portion of the read cache and non-volatile storage (NVS – write cache)
- Example: random access to one SPS volume
  - Usable portions of read cache and NVS much bigger because four ranks are involved
  - Up to four Device Adapters are in use



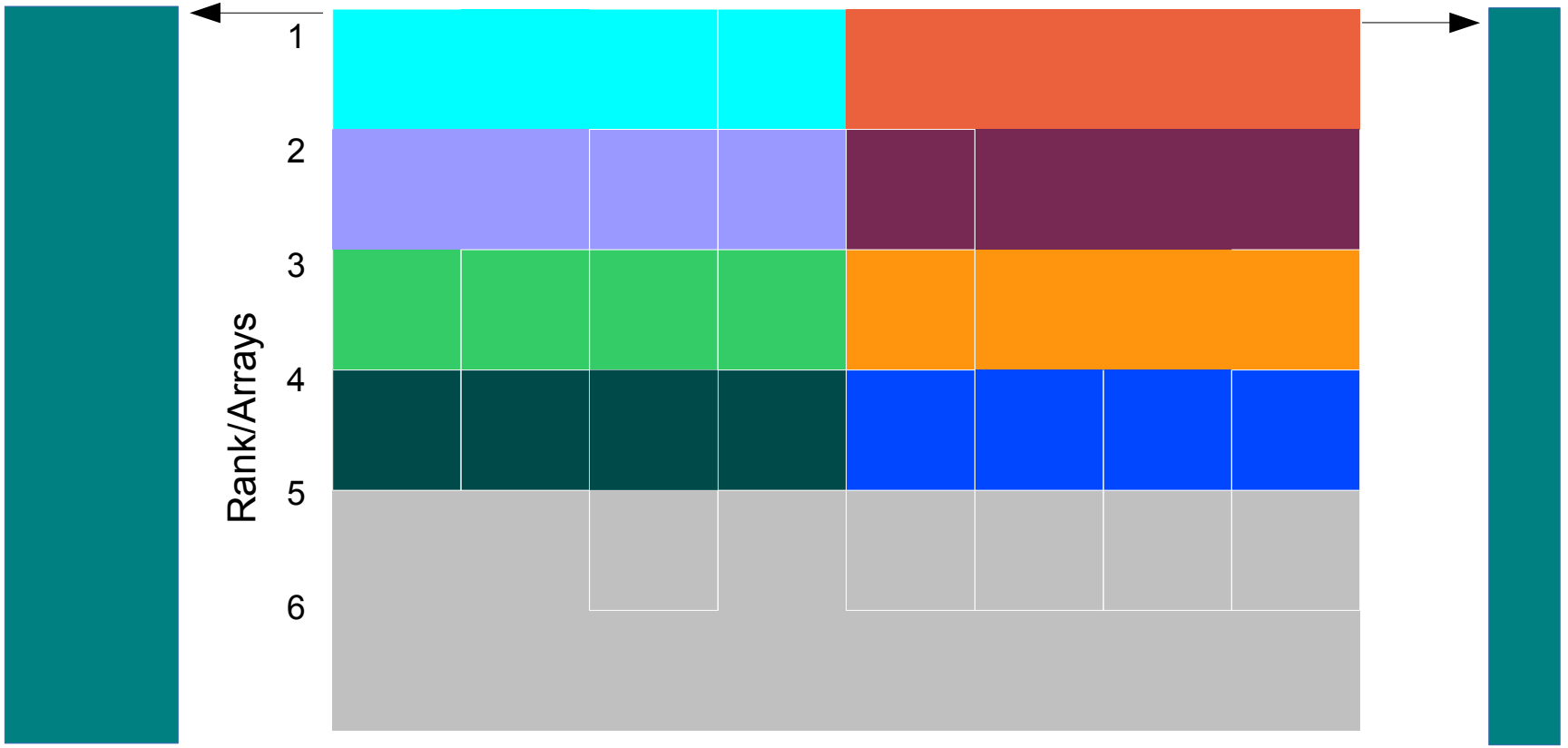
Disk I/O – Volumes with Storage Pool Striping (SPS)

- Extent pool example with 8 volumes each with 4 GiB and with Storage Pool Striping (SPS)
  - Each rank has access to an adequate portion of the read cache and non-volatile storage (NVS – write cache)
- Example: random access to one SPS volume
  - Usable portions of read cache and NVS much bigger because four ranks are involved
  - Up to four Device Adapters are in use



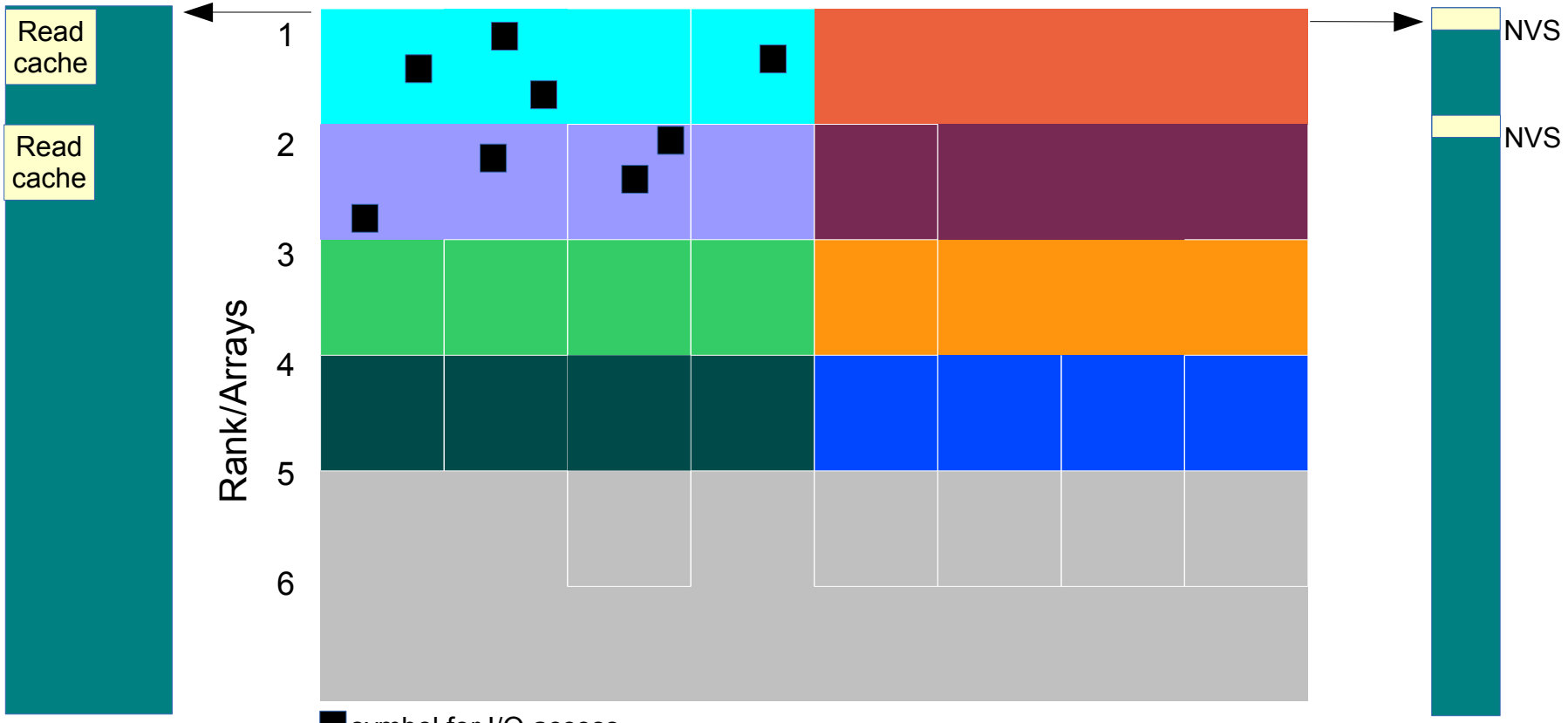
Disk I/O – two volumes in a striped logical volume (LV)

- Extent pool example with 8 volumes each of 4 GiB size
  - Each rank has access to an adequate portion of the read cache and non-volatile storage (NVS – write cache)
- Two volumes are used for the LV
  - Usable portions of read cache and NVS very limited because only two ranks are involved
  - Up to two Device Adapters are used for the connection to cache and NVS



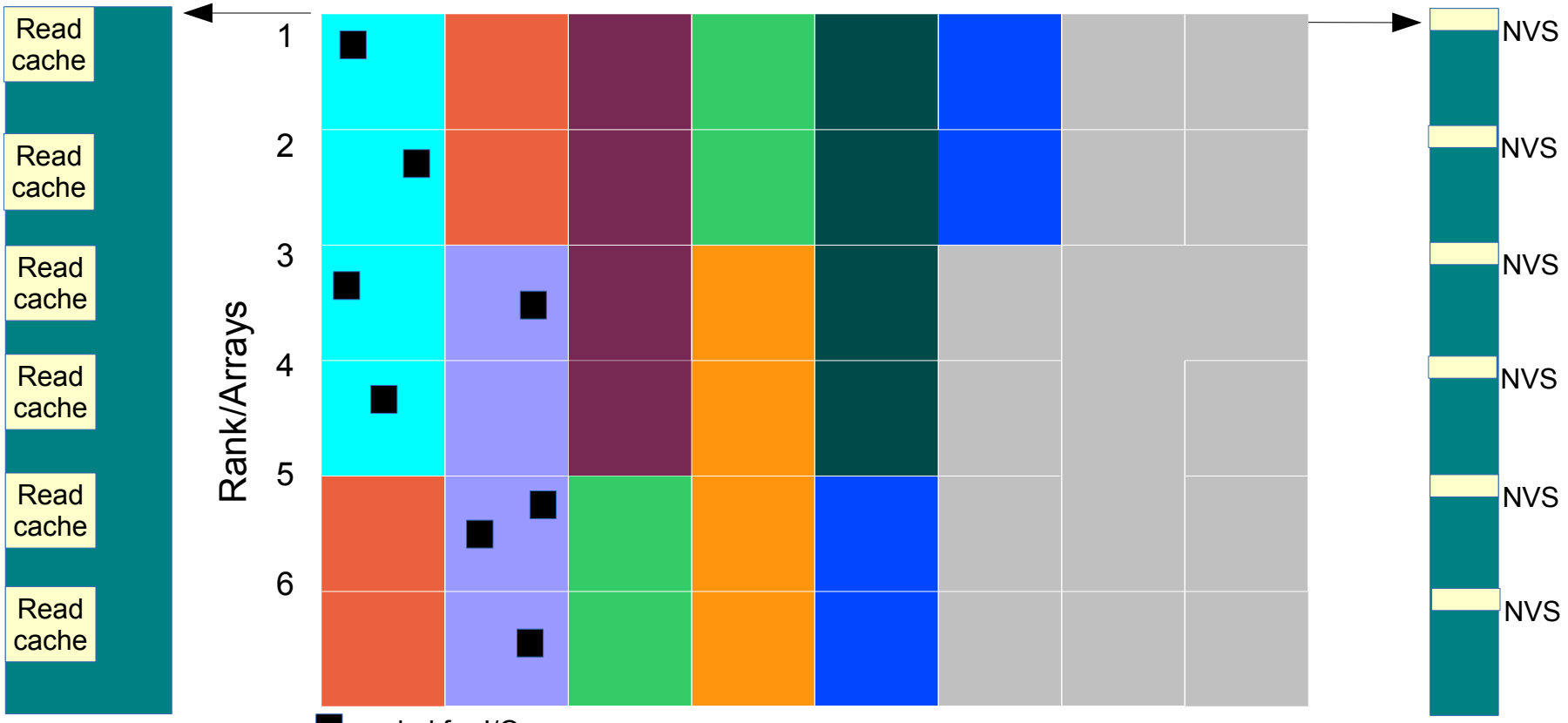
Disk I/O – two volumes in a striped logical volume (LV)

- Extent pool example with 8 volumes each of 4 GiB size
  - Each rank has access to an adequate portion of the read cache and non-volatile storage (NVS – write cache)
- Two volumes are used for the LV
  - Usable portions of read cache and NVS very limited because only two ranks are involved
  - Up to two Device Adapters are used for the connection to cache and NVS



Disk I/O – two SPS volumes in a striped LV

- Extent pool example with 8 volumes each with 4 GiB, with Storage Pool Striping (SPS)
  - Each rank has access to an adequate portion of the overall amount of read cache and non-volatile storage (NVS – write cache)
- Two SPS volumes are used for the LV
  - Usable portions of read cache and NVS much bigger because six ranks are involved
  - Up to six Device Adapters are used for the connection to cache and NVS



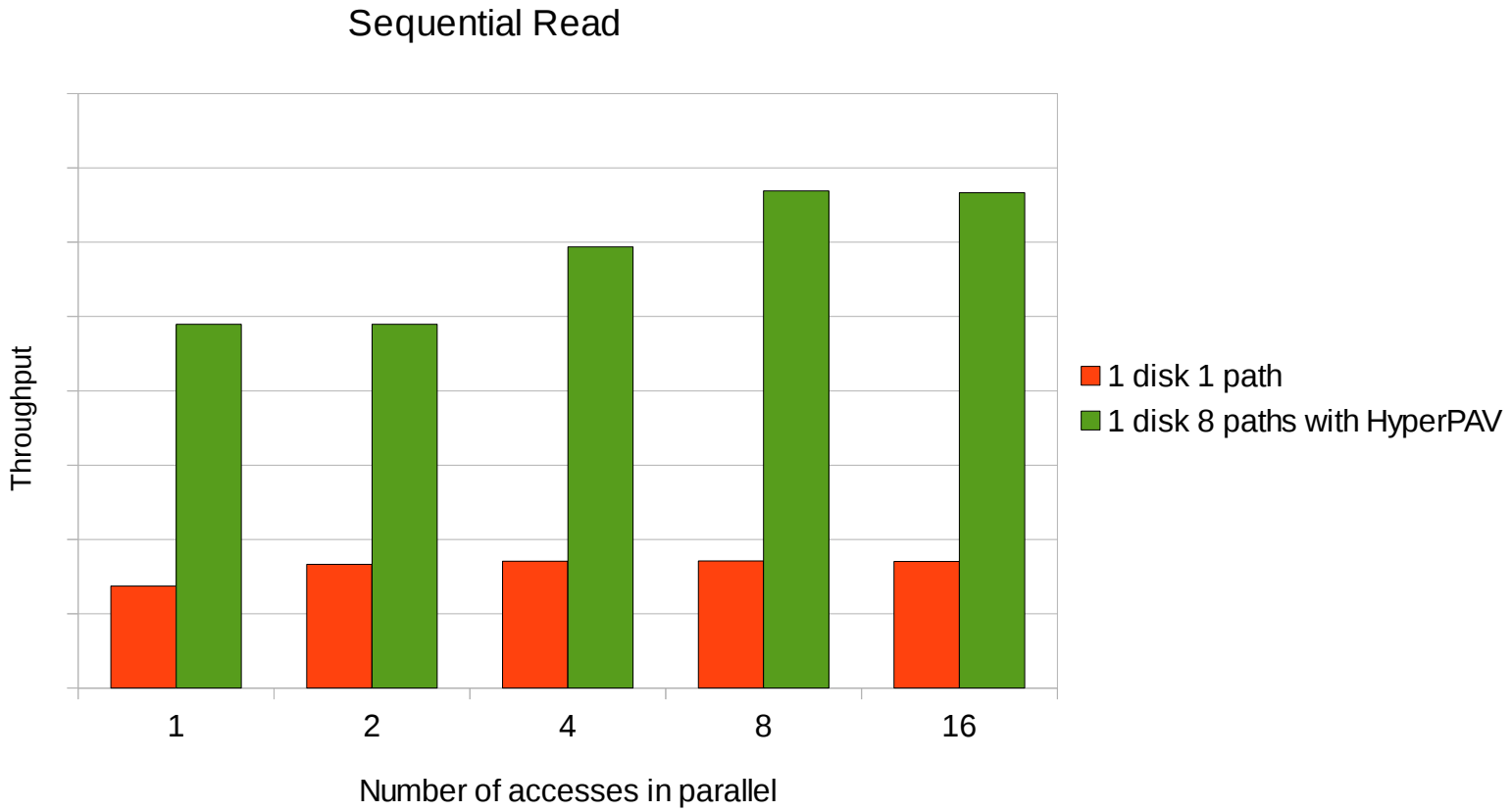
Disk I/O - striping options

- Striping is recommended and will result in higher throughput
  - Storage Pool Striped (SPS) volumes with linear LV will perform better when many disk I/O processes are involved ath the same time
  - Device mapper striping on SPS disks will have good performance with few disk I/O processes

	Storage Pool Striping (SPS)	Device mapper LV striping	Setup without striping
Performance improvement	yes	yes	no
Processor consumption in Linux	no	yes	no
Complexity of administration	low	high	no

Disk I/O FICON / ECKD – number of paths in use

- Comparison of I/O throughput to a single device (single subchannel) versus a single device with HyperPAV alias devices
  - Multiple (in example eight) paths perform much better
  - For reliable production systems you should use a multipath setup



Disk I/O FICON / ECKD – number of paths in use (cont.)

- iostat comparison (case 16 jobs in parallel)

single device (single subchannel)

```
...
04/10/14 23:52:20
Device:          rrqm/s   wrqm/s     r/s     w/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
dasda              0.00     0.20    0.00    0.20     0.00     1.60    16.00     0.00    0.00    0.00    0.00   0.00   0.00
dasdb              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dasdc           2830.60     0.00  750.60    0.00 340915.20     0.00   908.38    36.06   48.03   48.03    0.00   1.33 100.00
…
```

single device using HyperPAV alias devices

```
...
04/11/14 01:15:31
Device:          rrqm/s   wrqm/s     r/s     w/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
dasda              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dasdb              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dasdc          10243.20     0.00 2700.40    0.00 1229968.00     0.00   910.95    32.87   12.16   12.16    0.00   0.34  92.20
...
```

Disk I/O FICON / ECKD – number of paths in use (cont.)

- DASD statistics comparison (case 16 accesses in parallel)
- One channel program must be finished before the next can be executed in a no HyerPAV case
  - DASD driver queue size limited to maximal five entries
    - First table shows the distribution in statistics of one to five requests queued
- When more paths are used the requests gets distributed and parallel execution is possible
  - No more limitation to maximal five entries
    - Second table shows a distribution in statistics with up to seventeen requests queued
    - Most of the time eight to twelve requests queued

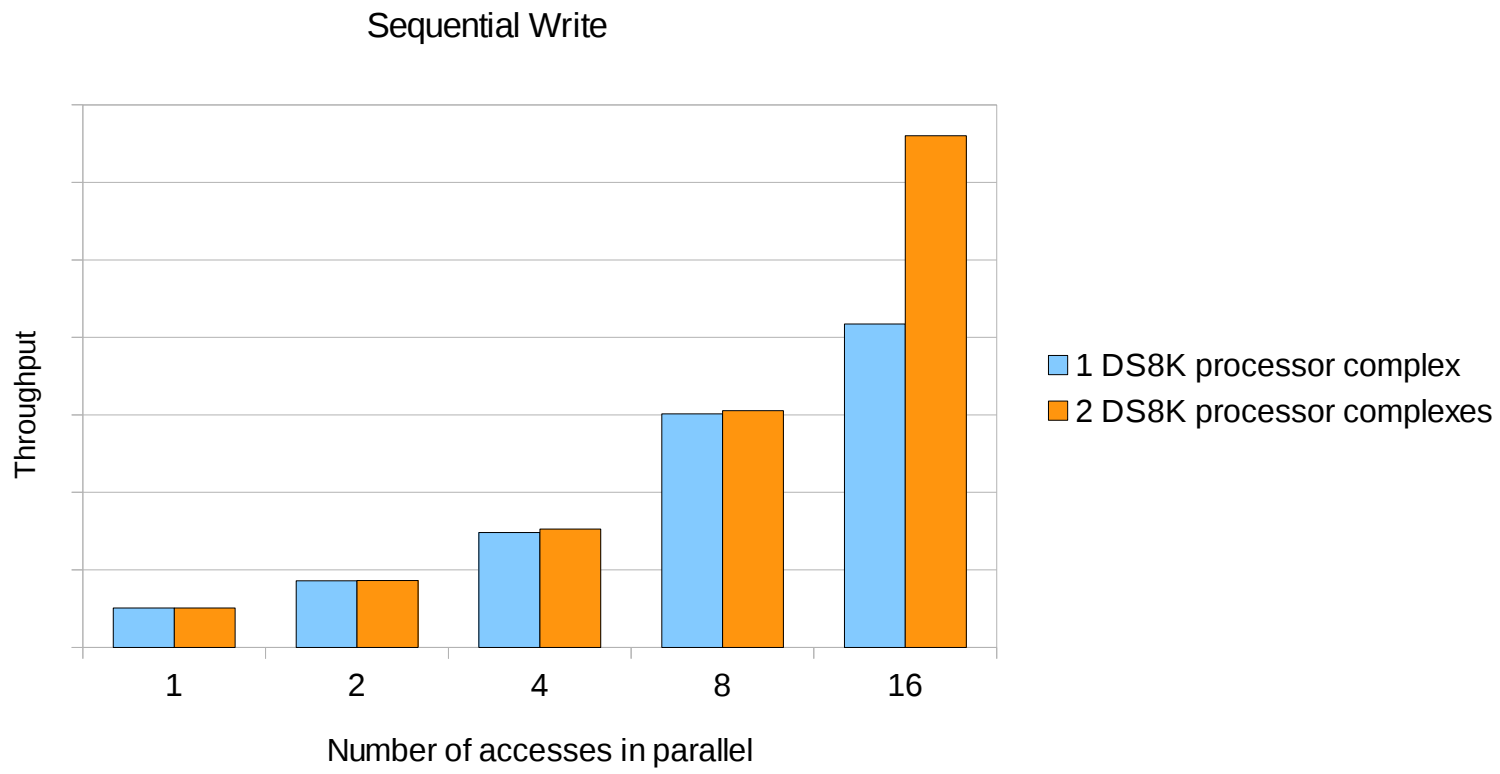
```
14513 dasd I/O requests
with 13108456 sectors(512B each)
Scale Factor is  1
  __<4    ___8    __16    __32    __64    _128    _256    _512    __1k    __2k    __4k    __8k    _16k    _32k    _64k    128k
  _256    _512    __1M    __2M    __4M    __8M    _16M    _32M    _64M    128M    256M    512M    __1G    __2G    __4G    _>4G


# of req in chanq at enqueuing (1..32)
     0      29    5396    7643    1445       0       0       0       0       0       0       0       0       0       0       0
     0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
```

```
...
# of req in chanq at enqueuing (1..32)
     0      14       8      28      95      85     181    1265    2958    3329    3755    1796     620     126      28      18
     9       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
...
```

Disk I/O FICON / ECKD – usage of DS8K processor complexes

- Comparison one DS8K processor complex versus both processor complexes with LVM and HyperPAV
  - Recommendation if throughput matters: distribute workload over both processor complexes
  - Write performance depends on available NVS



Disk I/O FICON / ECKD – usage of DS8K processor complexes (cont.)

- Run iostat using command "**`iostat -xtdk 10`**"
- iostat results for sequential write using one DS8K processor complex compared to both processor complexes (16 streams write in parallel)
  - Much more throughput for both processor complexes with more NVS available
  - Less await and service time with both processor complexes

1 DS8K processor complex

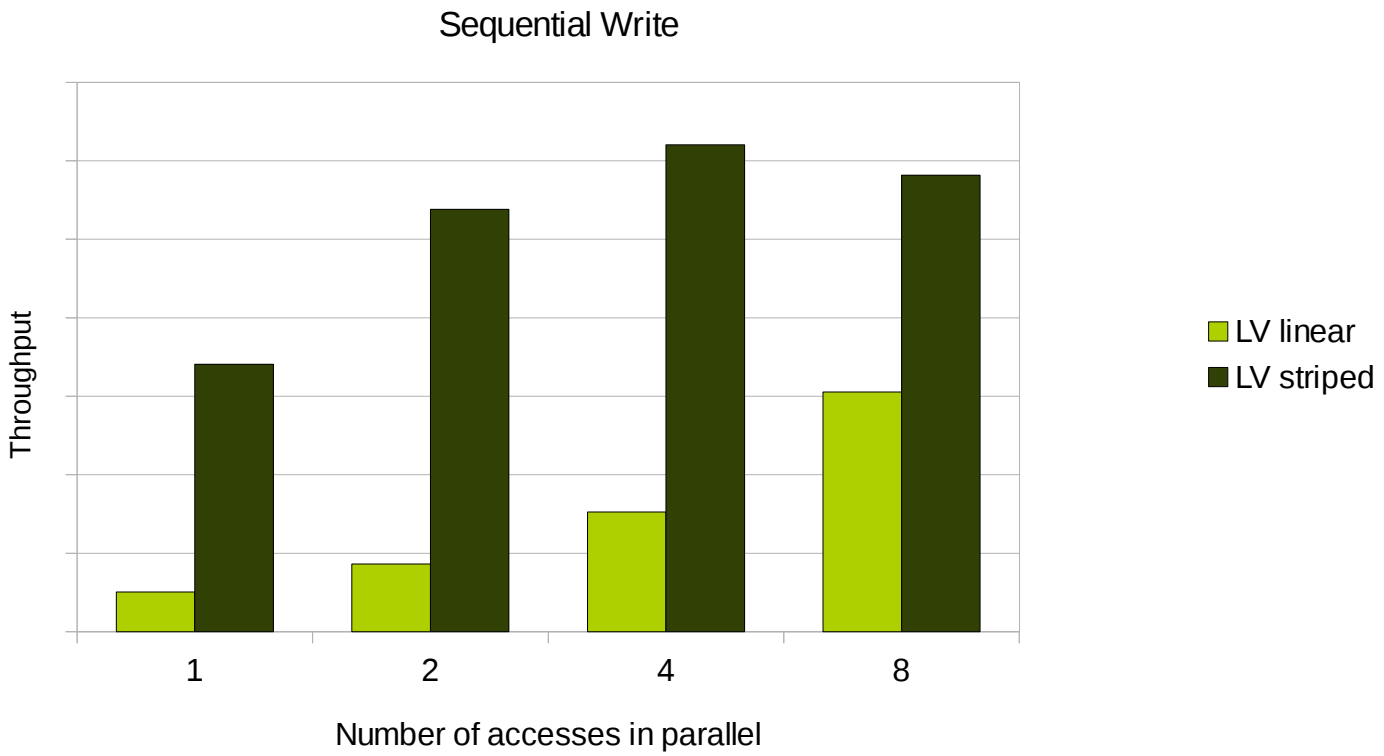
```
04/11/14 04:29:07
Device:         rrqm/s   wrqm/s     r/s     w/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
dasda             0.00     0.20    0.00    0.20     0.00     1.60    16.00     0.00    0.00    0.00    0.00   0.00   0.00
...
...
dasddz            0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dm-0              0.00     0.00    0.00 15577.60     0.00 1482777.60   190.37   139.00    9.41    0.00    9.41   0.06 100.00
...
```

2 DS8K processor complexes

```
04/11/14 20:58:22
Device:         rrqm/s   wrqm/s     r/s     w/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
dasda             0.00     0.00    0.00    0.20     0.00     0.80     8.00     0.00    0.00    0.00    0.00   0.00   0.00
...
...
dm-0              0.00     0.00    0.00 33563.60     0.00 3194752.00   190.37   161.00    4.80    0.00    4.80   0.03  98.60
```

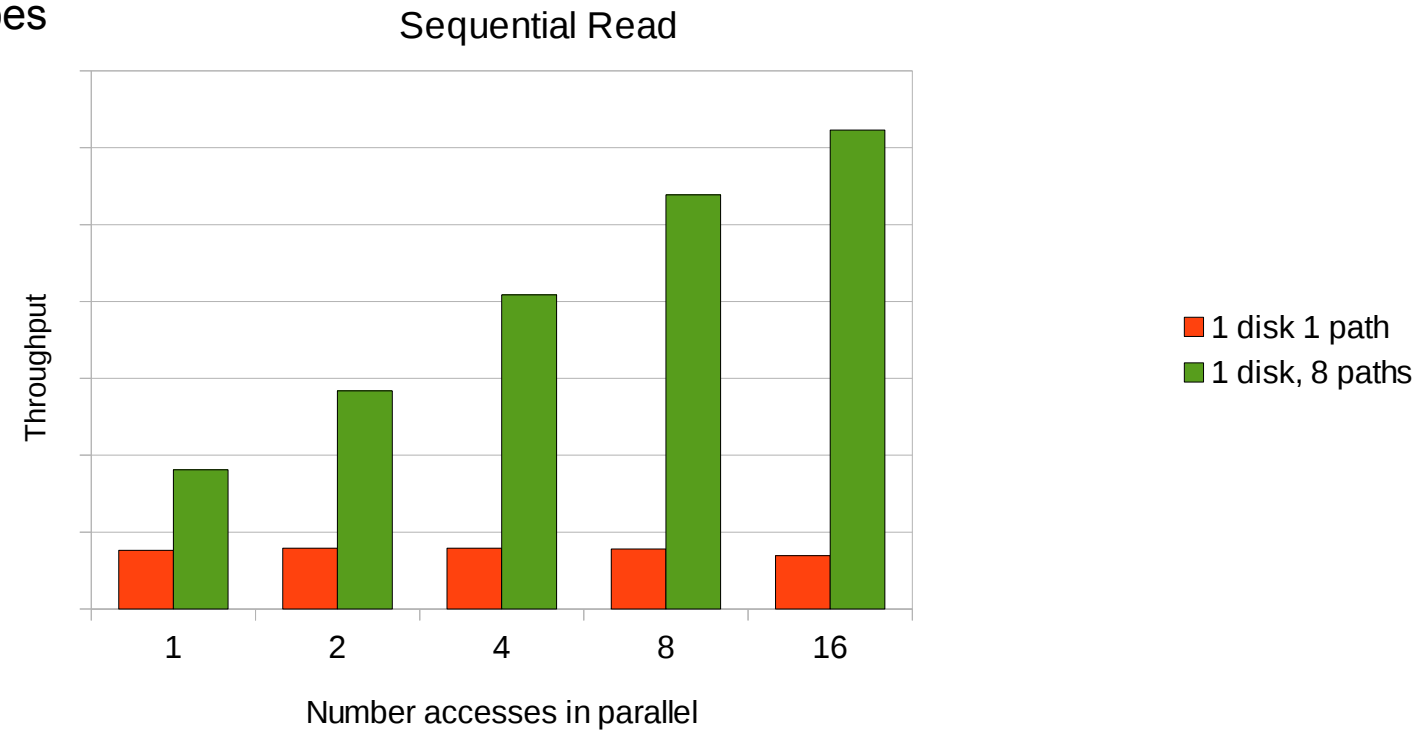
Disk I/O FICON / ECKD - LV linear versus LV striped

- Comparison Logical Volume linear versus Logical Volume striped
  - Much more parallelism when using striping with a few jobs running
  - Striping with sizes of 32KiB/64 KiB may split up single big I/Os (bad)
    - This applies especially to sequential workloads where read-ahead scaling take place
  - Striping adds extra effort / processor consumption to the system



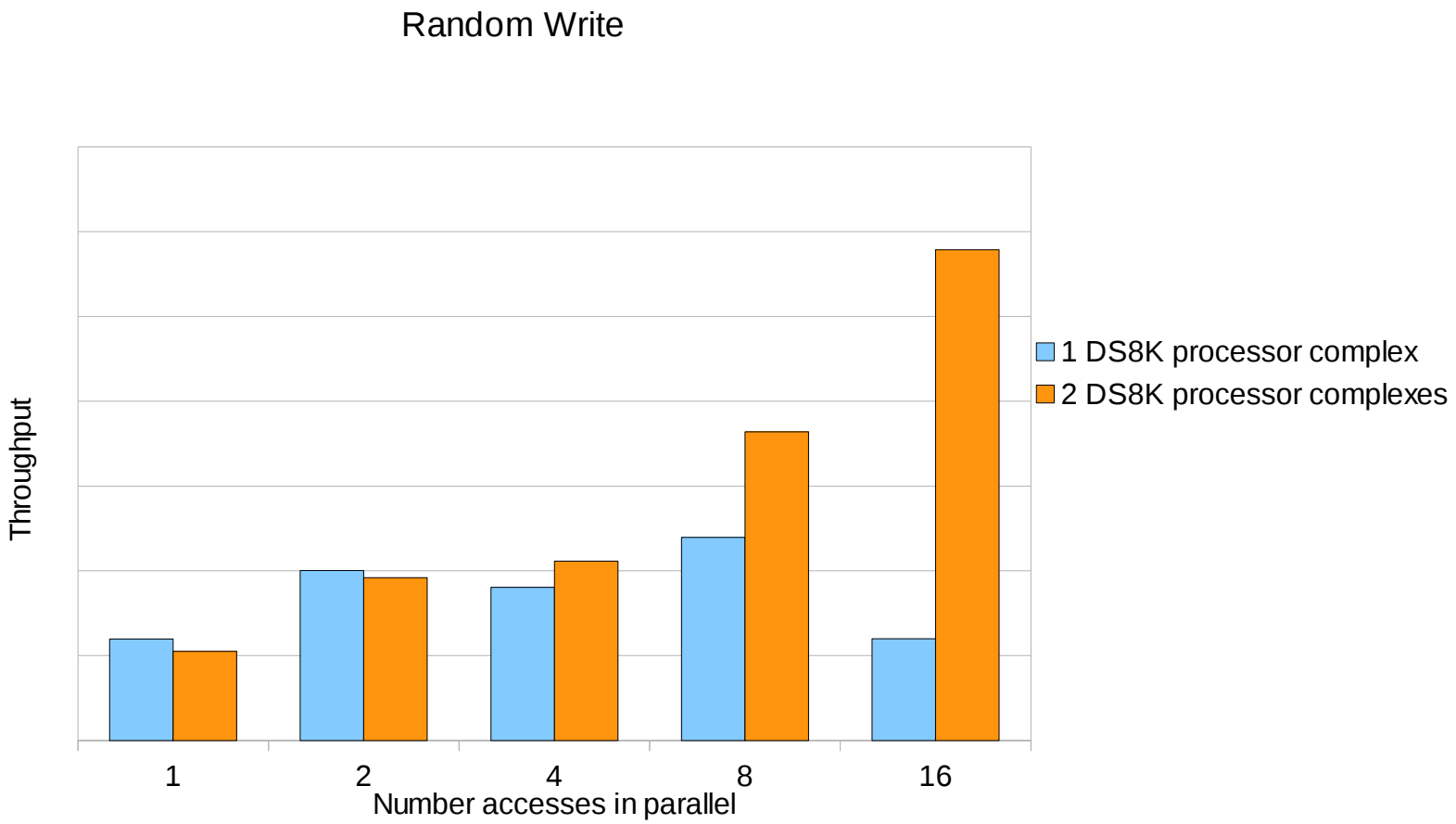
Disk I/O FCP / SCSI – number of paths in use

- Comparison single path setup to many paths
  - Multipath solution allows much more throughput
    - Multipath requires some extra processor cycles
  - Similar behavior to comparison single subchannel versus HyperPAV with ECKD / FICON
- **For reliable production systems you should use a multipath setup anyway**
  - Failover does not exploit the throughput capacity available of a path group, while multibus does



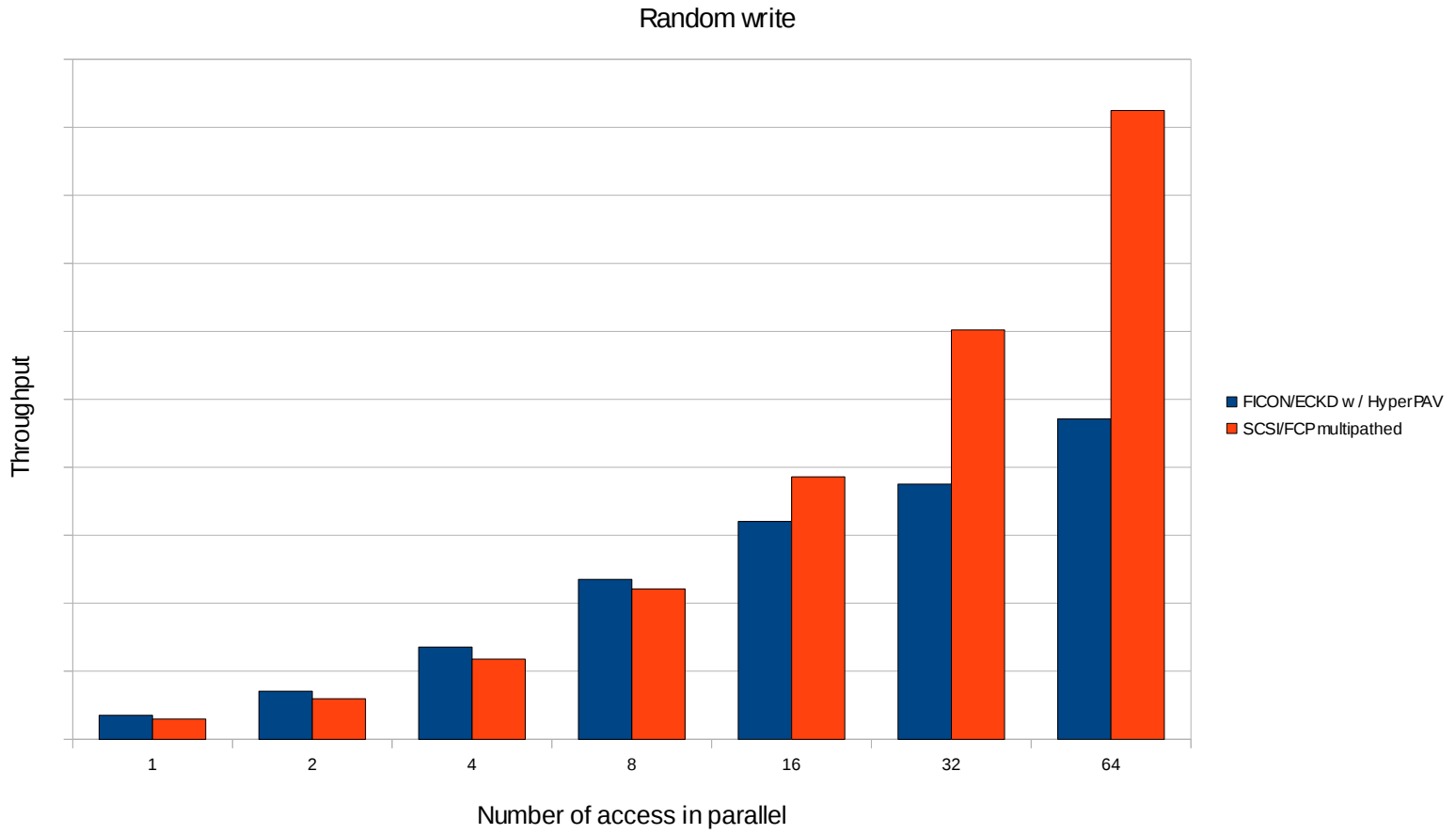
Disk I/O FCP / SCSI - usage of DS8K processor complexes

- Comparison usage of one processor complex versus both processor complexes with LVs
  - Usage of both processor complexes has an advantage if NVS became the limiting factor



Disk I/O FICON/ECKD vs SCSI/FCP

- Comparison of FICON/ECKD vs SCSI/FCP, in both cases a LV, using both storage server processor complexes, FICON/ECKD with HyperPAV and SCSI/FCP with device-mapper multipathing



Disk I/O – more tuning options

- Use latest hardware if throughput is important
  - Currently FICON Express 8S
- Use direct I/O and asynchronous I/O
  - Requires support by your used software products
  - More throughput at less processor consumption
  - In most cases advantageous if combined
- Use advanced FICON/ECKD techniques such as
  - High Performance FICON
  - Read Write Track Data
- Use the FCP/SCSI datarouter technique for further speedup (~5-15%)
  - Kernel parmline zfcp.datarouter=1, default “on” in more recent distribution releases
  - Requires 8S cards or newer
    - Feature similar to the store-forward architecture of recent OSA Cards
  - Allows the driver to avoid extra buffering in the card
    - No in card buffering also means there can't be a stalling buffer shortage

Disk I/O – performance considerations summary

- Use as much paths as possible
  - ECKD logical path groups combined with HyperPAV
  - SCSI Linux multipath multibus
- Use all advanced software, driver and Hardware features
- Storage Server
  - Use Storage Pool Striping (SPS) as a convenient tool
  - Define extent pools spanning over many ranks
  - Use both storage server complexes of the storage server (DS8K)
- If you use Logical Volumes (LV)
  - Linear: with SPS and random access
  - Linear: with SPS and sequential access and many processes
  - Striped: for special setups that proved to be superior to SPS

- **So long story short – focus on your needs and optimize wisely**

Questions ?

- Further information is located at
  - Linux on System z – Tuning hints and tips
    http://www.ibm.com/developerworks/linux/linux390/perf/index.html
  - Live Virtual Classes for z/VM and Linux
    http://www.vm.ibm.com/education/lvc/

Appendix: Used Hardware & Distro

- Hardware
  - z196 and zEC12 LPARs
- Storage Server
  - DS8870 (1-64 disks involved in test cases)
- SAN Adapter
  - FICON Express 8S – 8 paths
- Distro's
  - RHEL7
  - SLES11 SP3