L79

Linux Filesystems

Klaus Bergmann
Linux Architecture & Performance
IBM Lab Boeblingen

zSeries Expo, November 10 -14, 2003 | Hilton, Las Vegas, NV

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

- Journaling file systems
- Measurement setup
- Measurement results
  - LPAR – VM
  - 31 / 64 bit
  - single disk and LVM
  - DASD statistics
  - CPU load and CP overhead
  - journaling options
- Outlook

Problems of non-journaling file systems

- data and meta-data is written directly and in arbitrary order
- no algorithm to ensure data integrity
- after crash, complete structure of file system has to be checked to ensure integrity
- file system check times depend on size of file system

→ risk of data loss

→ long and costly system outages

advantages of journaling

- data integrity ensured
- in case of system crash only journal has to be replayed to recover consistent file system structure
- file system check time depends on size of journal

i much higher data integrity

i much shorter system outages

- but there is a cost...

Journaling file systems in SuSE SLES8

- ext3 v0.9.18
- jfs 1.0.24
- reiserfs 3.6.2

For reference :

- ext2 v0.5 (non-journaling)

ext3

- developed by Andrew Morton and others
- based on ext2
- extended by journaling features
- supports full data journaling
- resizing (only with unmount) possible
- http://www.zipworld.com.au/~akpm/linux/ext3/

jfs

- developed by IBM Austin Lab
- ported from OS/2 Warp Server
- only metadata journaling
- max. file system size 4 PB
- http://www.ibm.com/developerworks/oss/jfs/index.html

reiserfs

- developed by a group around Hans Reiser
- SUSE's default choice
- only metadata journaling
- disk space optimization algorithm
- online enlargement of file system
- http://www.namesys.com/

Measurement setup

Hardware

- **2064-216 (z900)**
  - 1.09ns (917MHz)
  - 2 * 16 MB L2 Cache (shared)
  - 64 GB
  - 6 FICON channels
- **2105-F20 (Shark)**
  - 384 MB NVS
  - 16 GB Cache
  - 128 * 36 GB disks
  - 10.000 RPM
  - FICON (1 Gbps)

Software

- **SUSE SLES8**
- **Dbench**

Measurement setup

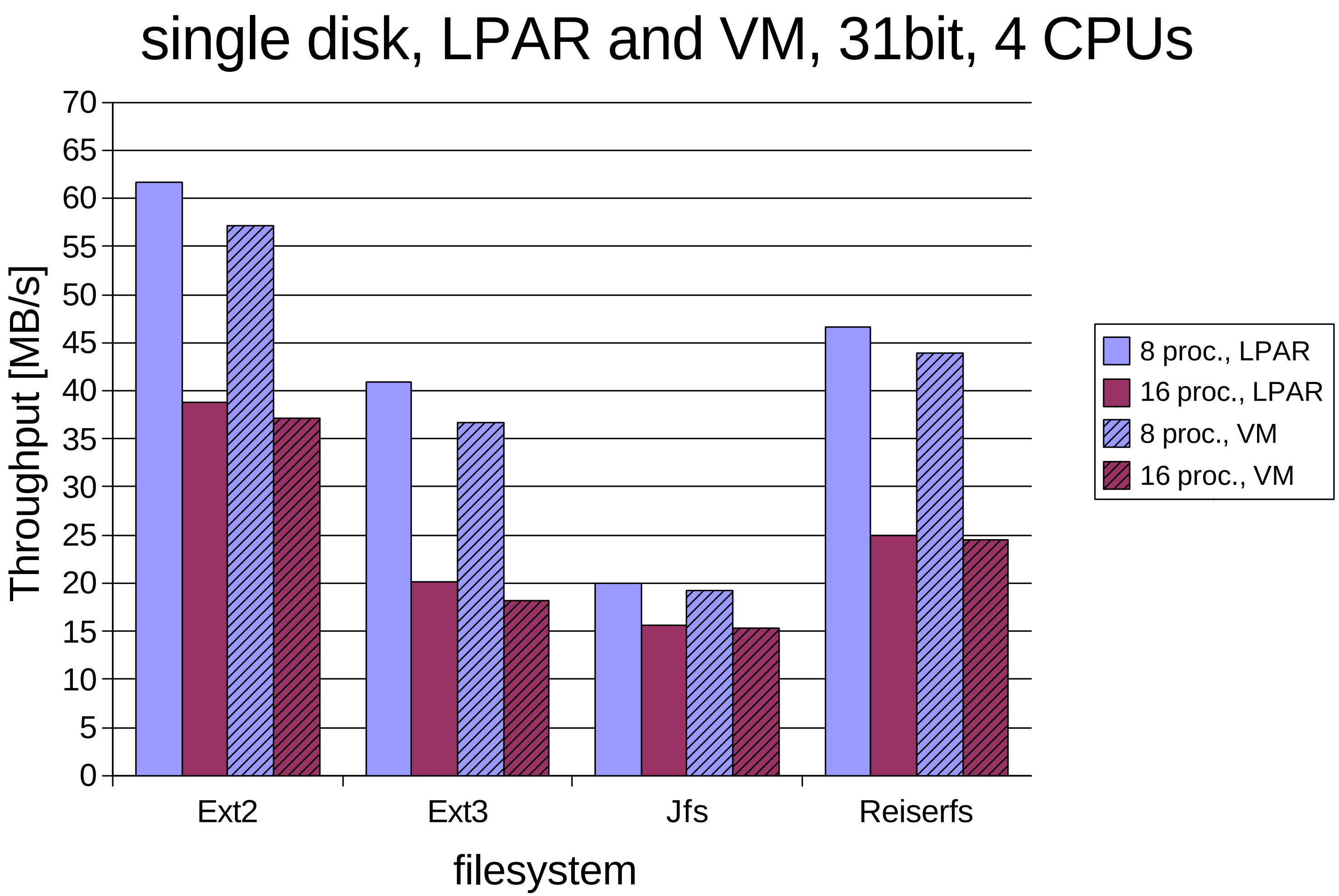
- dbench
- 128MB main memory
- 1, 2 and 4 CPUs
- LPAR and z/VM 4.3
- 31-bit and 64-bit
- Single 3390 model 3 disk
- 6 pack of 3390-3 using striped LVM. Attached via 6 FICON channels
- Running 8 and 16 processes

Dbench File I/O

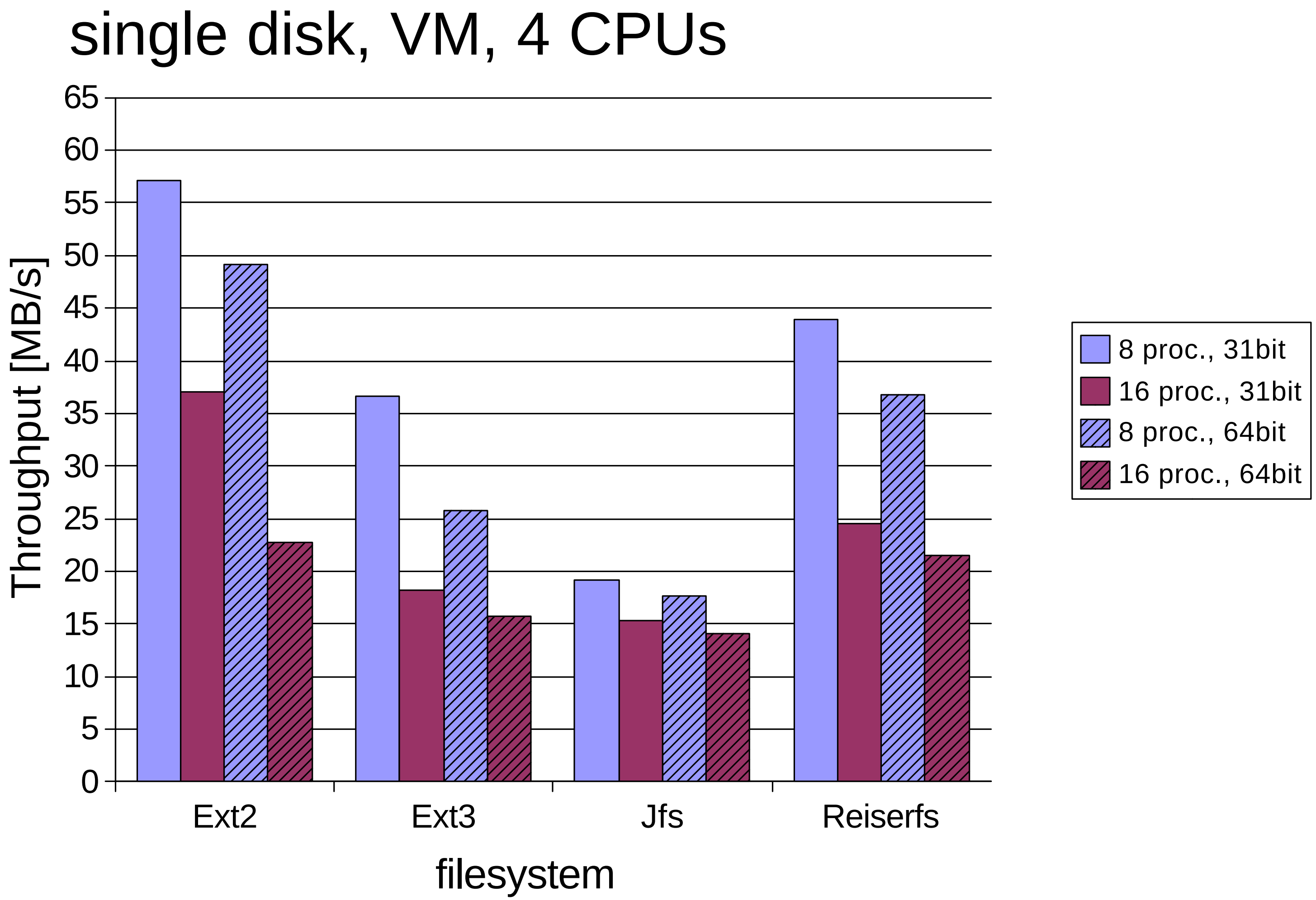
- Emulation of Netbench benchmark, rates windows fileservers
- Large set of mixed file operations workload for each process: create, write, read, append, delete
  - Scaling for Linux with 1, 2, 4 PUs
  - Scaling for 8 and 16 clients (processes) simultaneously
- forced to do I/O while memory is filling up with data

Measurement results

LPAR and VM


single disk, LPAR and VM, 31bit, 4 CPUs
Throughput [MB/s]
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
Ext2
Ext3
Jfs
Reiserfs
filesystem
8 proc., LPAR
16 proc., LPAR
8 proc., VM
16 proc., VM

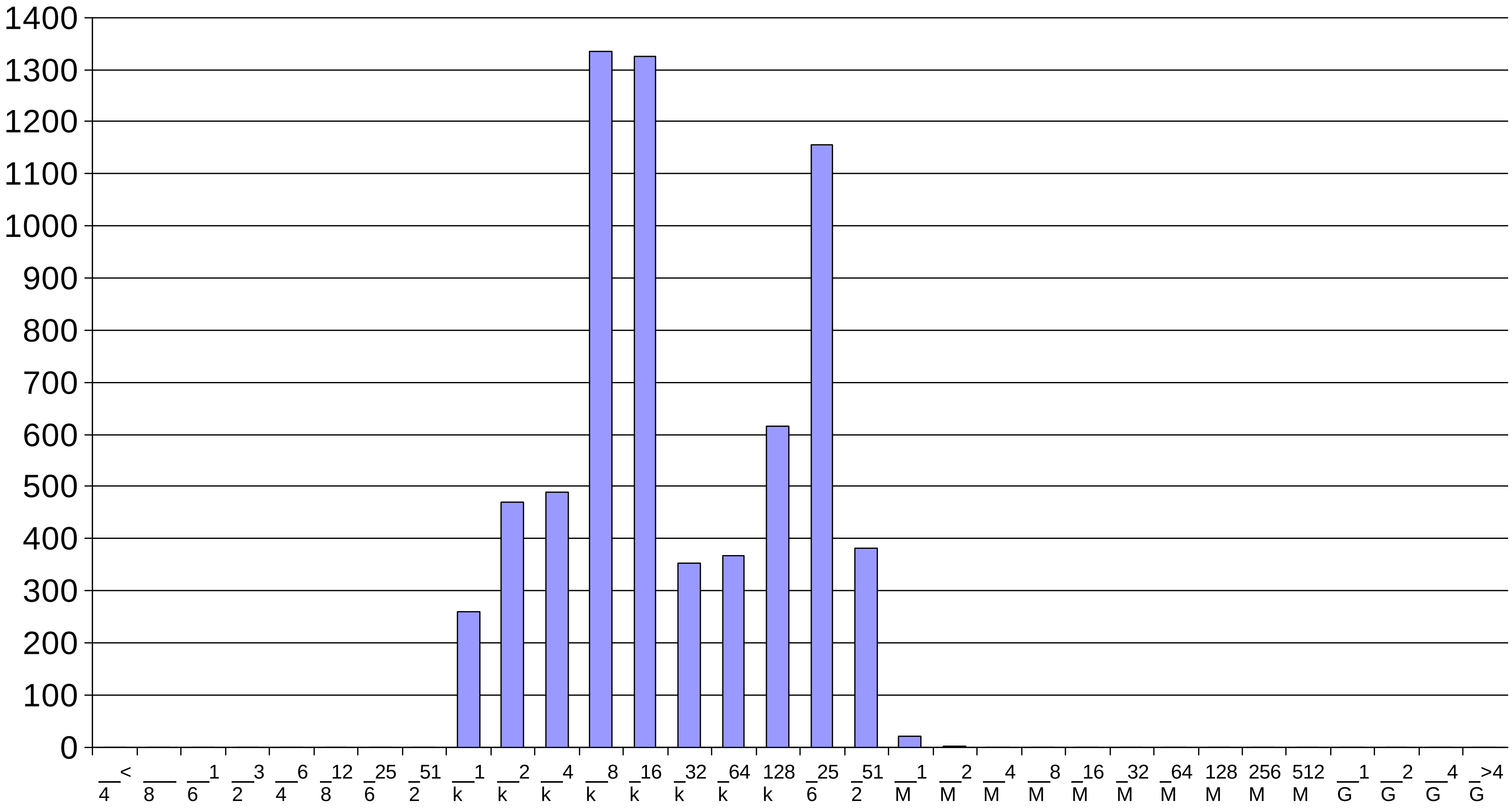
31-bit and 64-bit



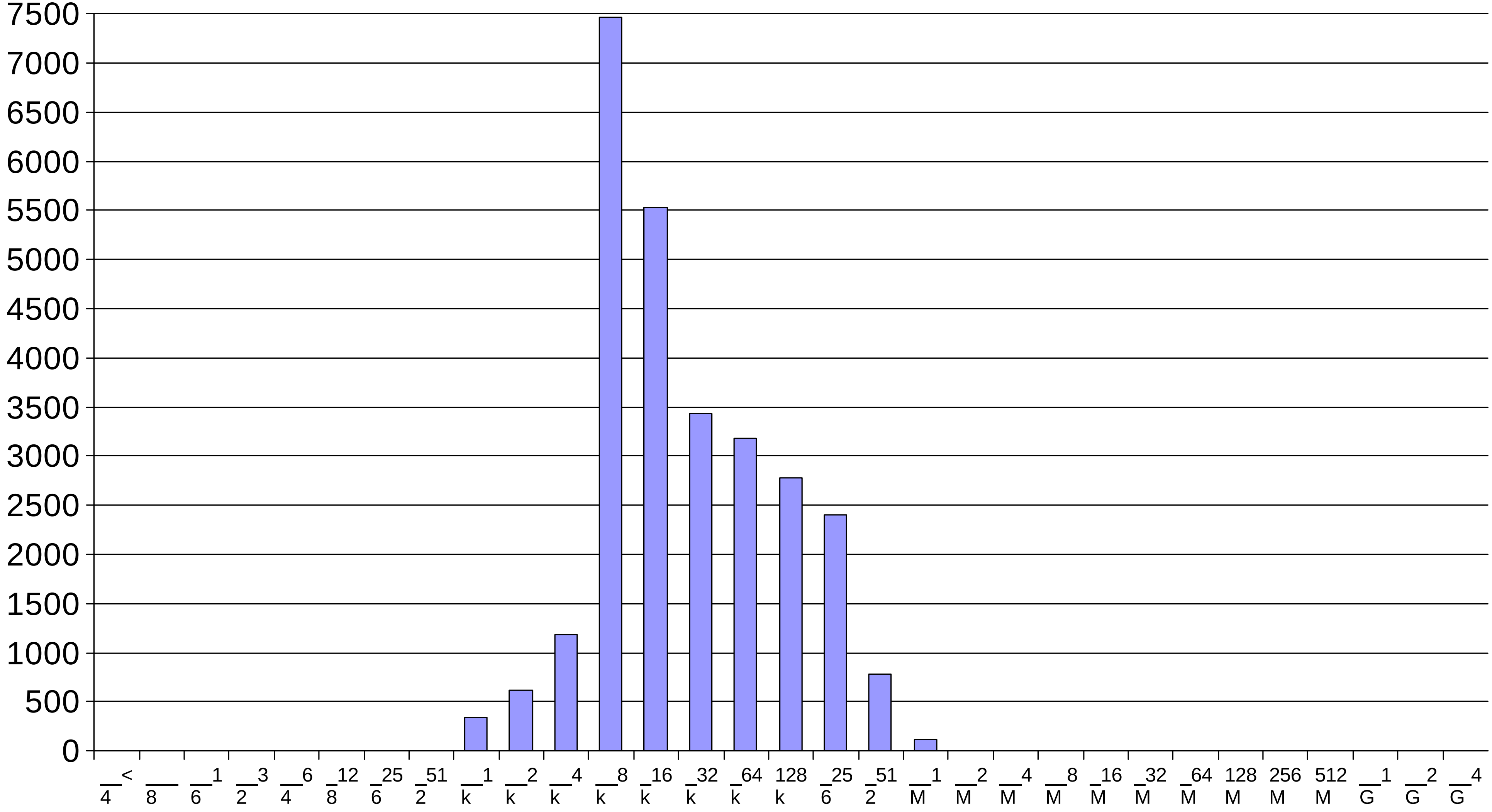
/proc/dasd/statistics – Example

```
root@g73vm1:~# cat /proc/dasd/statistics
56881 dasd I/O requests
with 5270816 sectors(512B each)
   __<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
Histogram of sizes (512B secs)
      0       0    1039    4799    8102   36557    4475     292     195    1422       0       0       0       0       0       0
      0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
Histogram of I/O times (microseconds)
      0       0       0       0       0       0       0       0       2       8     109    3244   25570   17480    7666    1248
   1390     153      11       0       0       0       0       0       0       0       0       0       0       0       0       0
Histogram of I/O times per sector
      0       0       0       0     176    4141   24084   15639    9506    2513     601     173      41       7       0       0
      0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
Histogram of I/O time till ssch
      5       1       2       0       0       0       0       0       2       4     301   11527   25339   12278    5156    1759
    383     118       6       0       0       0       0       0       0       0       0       0       0       0       0       0
Histogram of I/O time between ssch and irq
      0       0       0       0       0       0       0       0    2584   23896   18720    5307    2325    2725    1217      62
     23      21       1       0       0       0       0       0       0       0       0       0       0       0       0       0
Histogram of I/O time between ssch and irq per sector
      0       0       0   21722   26243    3939    2184    1798     774     159      47      12       3       0       0       0
      0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
Histogram of I/O time between irq and end
      7       0   43393   11341     457     179    1494       3       3       1       2       1       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)
      8       3       4       5   56861       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
```

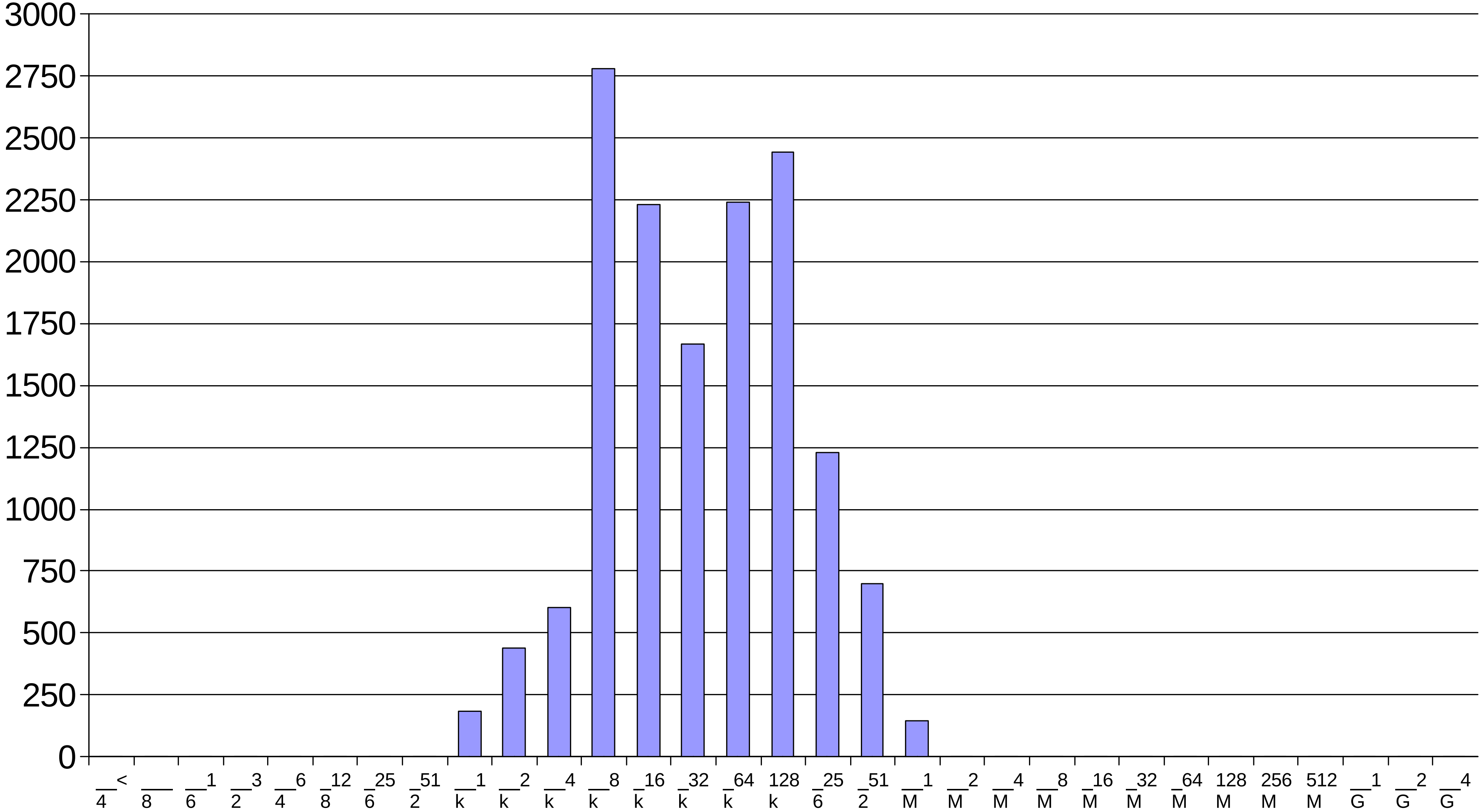
ext2, 8 Processes
Histogram of I/O times (microseconds)



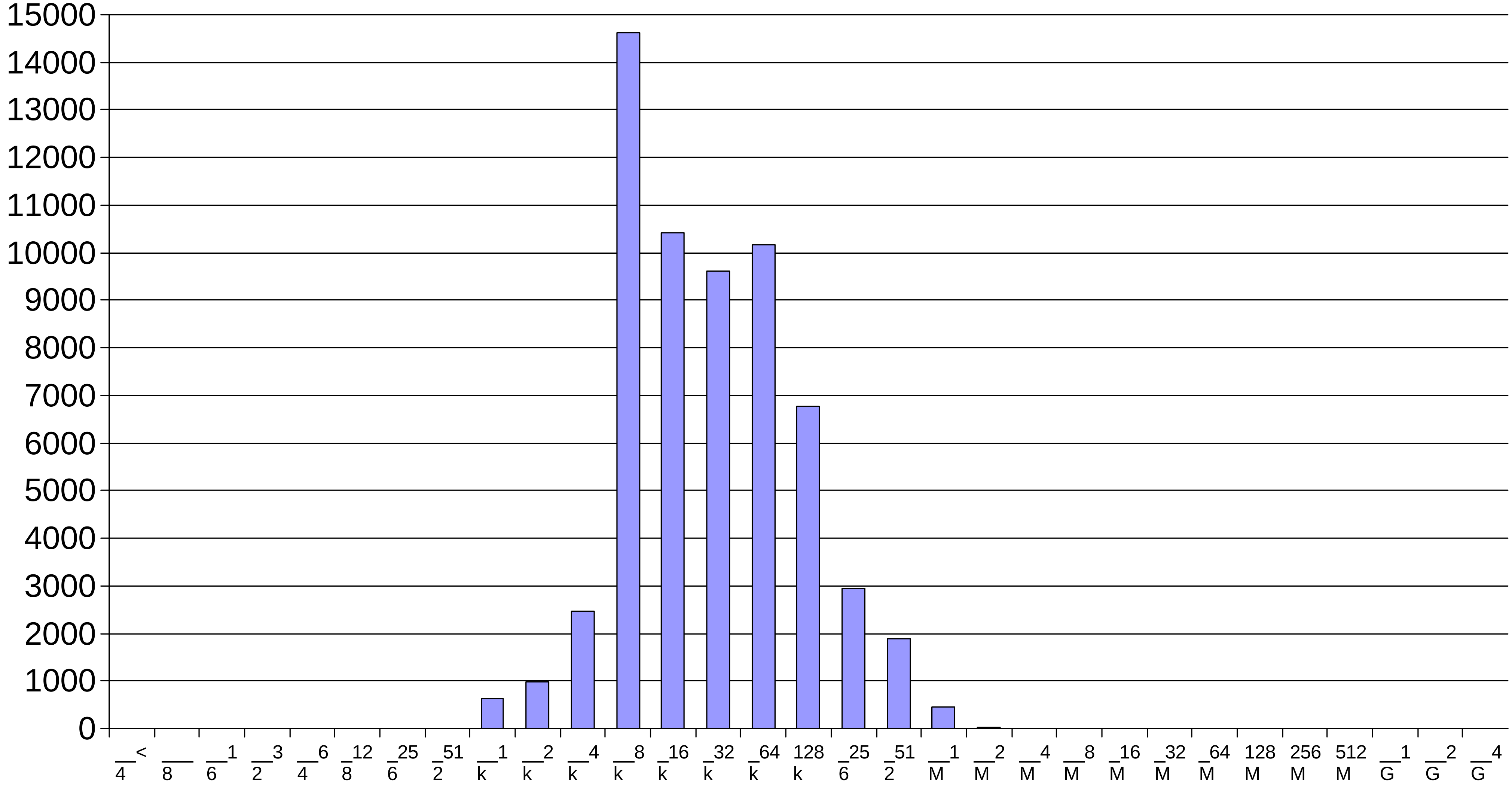
ext2, 16 Proceses
Histogram of I/O times (microseconds)



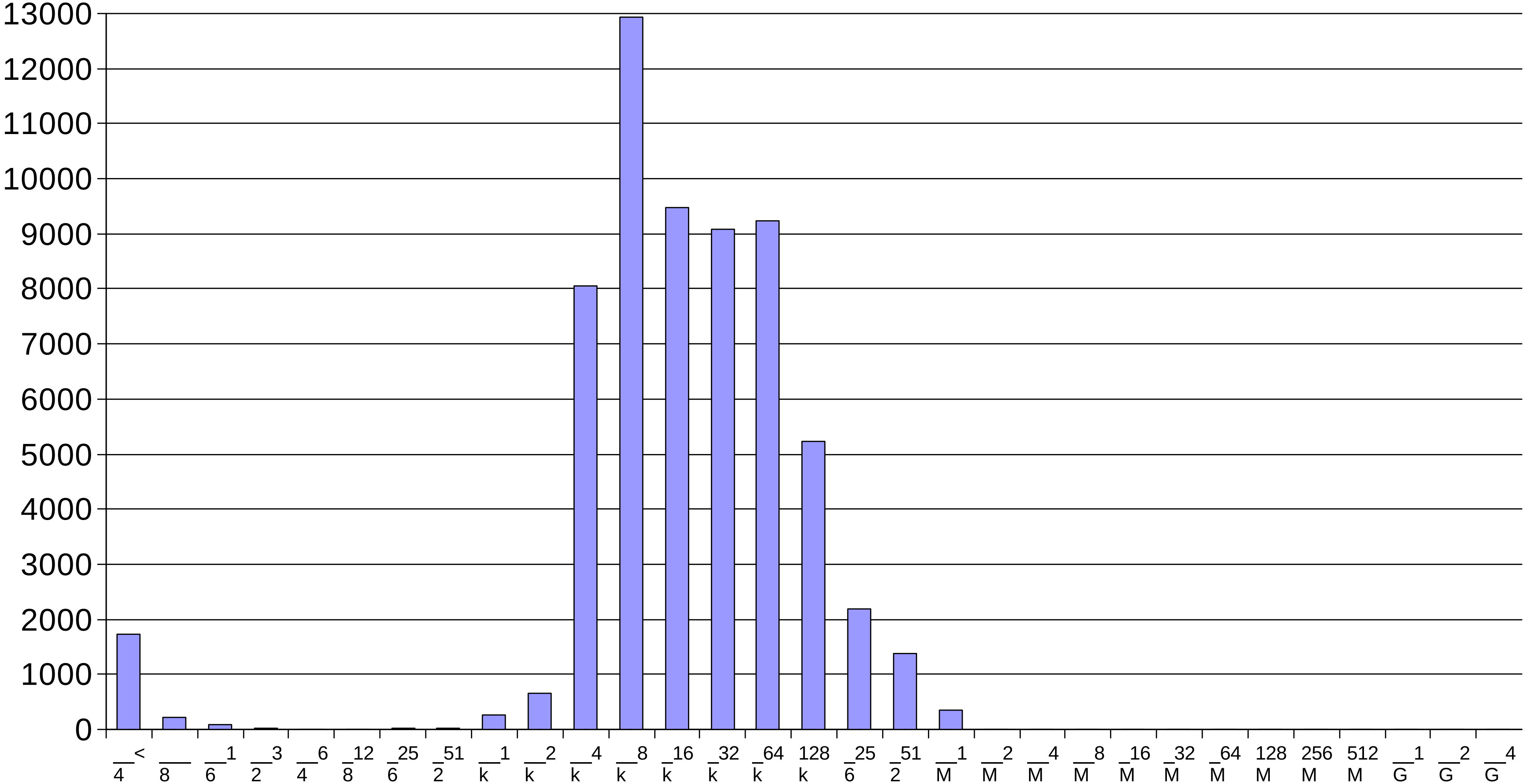
ext3, 8 Processes
Histogram of I/O times (microseconds)



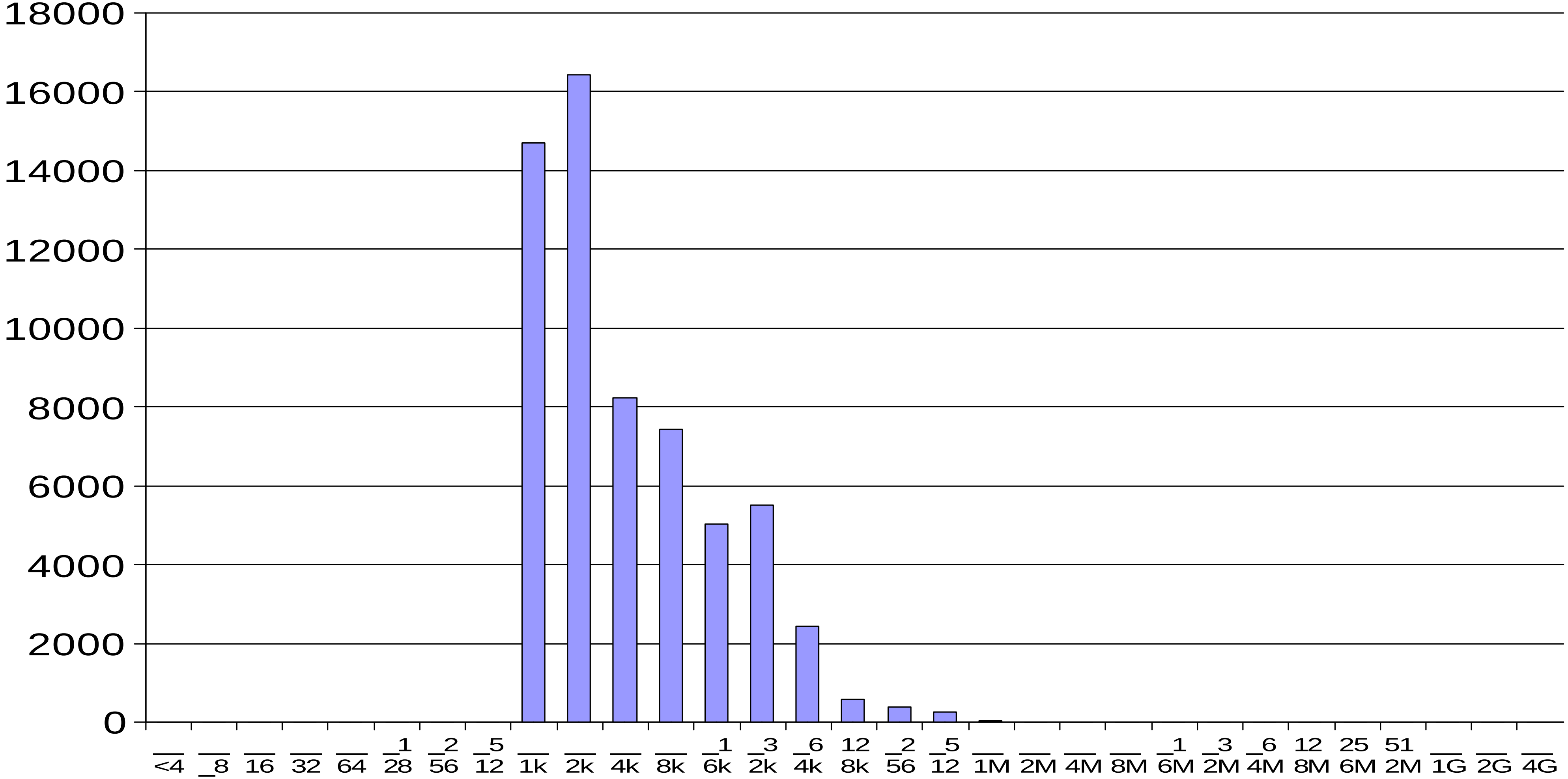
ext3, 16 Processes
Histogram of I/O times (microseconds)



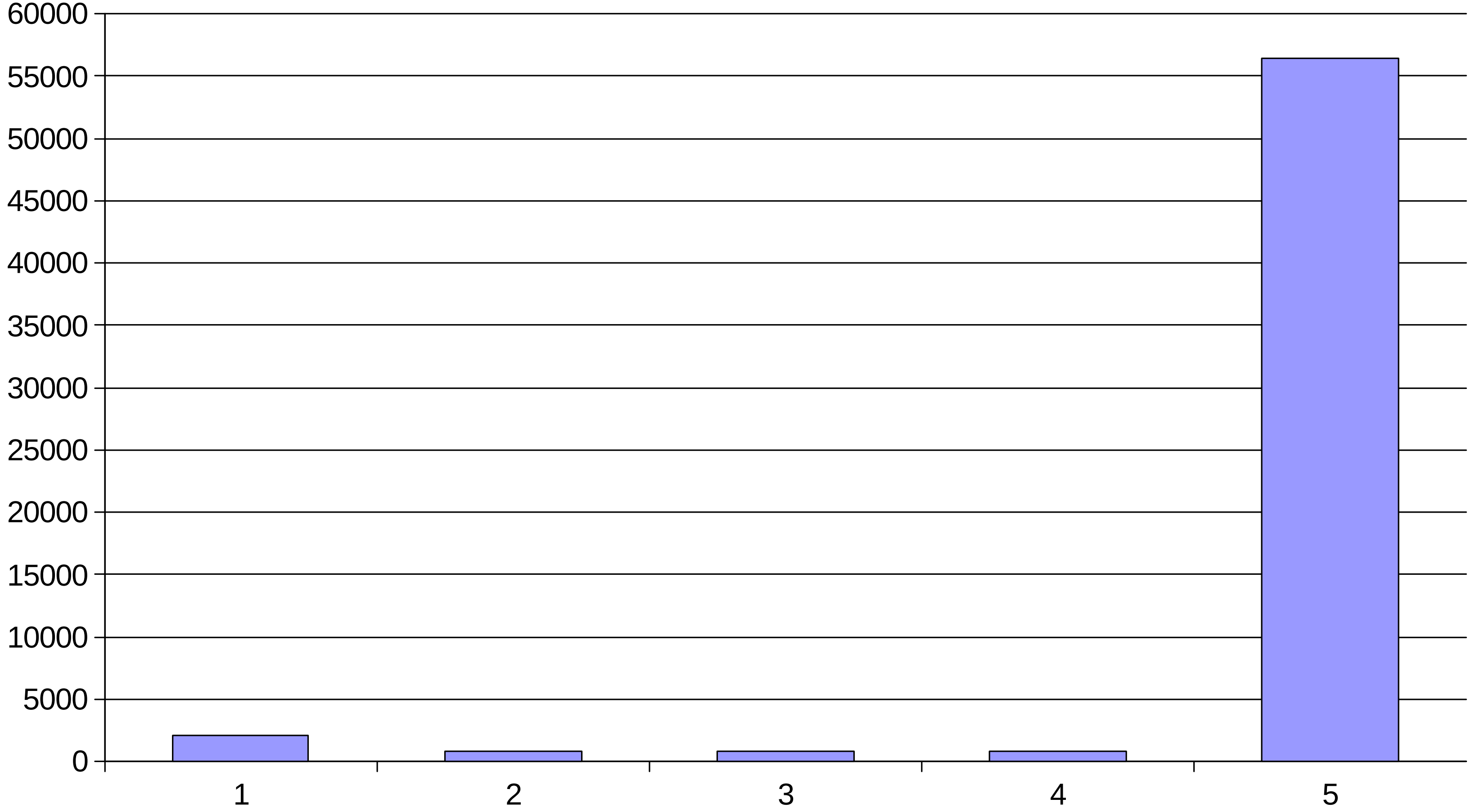
ext3, 16 Processes
Histogram of I/O time before SSCH (IOSQ)



Ext3, 16 Processes
Histogram of I/O time between SSCH and IRQ



Ext3, 16 Processes
number of requests in subchannel-queue at enqueuing

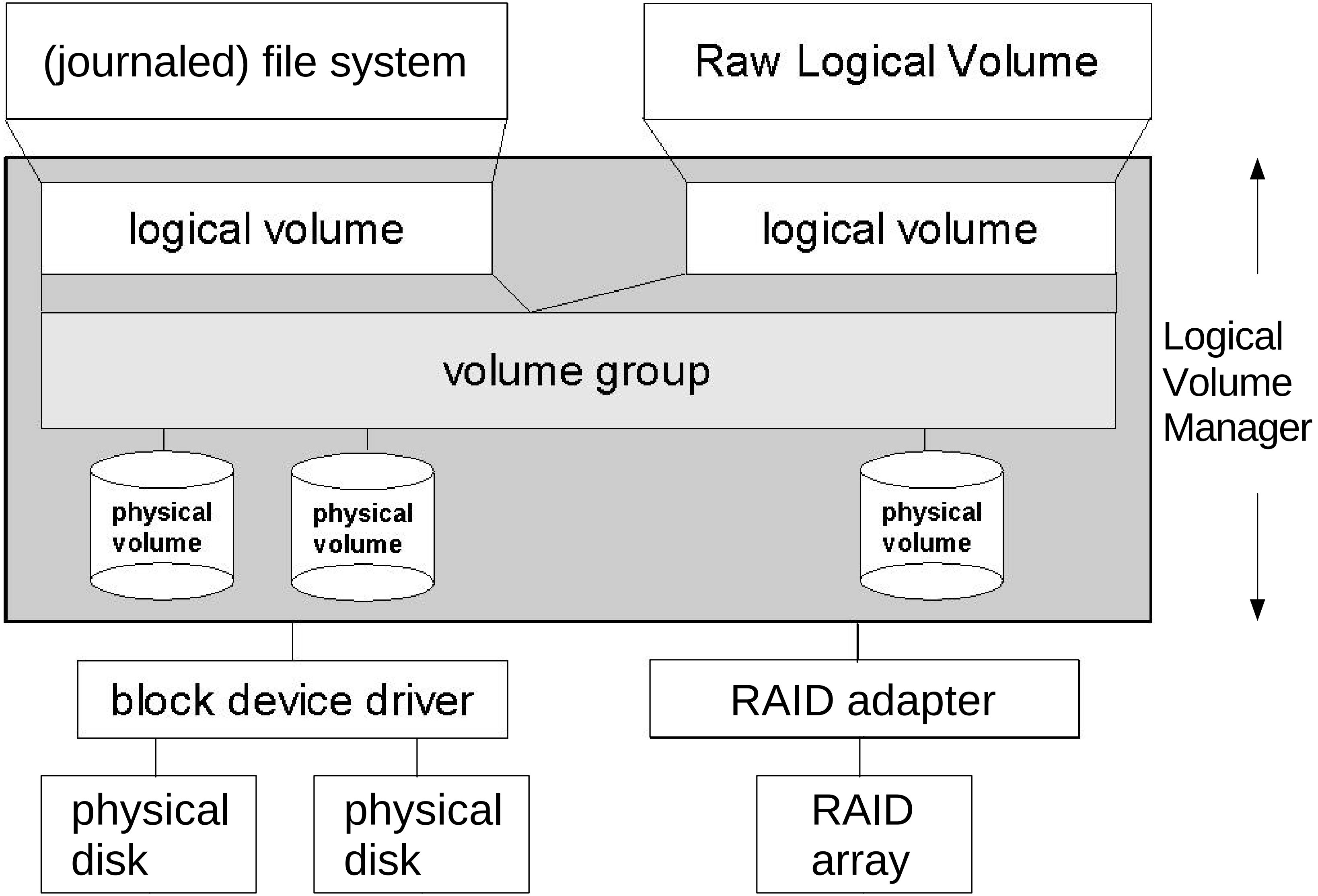


Logical Volume Manager (LVM)

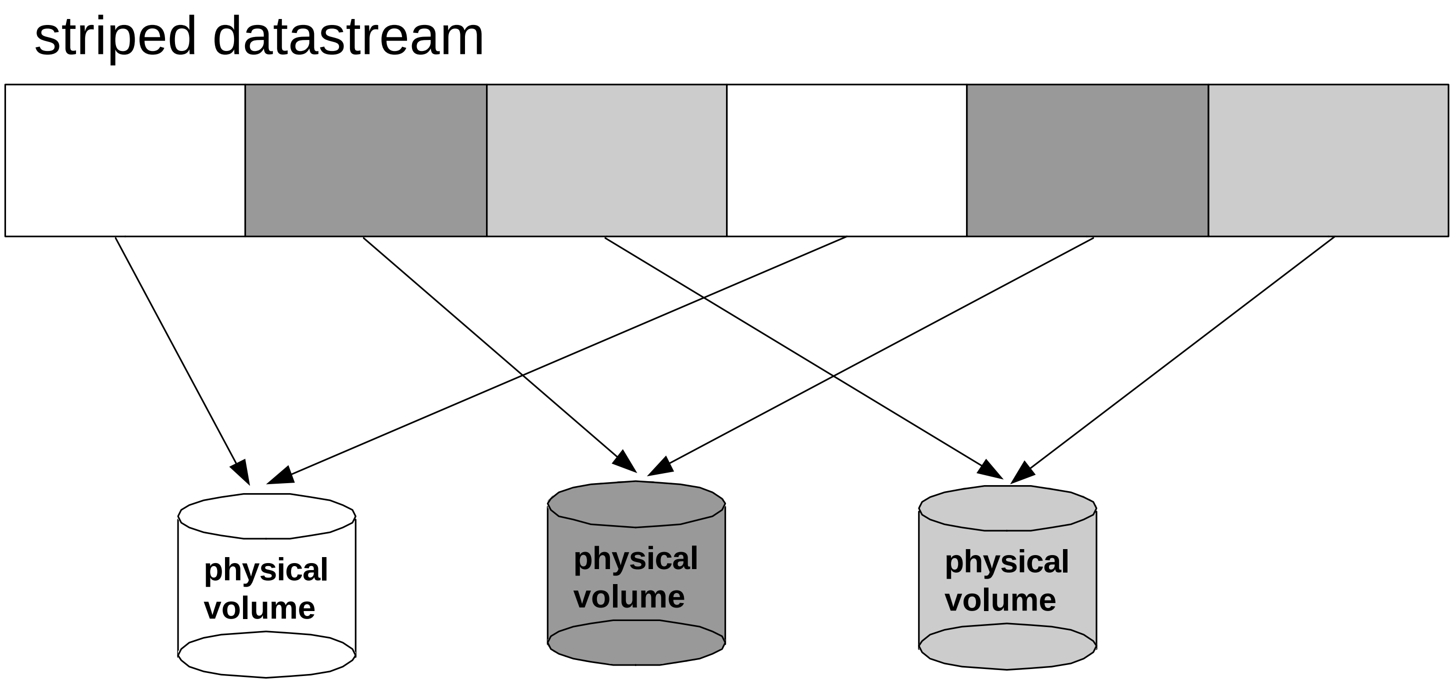
- Linux software raid with raid levels 0,1, 4 and 5
- excellent performance
- excellent flexibility (resizing, adding/removing disks)
- available in SLES7, SLES8, and RedHat RHEL 3
- on zSeries, support multipath and PAV (under z/VM)
- http://www.sistina.com/products_lvm.htm

SISTINA

LVM system structure

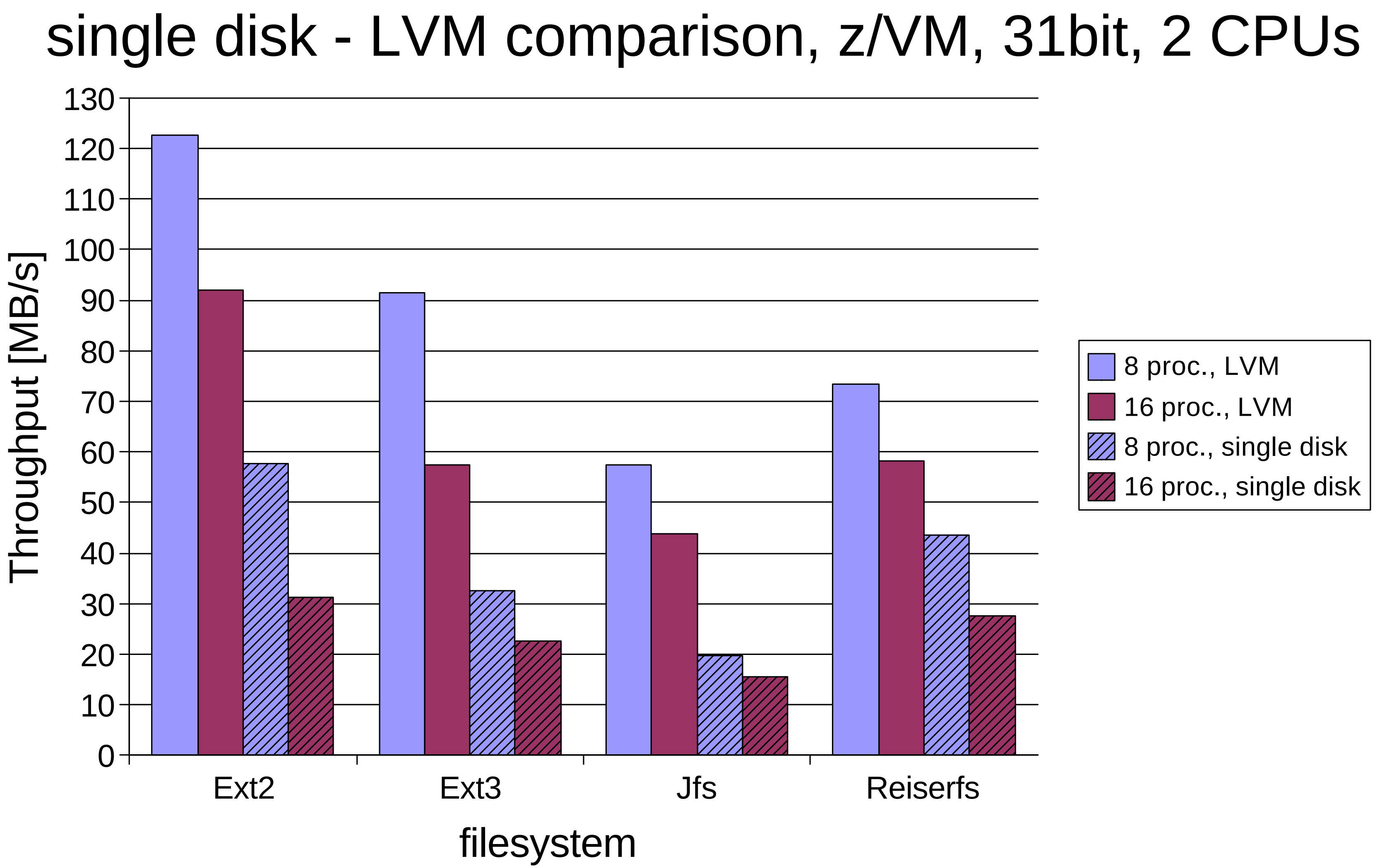


Improving disk performance with LVM



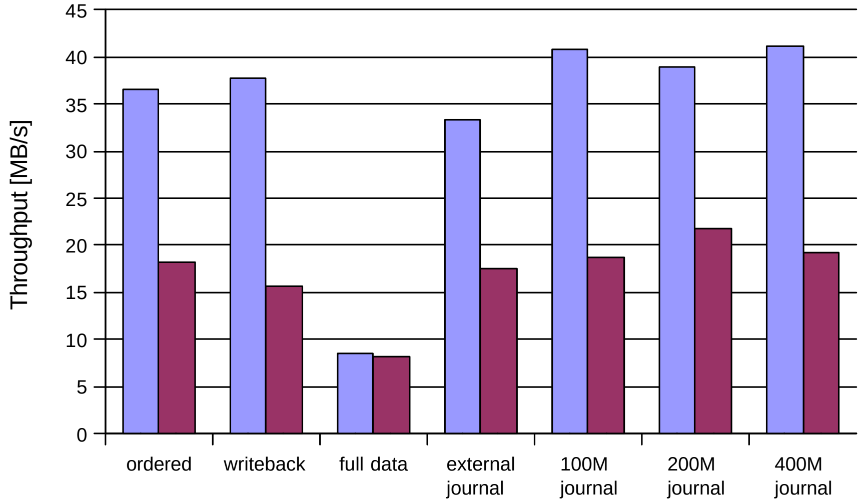
- With LVM and striping parallelism is achieved

LVM results

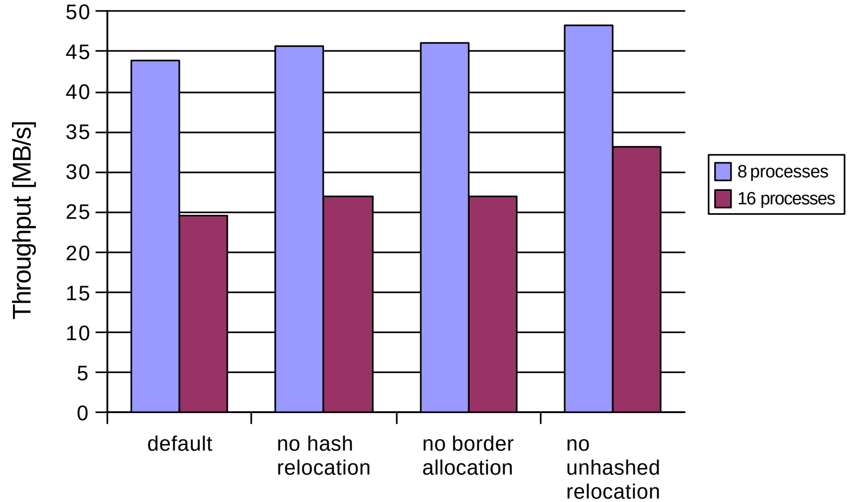

single disk - LVM comparison, z/VM, 31bit, 2 CPUs
Throughput [MB/s]
130
120
110
100
90
80
70
60
50
40
30
20
10
0
Ext2
Ext3
Jfs
Reiserfs
filesystem
8 proc., LVM
16 proc., LVM
8 proc., single disk
16 proc., single disk

filesystem options

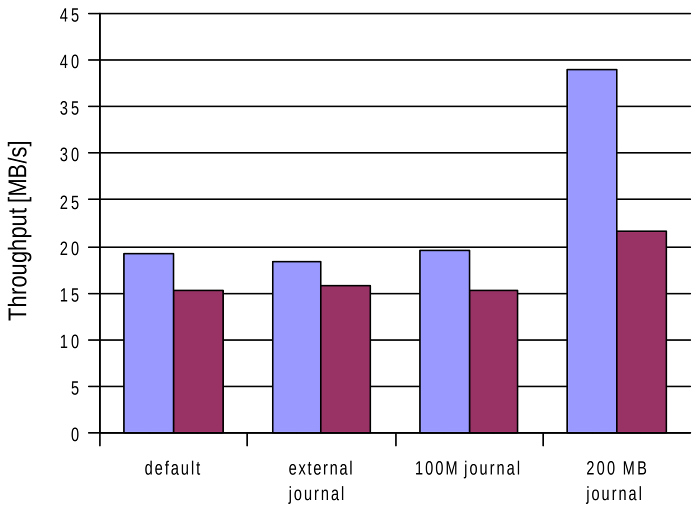
single disk, ext3 optimizations, z/VM, 31bit, 4 CPUs



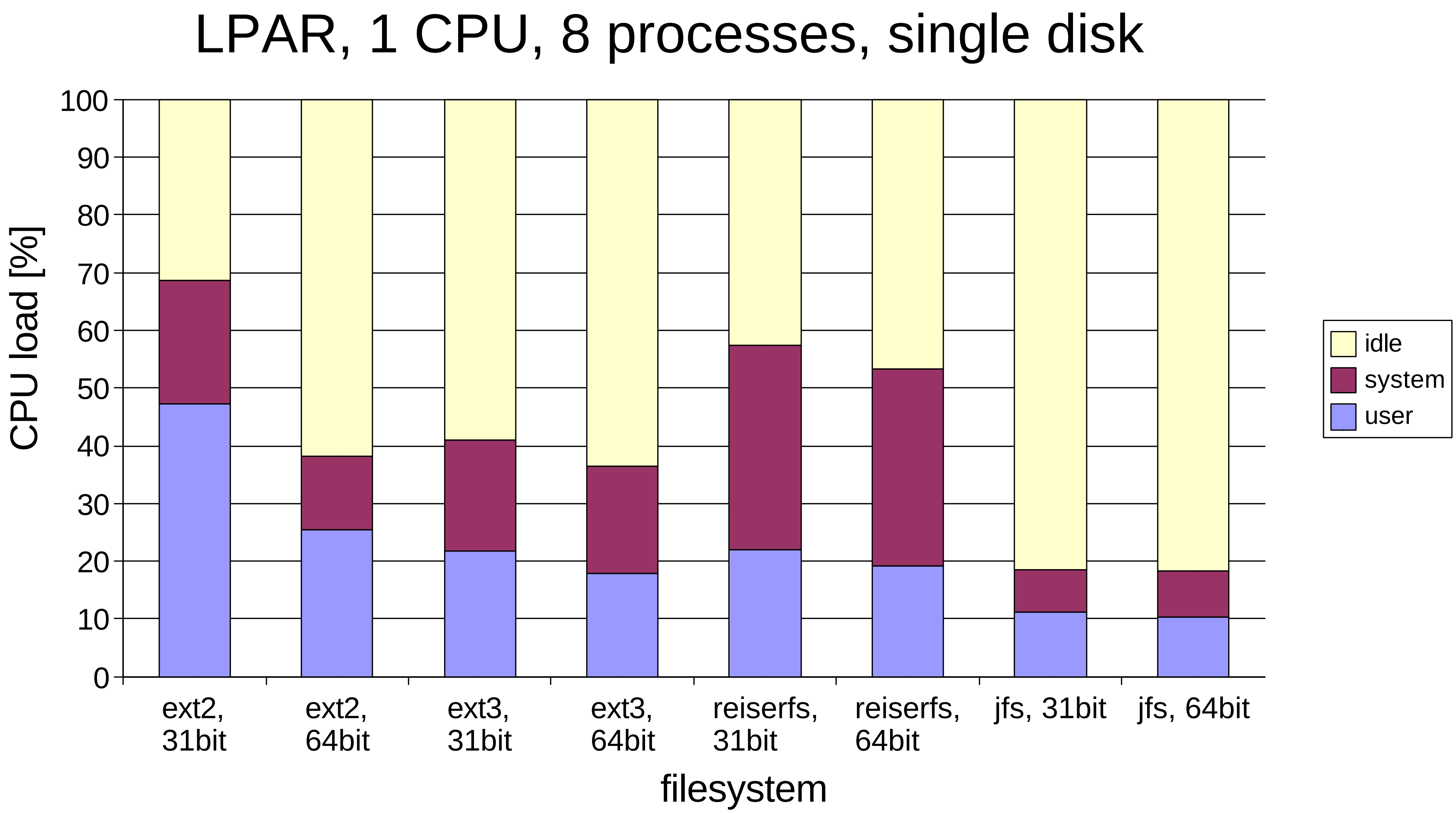
single disk, reiserfs optimizations, z/VM, 31bit, 4 CPUs



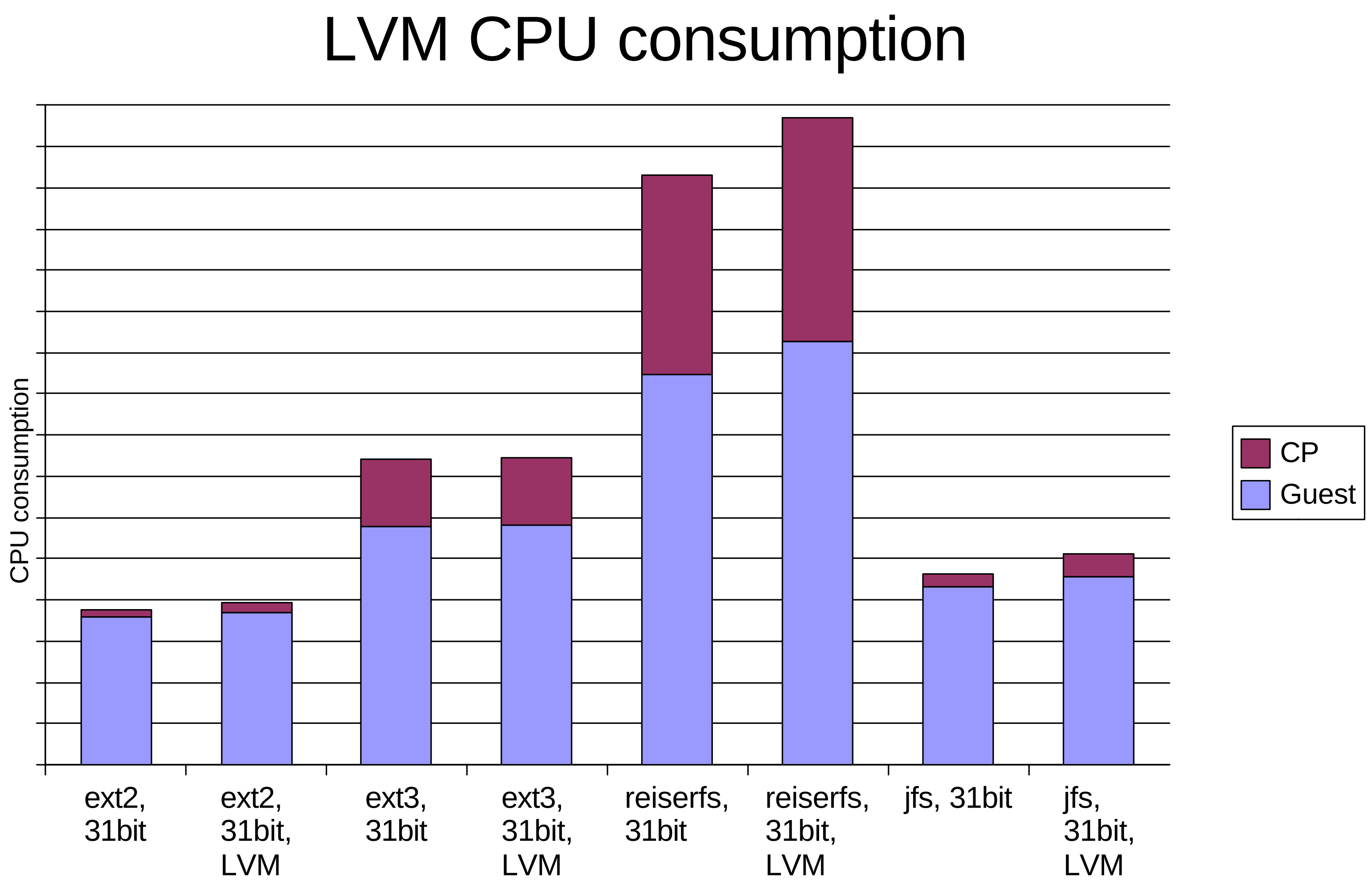
single disk, jfs optimizations, z/VM, 31bit, 4 CPUs



CPU load

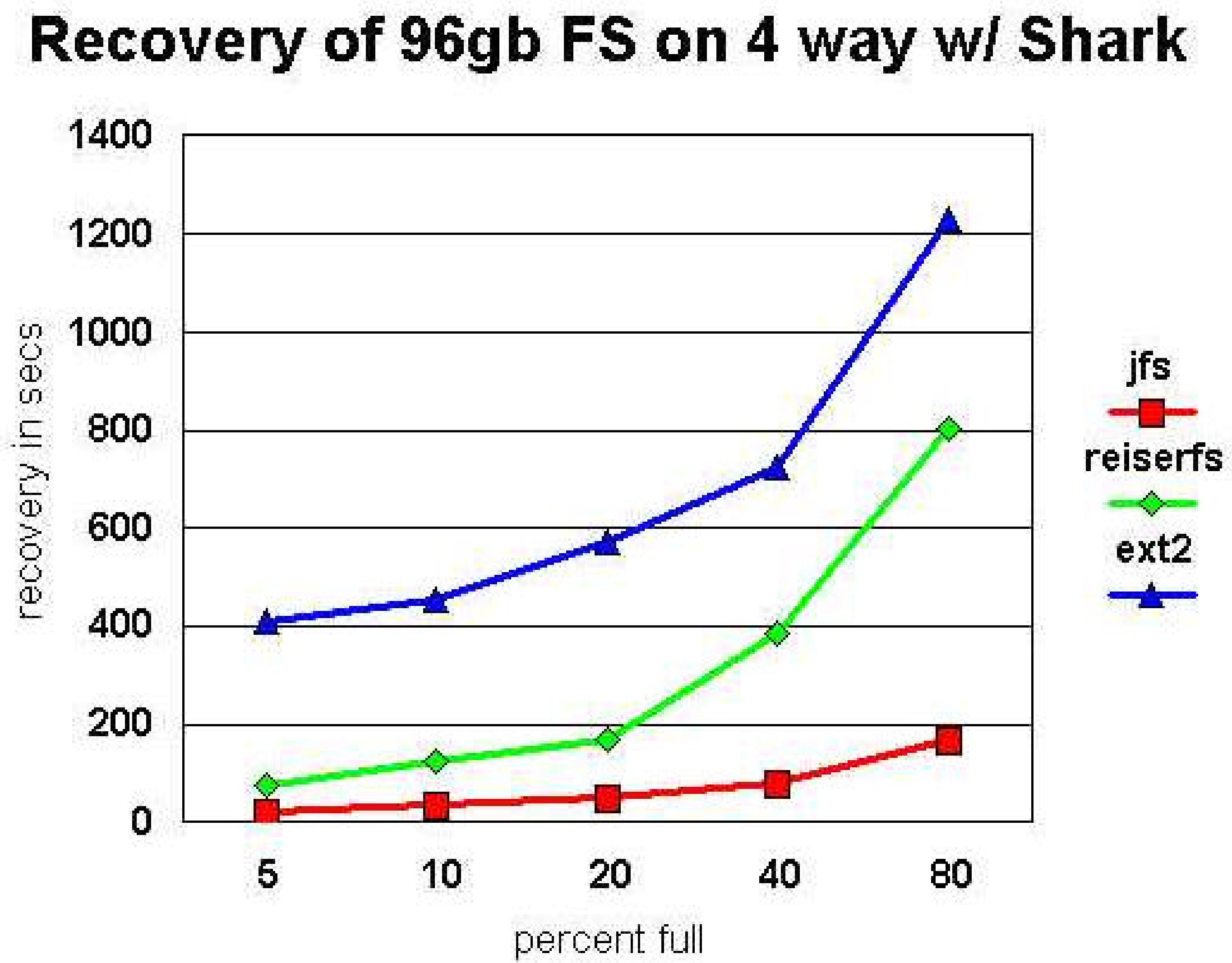

LPAR, 1 CPU, 8 processes, single disk
CPU load [%]
100
90
80
70
60
50
40
30
20
10
0
ext2, 31bit
ext2, 64bit
ext3, 31bit
ext3, 64bit
reiserfs, 31bit
reiserfs, 64bit
jfs, 31bit
jfs, 64bit
filesystem
idle
system
user

VM overhead

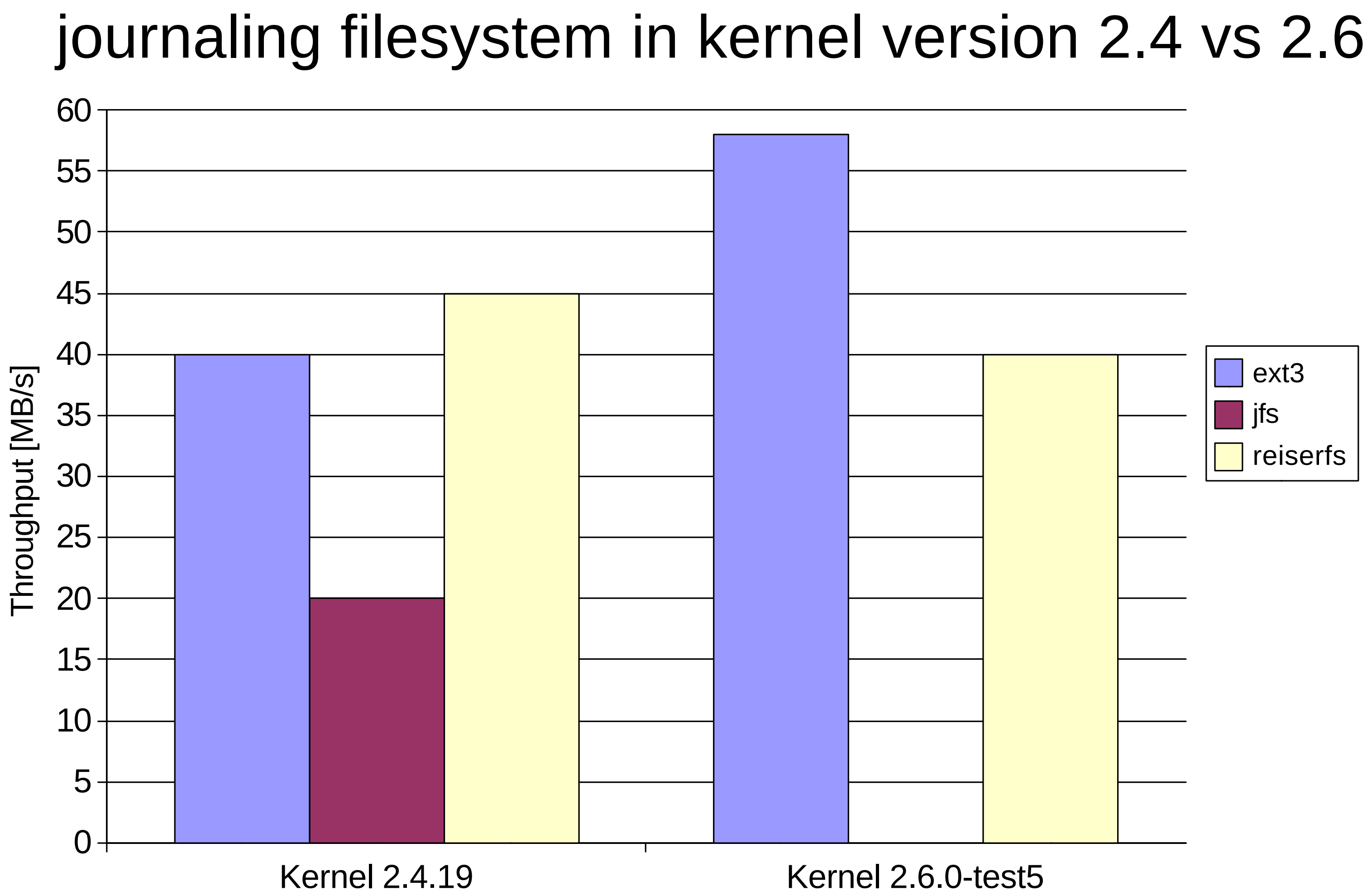


LVM CPU consumption
CPU consumption
CP
Guest
ext2, 31bit
ext2, 31bit, LVM
ext3, 31bit
ext3, 31bit, LVM
reiserfs, 31bit
reiserfs, 31bit, LVM
jfs, 31bit
jfs, 31bit, LVM

recovery times



Outlook on kernel 2.6



- preliminary results
- jfs was not compiled into 2.6 kernel

Summary

- journaling file systems increase data integrity significantly
- journaling file systems dramatically reduce system outage times
- performance cost is at least 30%
- reiserfs is slightly faster than ext3, but needs much more CPU
- journaling file systems profit from LVM
- jfs has fastest recovery times
- 2.6 will bring more improvements (increased throughput, reduced CPU load, iostat for ECKD)

Questions ?