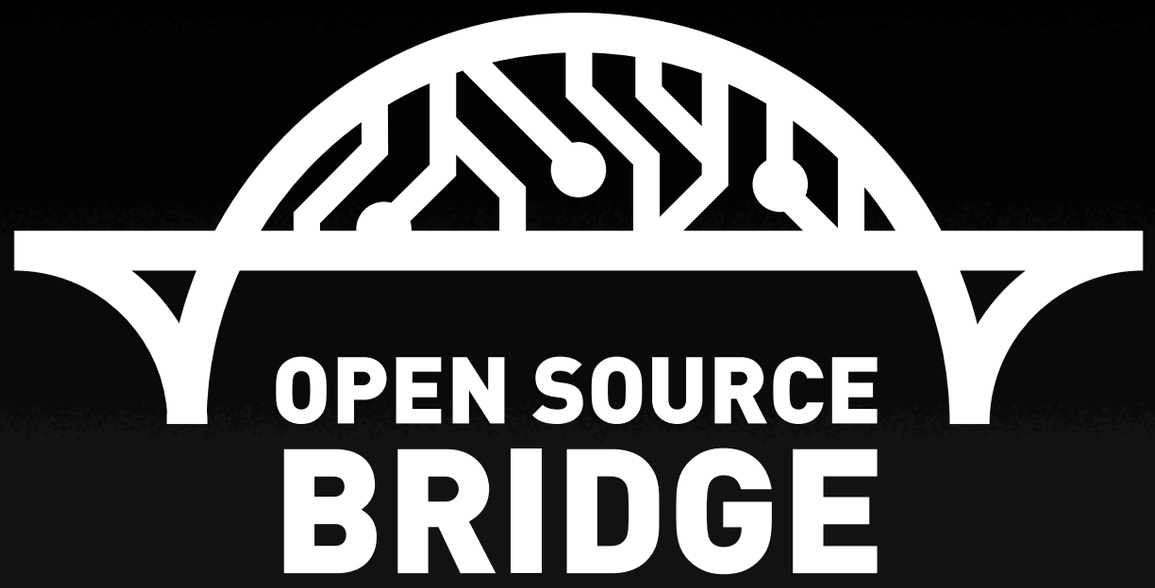
Linux Filesystems Performance for Databases

Portland PostgreSQL Performance Pad

Selena Deckelmann
selena@postgresql.org
PostgreSQL Global Development Group
twitter: @selenamarie


OPEN SOURCE
BRIDGE


PostgreSQL
PDXPUG SINCE 2006
pugs.postgresql.org/pdx

OSCON 2009

Do filesystems do what we expect?

OSCON 2009

We are volunteers.

We think you should run these tests.

We are:
DBAs
Sysadmins
Performance tuners

How will this hardware perform?

How will this filesystem perform?

Why should you care about filesystem-specific performance?

Expectations

Where to start?

The Defaults.

HOW TO START A FIGHT

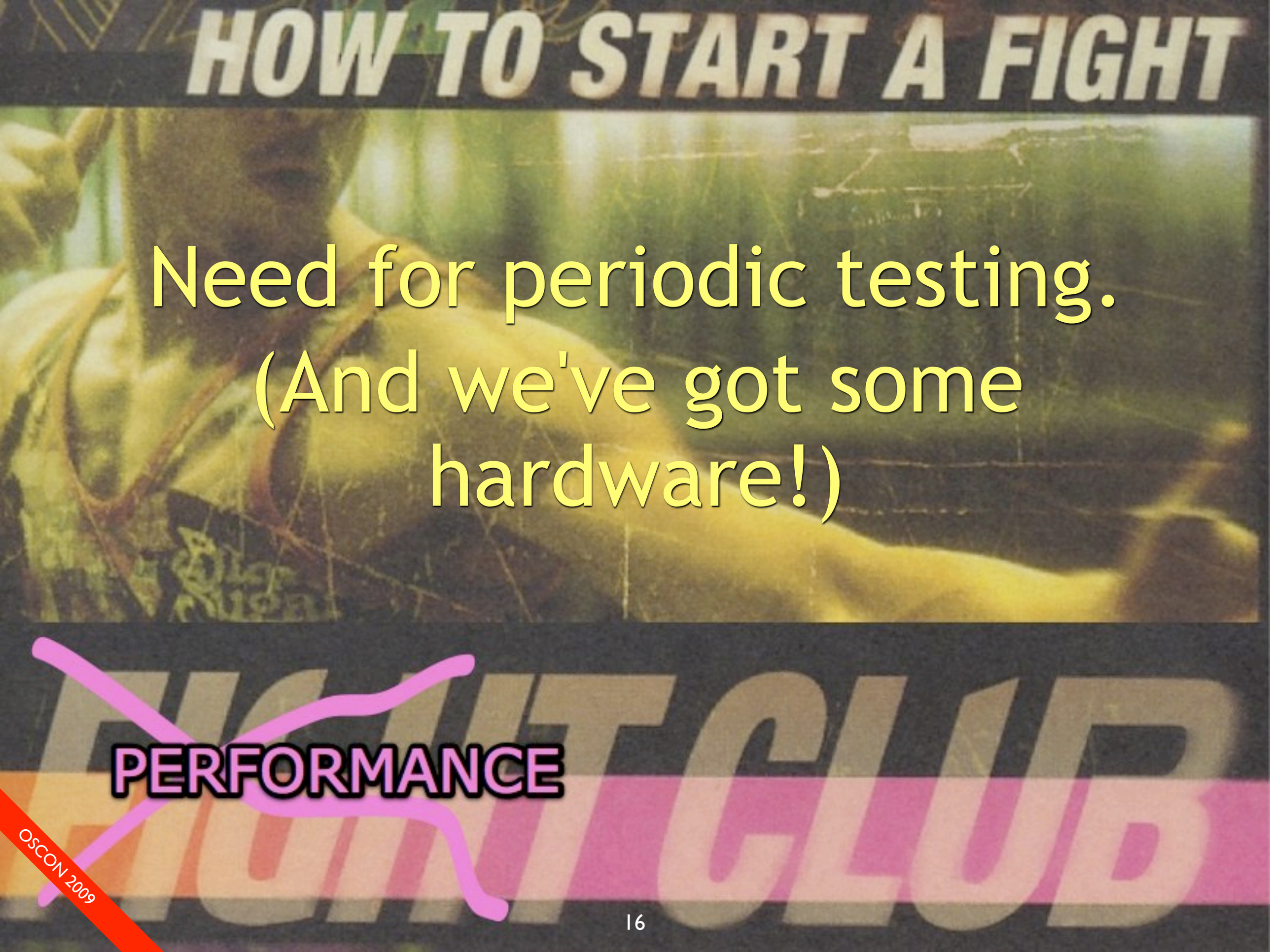
FIGHT CLUB

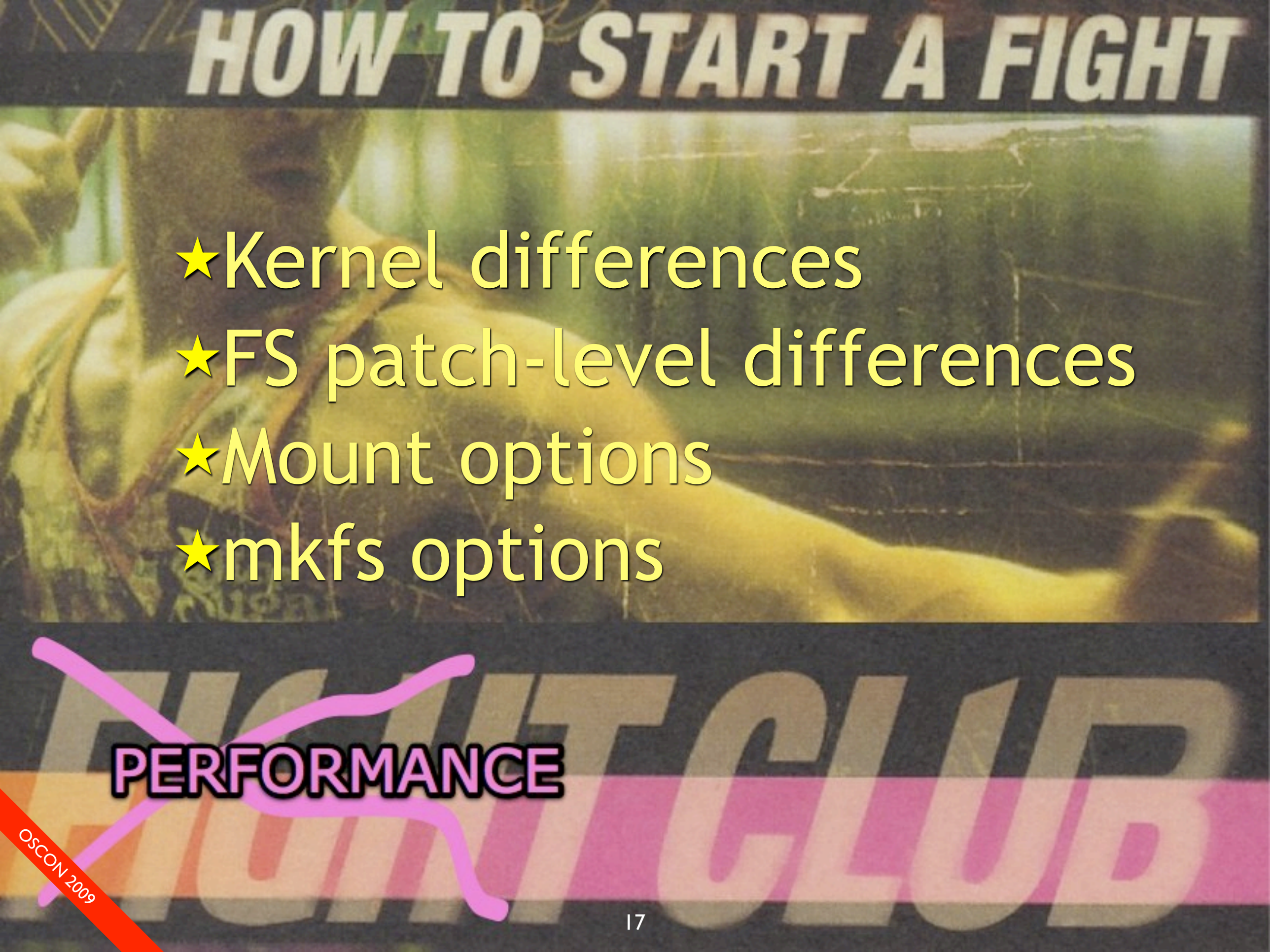
PERFORMANCE

~~PERFORMANCE~~

Very Narrow Use Case: A Relational Database

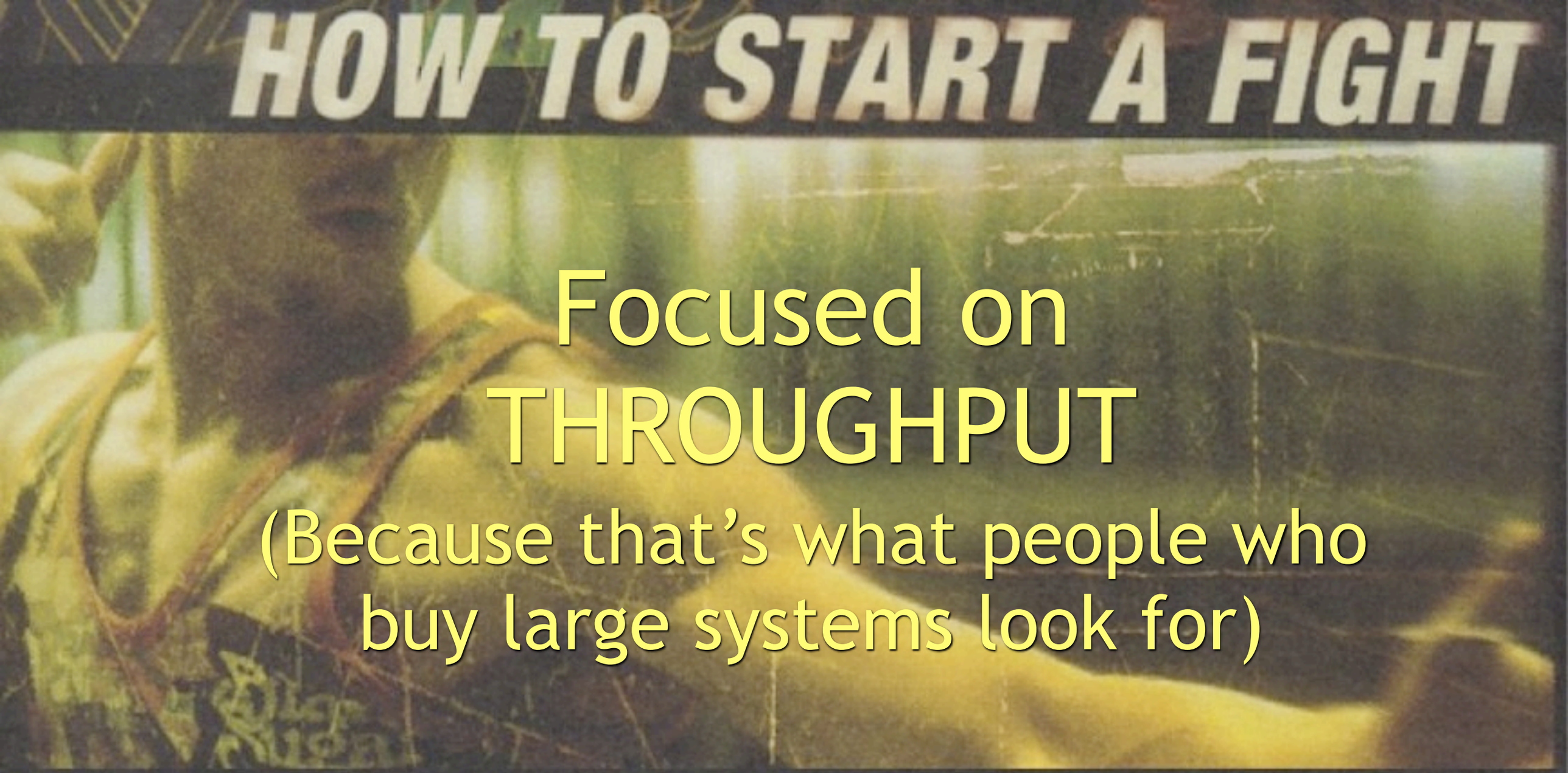
~~PERFORMANCE~~


HOW TO START A FIGHT
Need for periodic testing.
(And we've got some
hardware!)
FIGHT CLUB
PERFORMANCE
OSCON 2009



- Kernel differences
- FS patch-level differences
- Mount options
- mkfs options

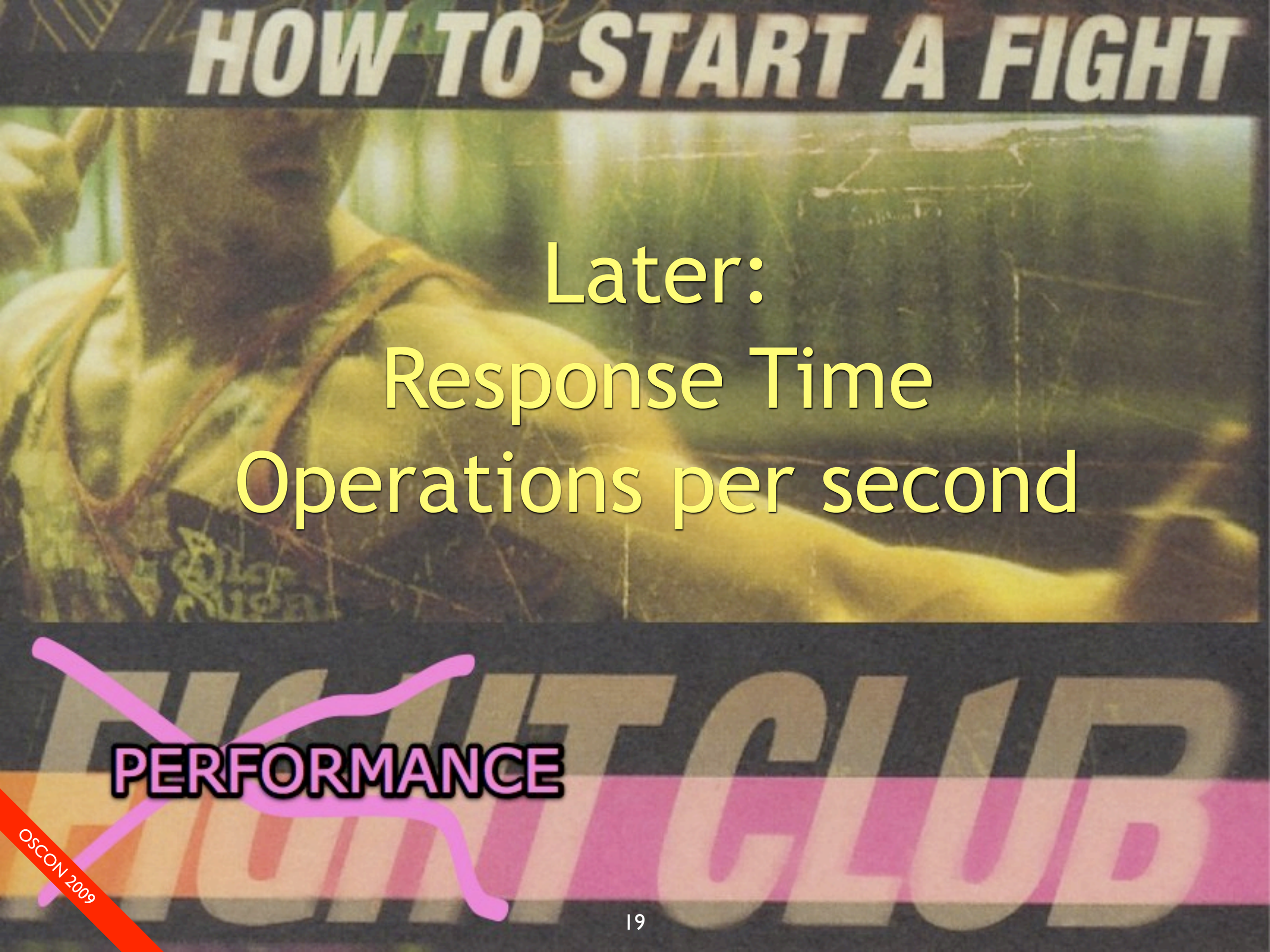
~~PERFORMANCE~~



Focused on THROUGHPUT

(Because that's what people who buy large systems look for)

~~PERFORMANCE~~


HOW TO START A FIGHT
Later:
Response Time
Operations per second
PERFORMANCE
FIGHT CLUB

No, we will not be testing ZFS.

BtrFS (nope, not yet)

What do we expect?

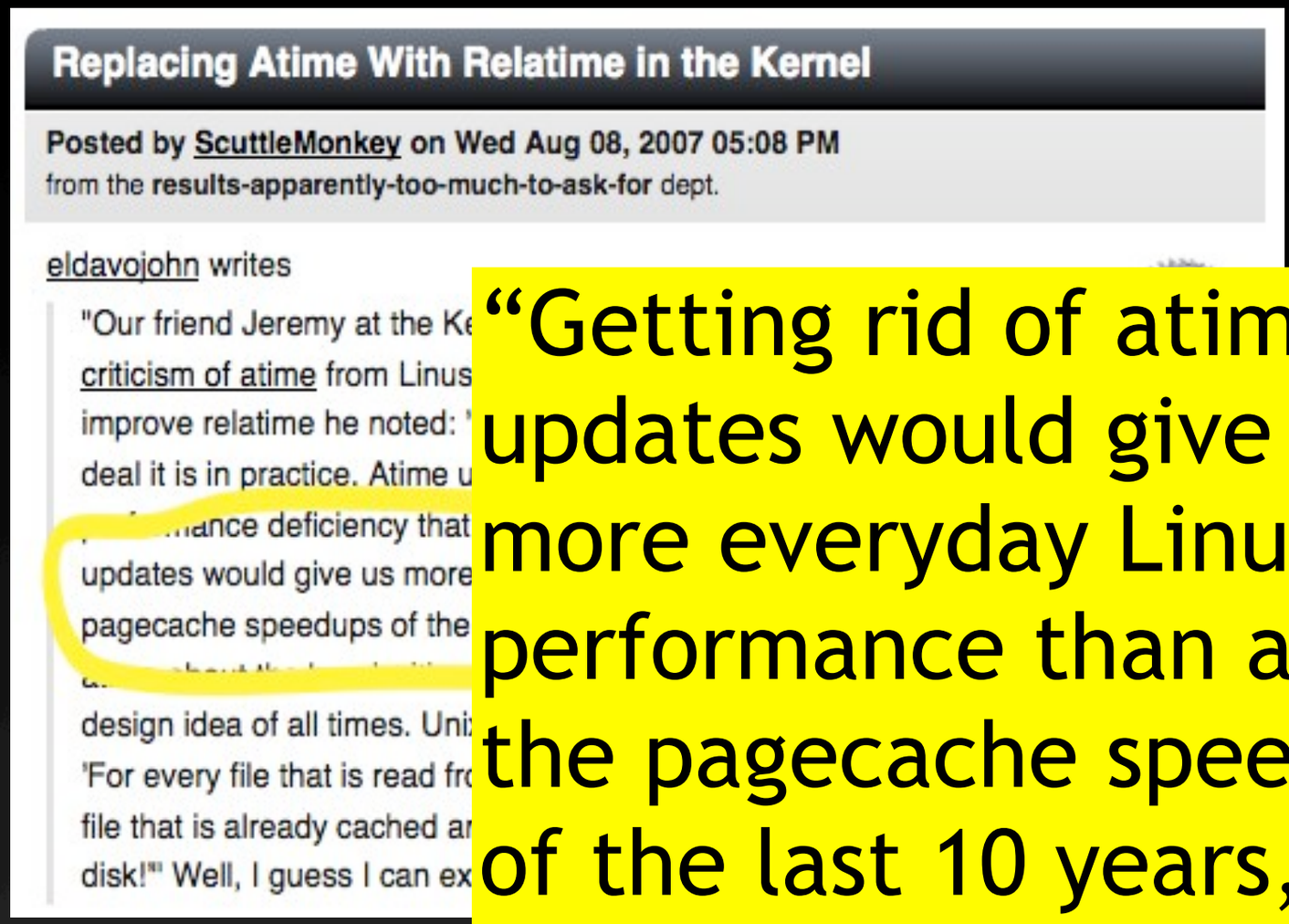
Some conventional wisdom:

“RAID5 is the worst choice for a database.”

“LVM incurs
too much overhead
to use.”

“Striping doubles performance.”

“Turning off 'atime'
is a big
performance gain.”



Replacing Atime With Relatime in the Kernel

Posted by ScuttleMonkey on Wed Aug 08, 2007 05:08 PM
from the **results-apparently-too-much-to-ask-for** dept.

eldavojohn writes

> "Our friend Jeremy at the Ke
> criticism of atime from Linus
> improve relatime he noted: '
> deal it is in practice. Atime u
> ...nance deficiency that
> updates would give us more
> pagecache speedups of the
> ...
> design idea of all times. Uni
> 'For every file that is read fr
> file that is already cached ar
> disk!'" Well, I guess I can ex

"Getting rid of atime updates would give us more everyday Linux performance than all the pagecache speedups of the last 10 years, _combined_."

“Journaling filesystems (ext3) will have worse performance than non-journaling filesystems (ext2).”

“Your read-ahead buffer is big enough.”

Now... on to the good stuff.


INTERNET.
SERIOUS BUSINESS.

PostgreSQL's Portland Performance Pad

Hosted by CommandPrompt, Inc.

Our machine:

HP ProLiant DL380G5
Smart Array p800
72GB 15,000 RPM SAS (up to 25 disks)
32GB RAM

Linux:
2.6.25-gentoo-r6
*New tests being run with 2.6.28

Our machine:
Chosen because
of it's low, low price.

Thank you, HP.

Our tests:

fio
64 GB working set
8 threads
no fadvise
no direct i/o
8KB blocksize
I/O elevator: deadline

Our tests:

fio

- read (sequential, random)
- write (sequential, random)
- read-write (50/50 mix)

Our stats:

sar
mpstat
iostat
vmstat
readprofile

Our tests:
Chosen because of their relevance to PostgreSQL

Filesystems Tested:

ext2
ext3
jfs
xfs
reiserfs
ext4 (but had trouble)

Disk configs tested:

Single disk
RAID-0
RAID-1
RAID-5
RAID-10
RAID-6

The Data:

http://moourl.com/fsperf

OSCON 2009

Confessions:

- May be high standard deviation with results (don't know yet!)

- No filesystem tuning, all default create and mount options

- No software raid comparison or lvm (volume management test) for 2.6.28 tests

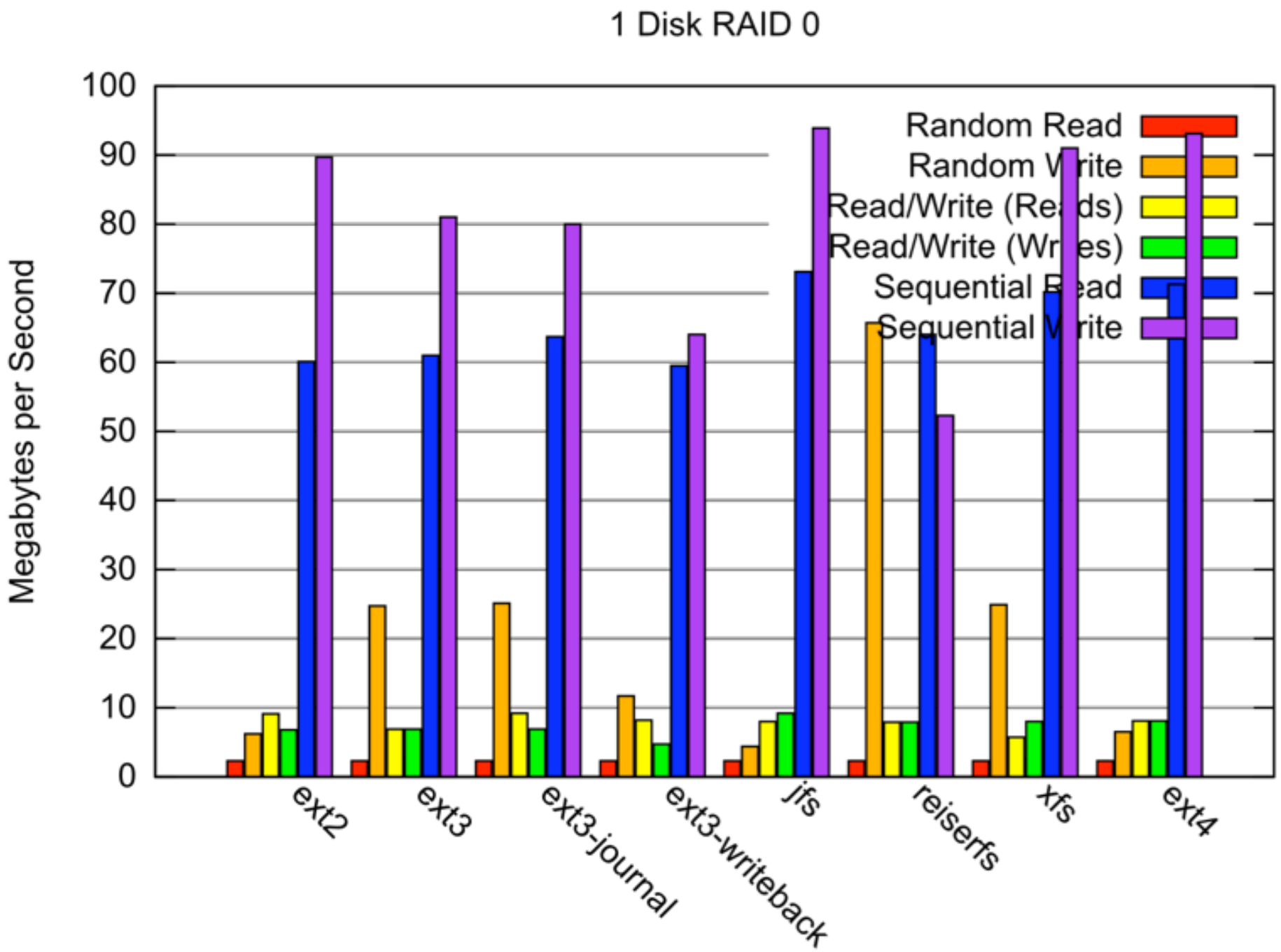
Confessions:

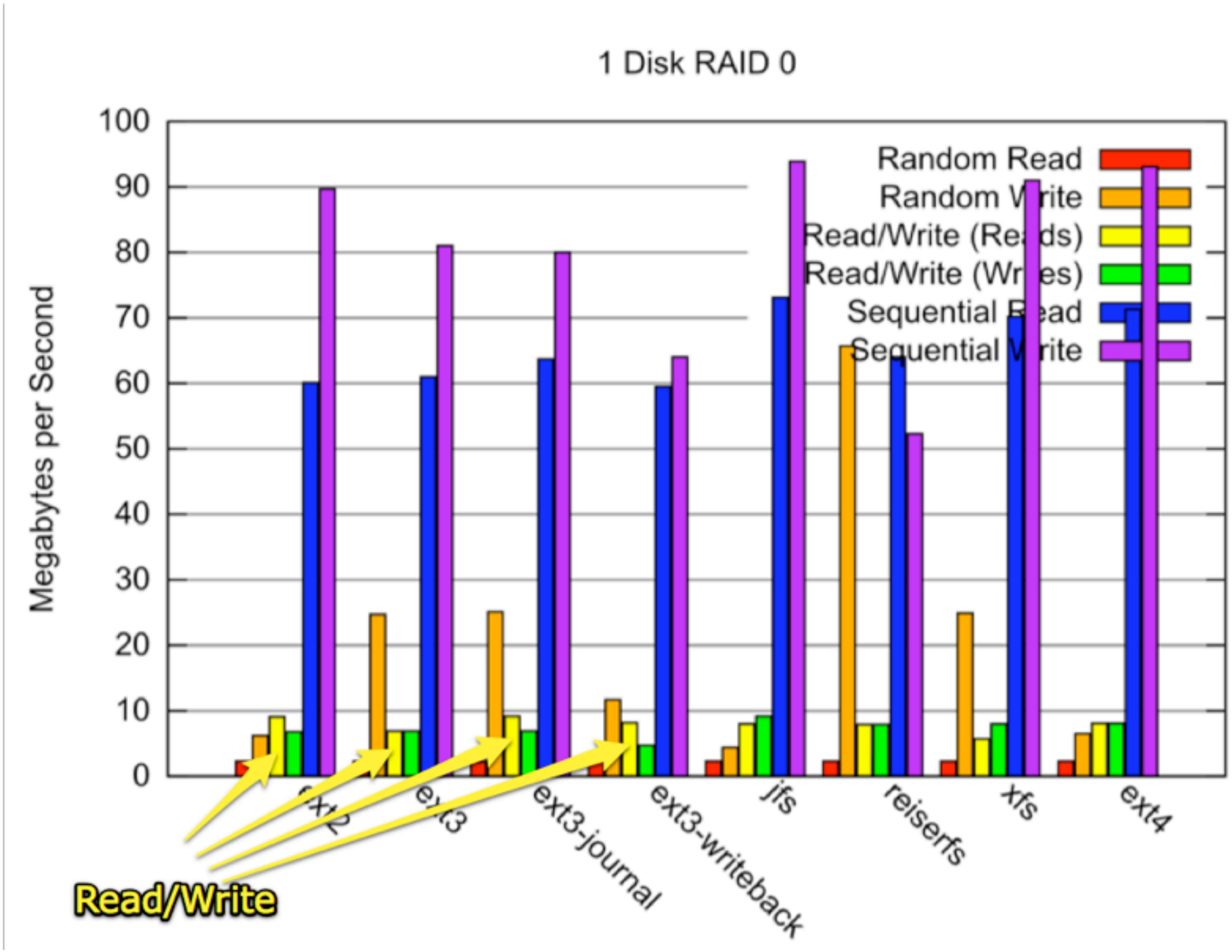
- Some xfs runs had to be repeated and some ext4 runs did not complete successfully
- Only presenting throughput
- Interested in system performance for a specific application, not code performance

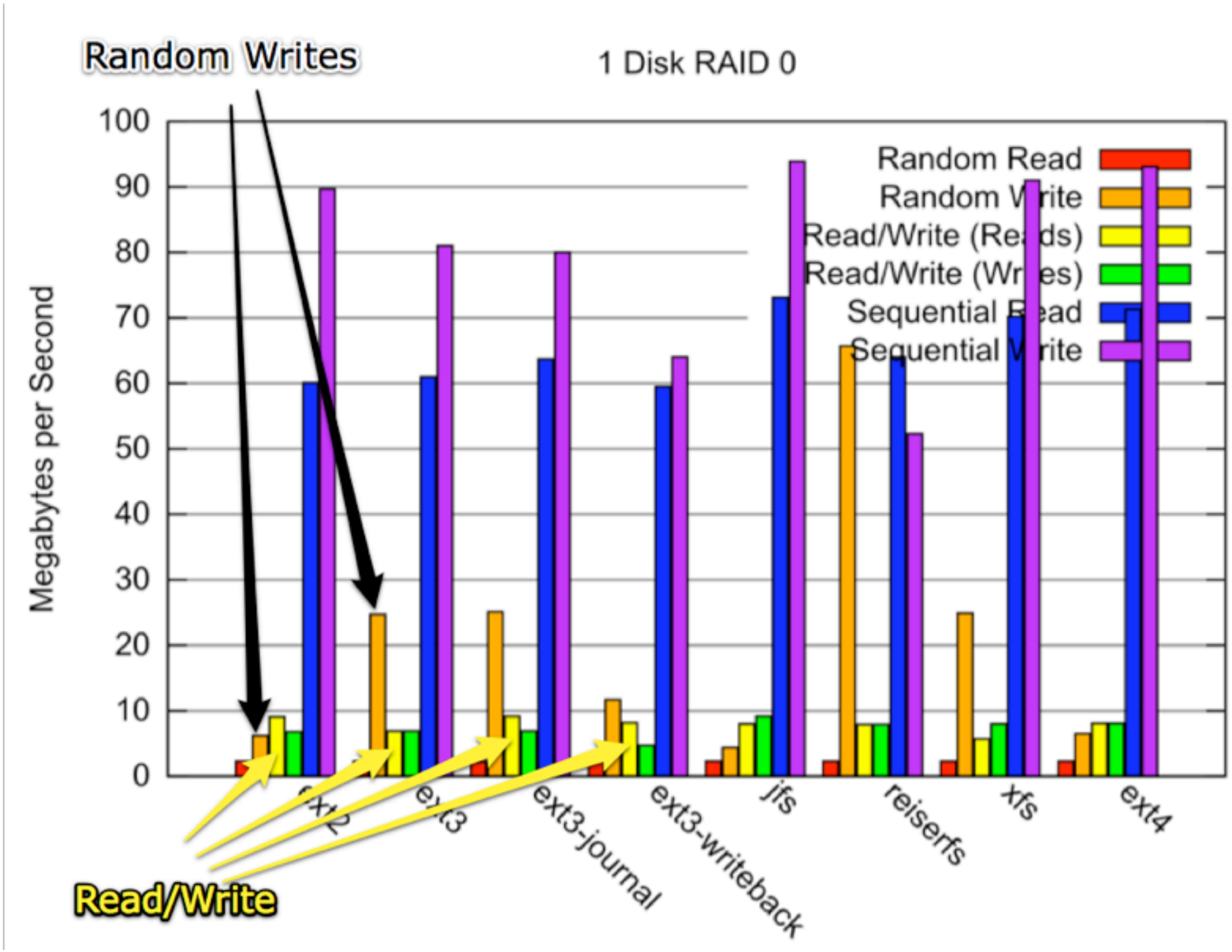
Confessions:

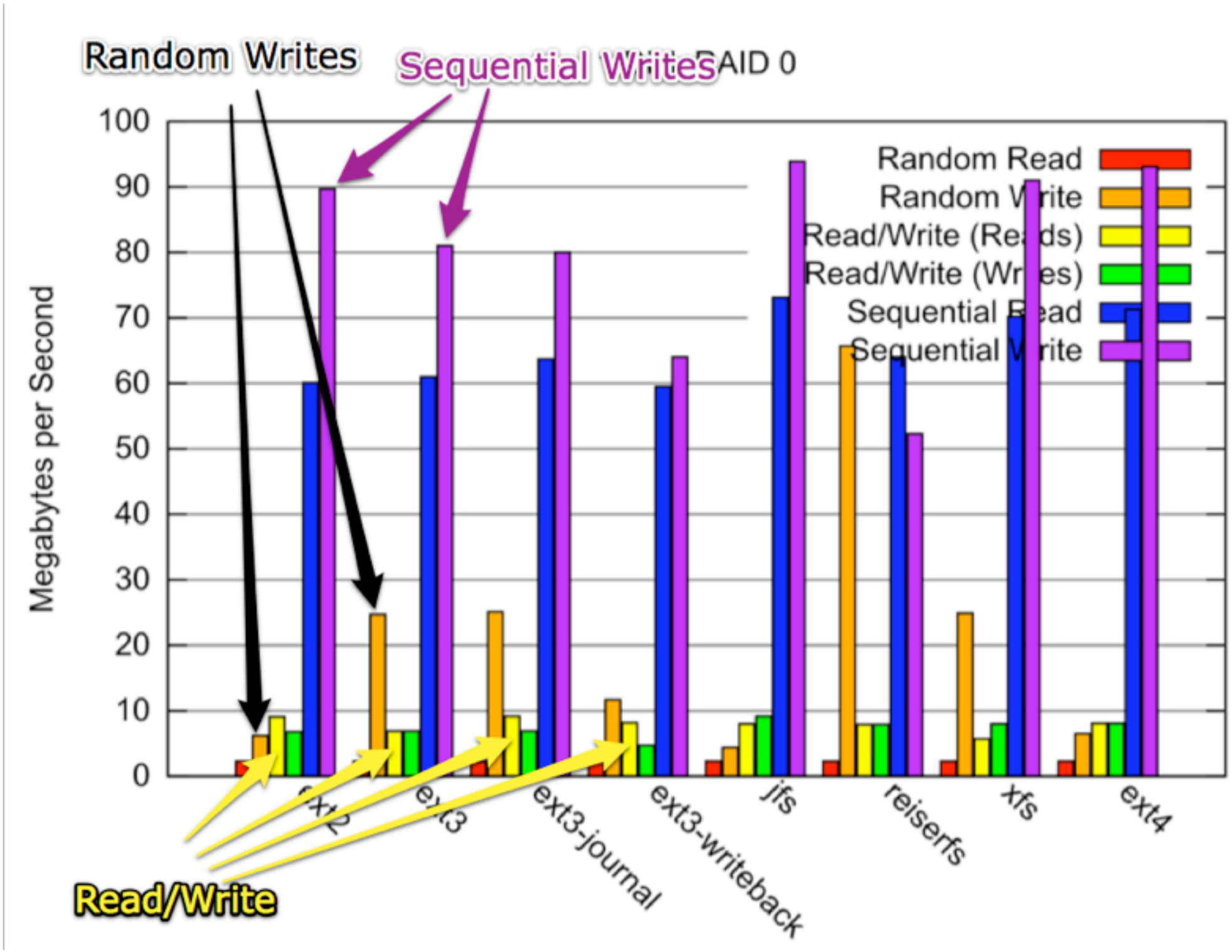
- I/O profiles don’t exhibit atime or partition alignment issues
- Disk controller firmware not at the latest version in 2.6.25 tests
- Software RAID is on top of 1 disk RAID 0 devices (HP SmartArray doesn’t have JBOD option)

HOW TO START A FIGHT
AUDIENCE PARTICIPATION
Higher throughput:
ext2 or ext3?
PERFORMANCE
FIGHT CLUB


1 Disk RAID 0
Megabytes per Second
100
90
80
70
60
50
40
30
20
10
0
Random Read
Random Write
Read/Write (Reads)
Read/Write (Writes)
Sequential Read
Sequential Write
ext2
ext3
ext3-journal
ext3-writeback
jfs
reiserfs
xfs
ext4
(Default Mount Options)

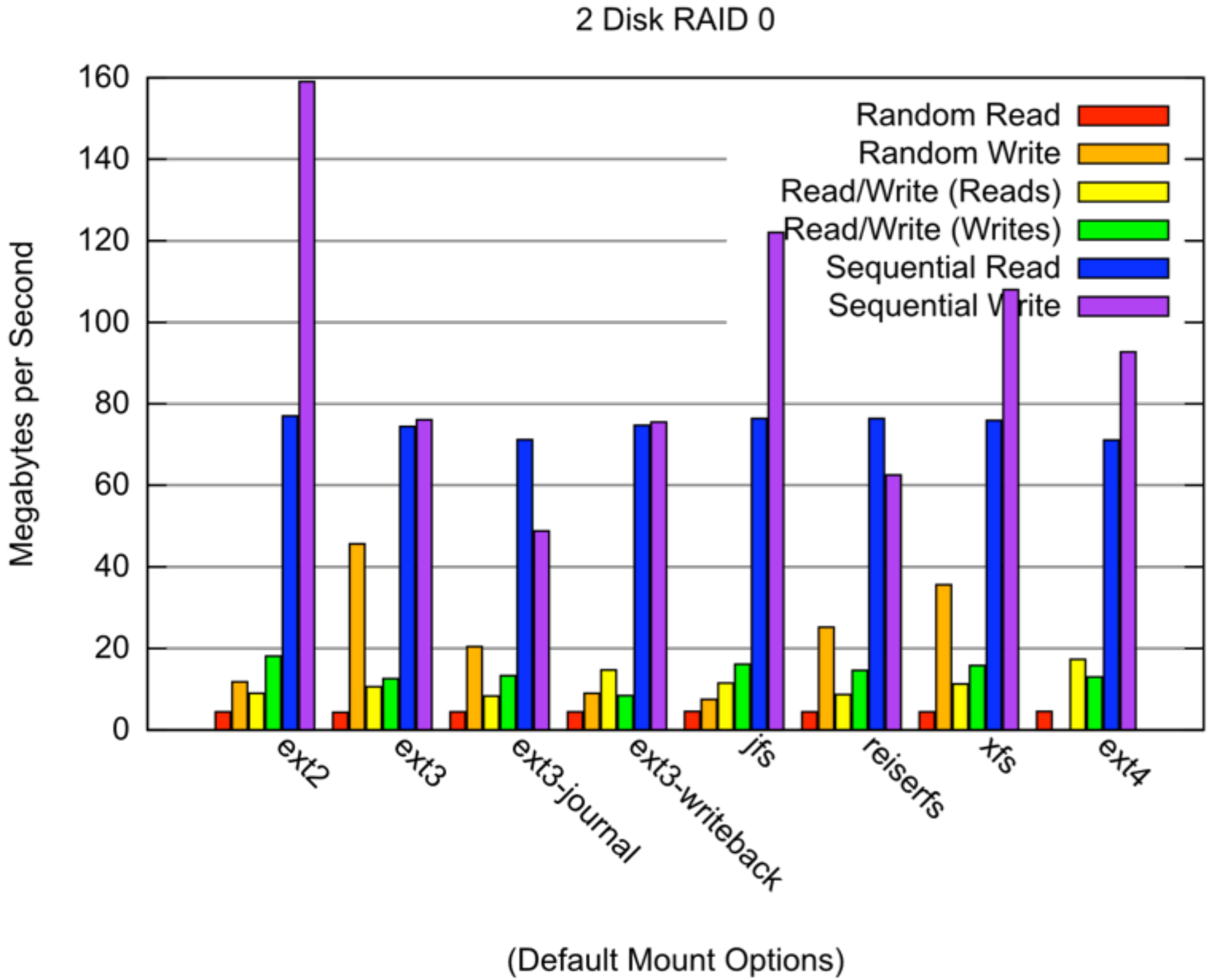

1 Disk RAID 0
Megabytes per Second
100
90
80
70
60
50
40
30
20
10
0
Random Read
Random Write
Read/Write (Reads)
Read/Write (Writes)
Sequential Read
Sequential Write
ext2
ext3
ext3-journal
ext3-writeback
jfs
reiserfs
xfs
ext4
Read/Write

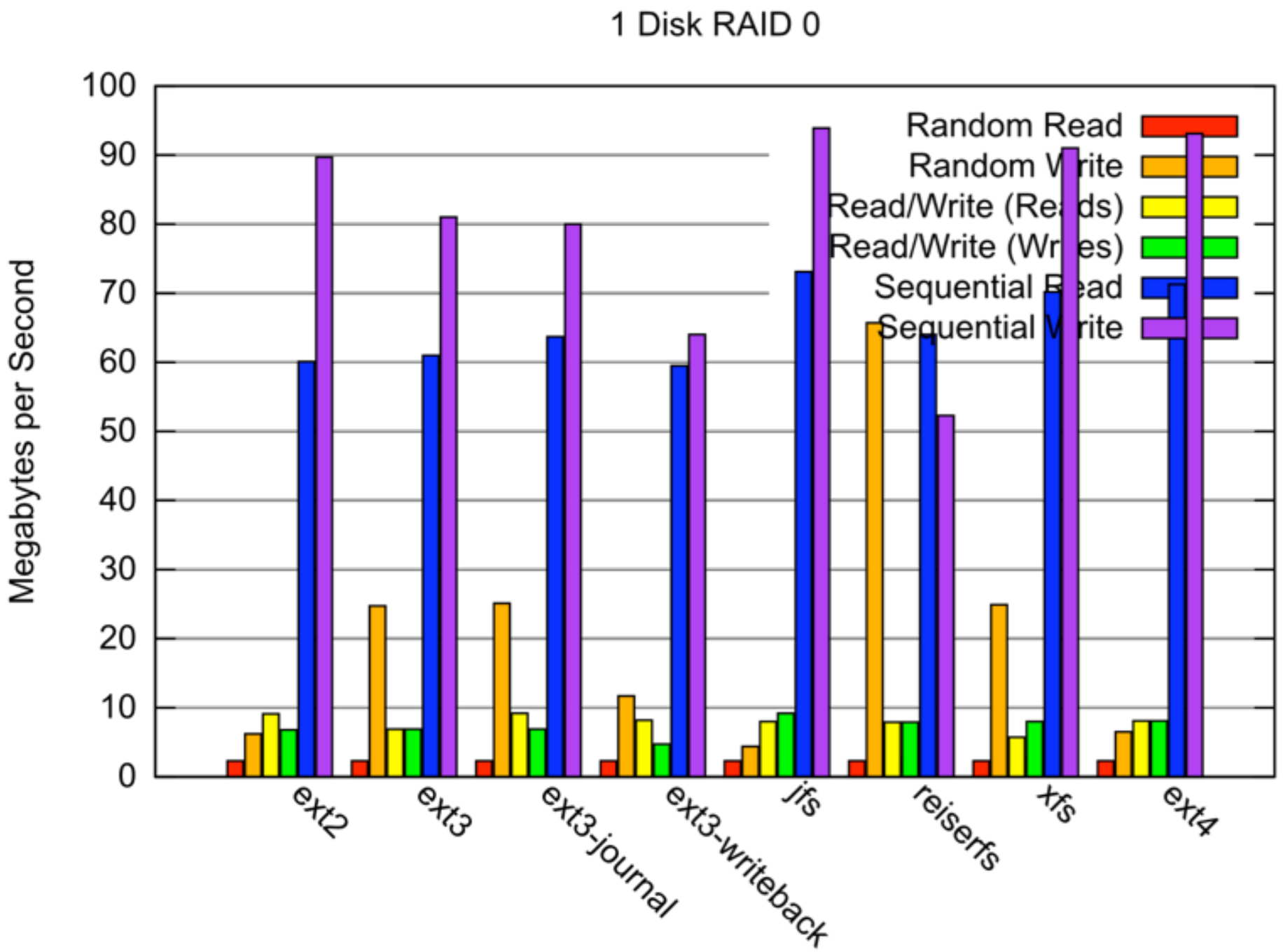

Random Writes
1 Disk RAID 0
Megabytes per Second
100
90
80
70
60
50
40
30
20
10
0
Random Read
Random Write
Read/Write (Reads)
Read/Write (Writes)
Sequential Read
Sequential Write
ext2
ext3
ext3-journal
ext3-writeback
jfs
reiserfs
xfs
ext4
Read/Write

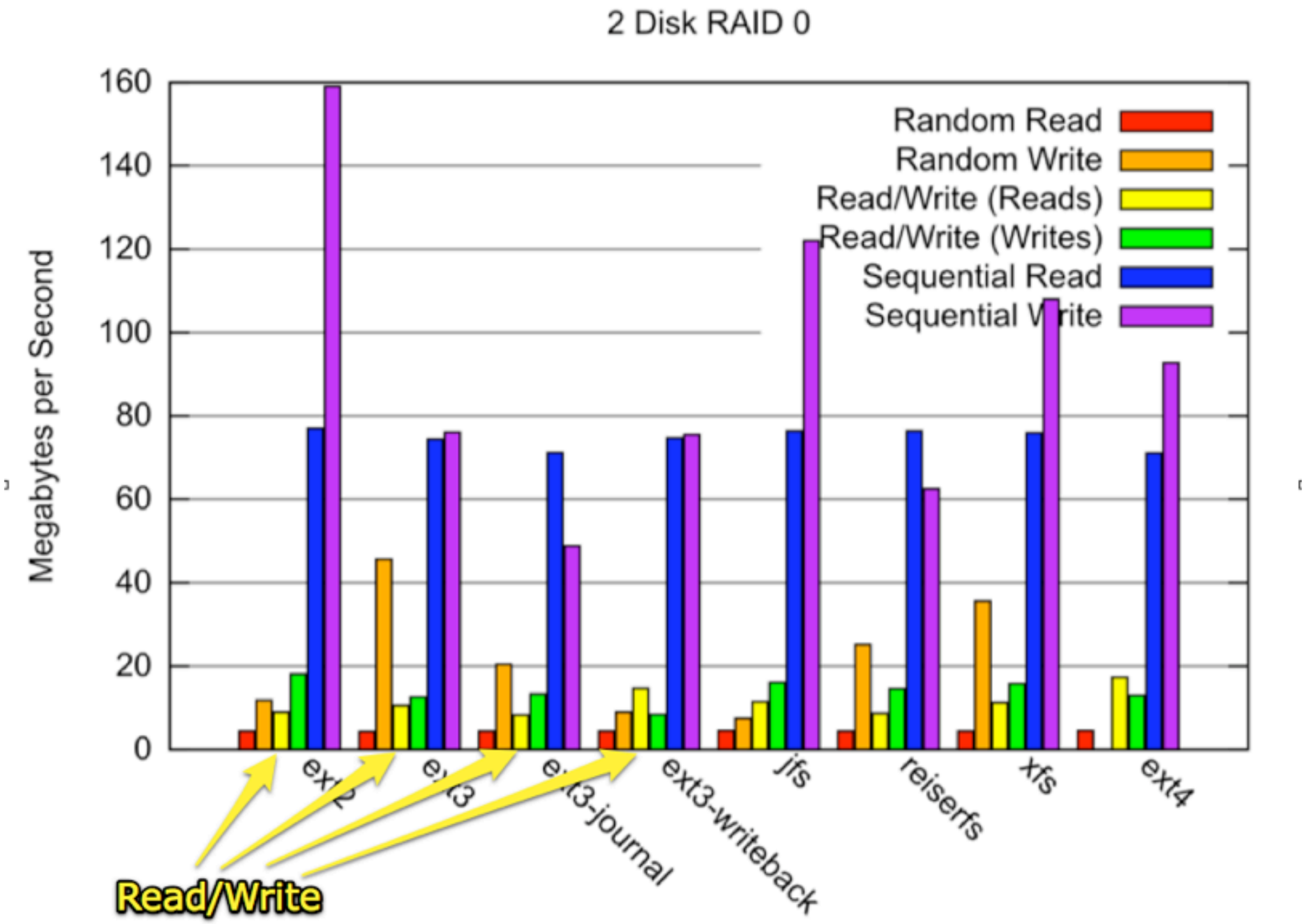

Random Writes
Sequential Writes
RAID 0
Megabytes per Second
100
90
80
70
60
50
40
30
20
10
0
Random Read
Random Write
Read/Write (Reads)
Read/Write (Writes)
Sequential Read
Sequential Write
ext2
ext3
ext3-journal
ext3-writeback
jfs
reiserfs
xfs
ext4
Read/Write

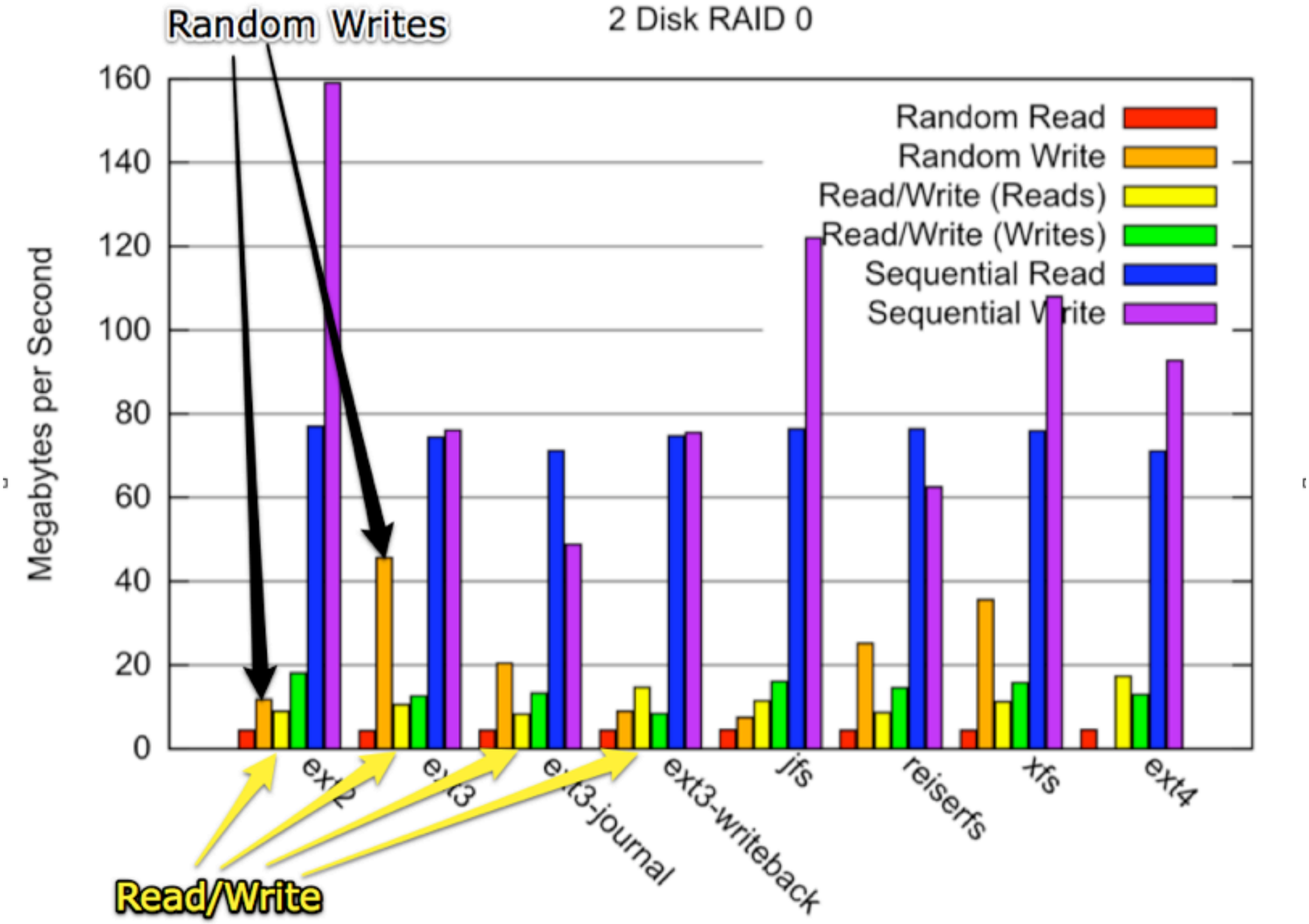
Seek bundling/batching in ext3 is better?

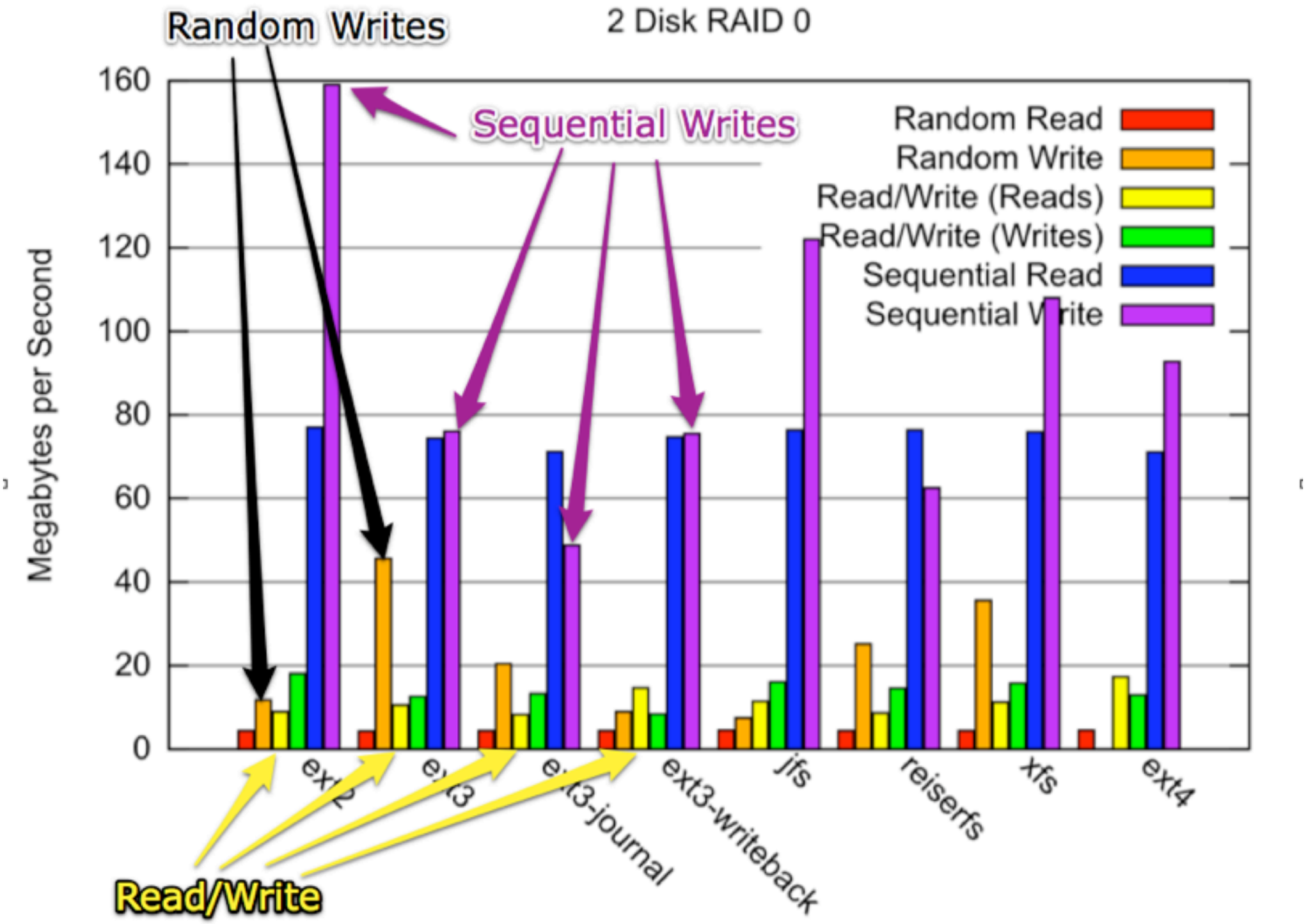
What if we add a disk?

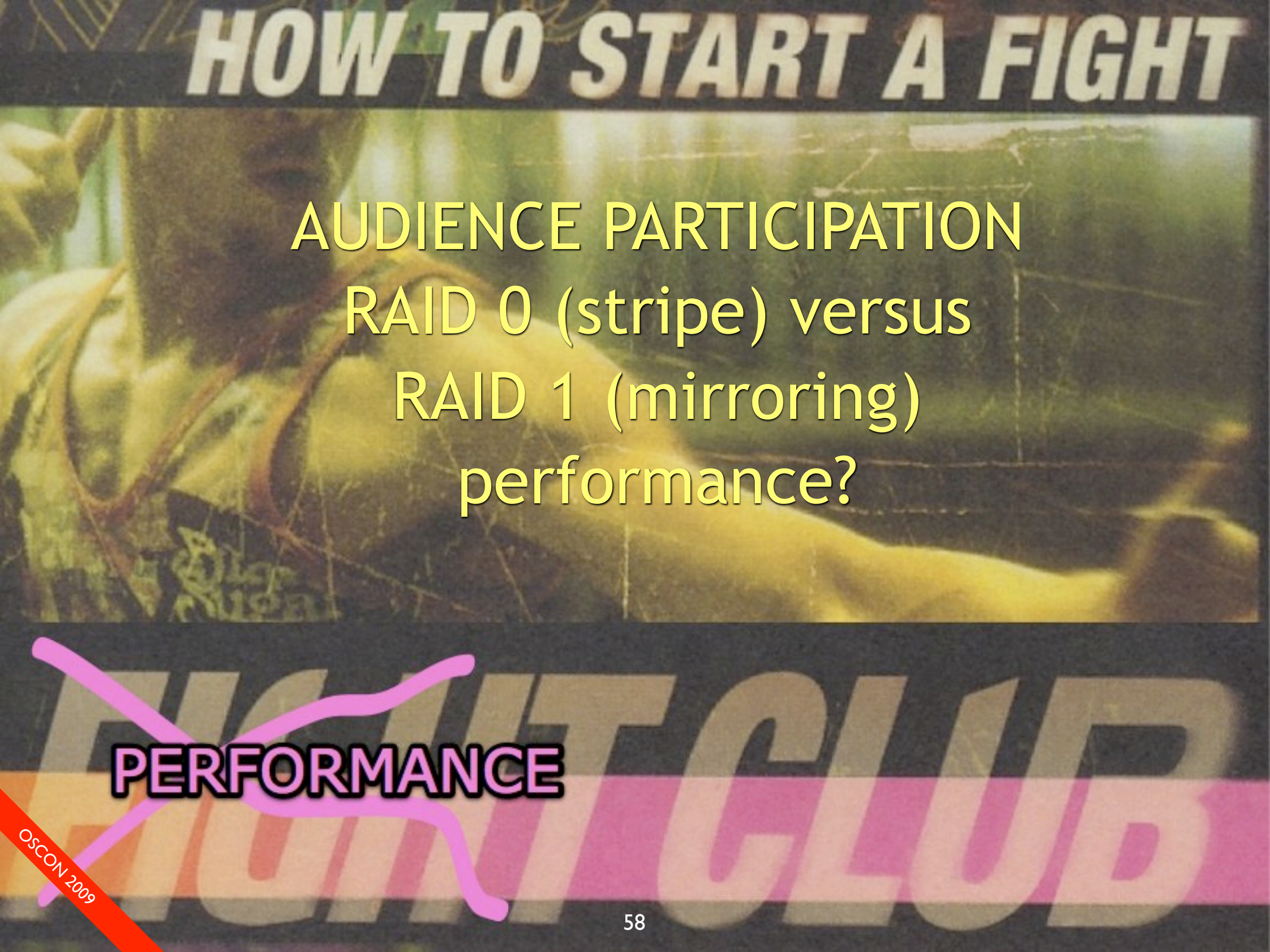

2 Disk RAID 0
Megabytes per Second
160
140
120
100
80
60
40
20
0
Random Read
Random Write
Read/Write (Reads)
Read/Write (Writes)
Sequential Read
Sequential Write
ext2
ext3
ext3-journal
ext3-writeback
jfs
reiserfs
xfs
ext4
(Default Mount Options)

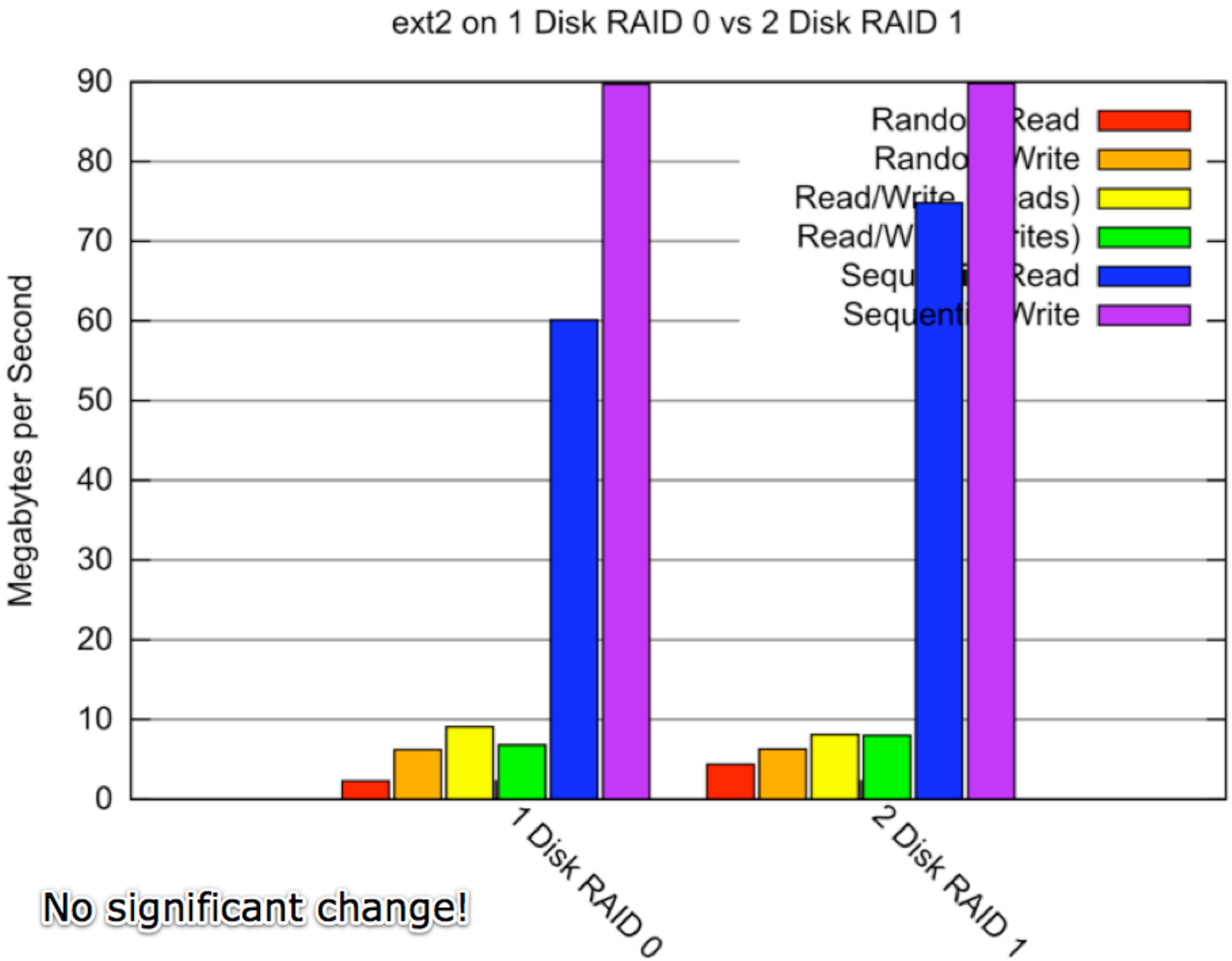

1 Disk RAID 0
Megabytes per Second
100
90
80
70
60
50
40
30
20
10
0
Random Read
Random Write
Read/Write (Reads)
Read/Write (Writes)
Sequential Read
Sequential Write
ext2
ext3
ext3-journal
ext3-writeback
jfs
reiserfs
xfs
ext4
(Default Mount Options)

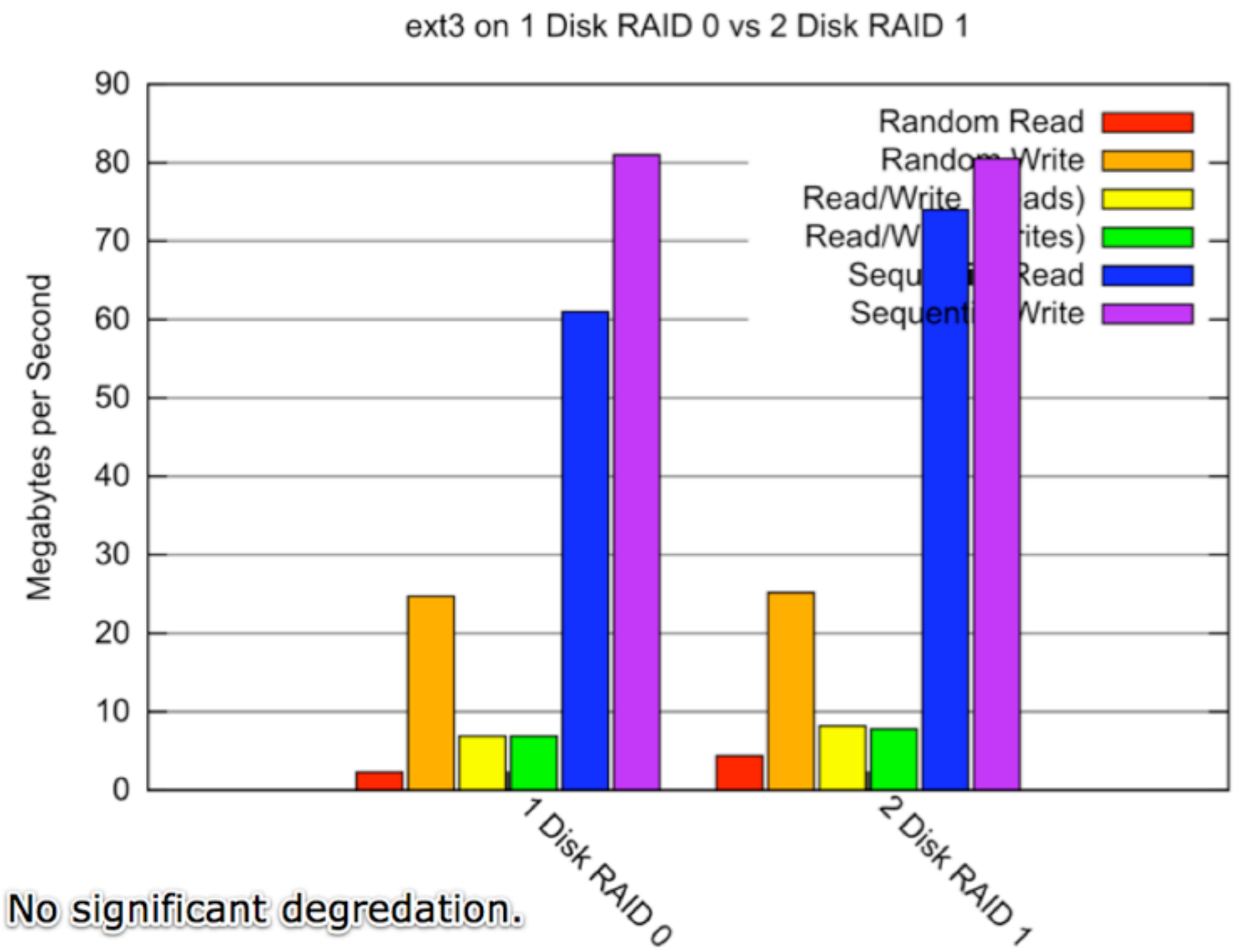

2 Disk RAID 0
Megabytes per Second
160
140
120
100
80
60
40
20
0
Random Read
Random Write
Read/Write (Reads)
Read/Write (Writes)
Sequential Read
Sequential Write
ext2
ext3
ext3-journal
ext3-writeback
jfs
reiserfs
xfs
ext4
Read/Write


Random Writes
2 Disk RAID 0
Megabytes per Second
160
140
120
100
80
60
40
20
0
Random Read
Random Write
Read/Write (Reads)
Read/Write (Writes)
Sequential Read
Sequential Write
ext2
ext3
ext3-journal
ext3-writeback
jfs
reiserfs
xfs
ext4
Read/Write

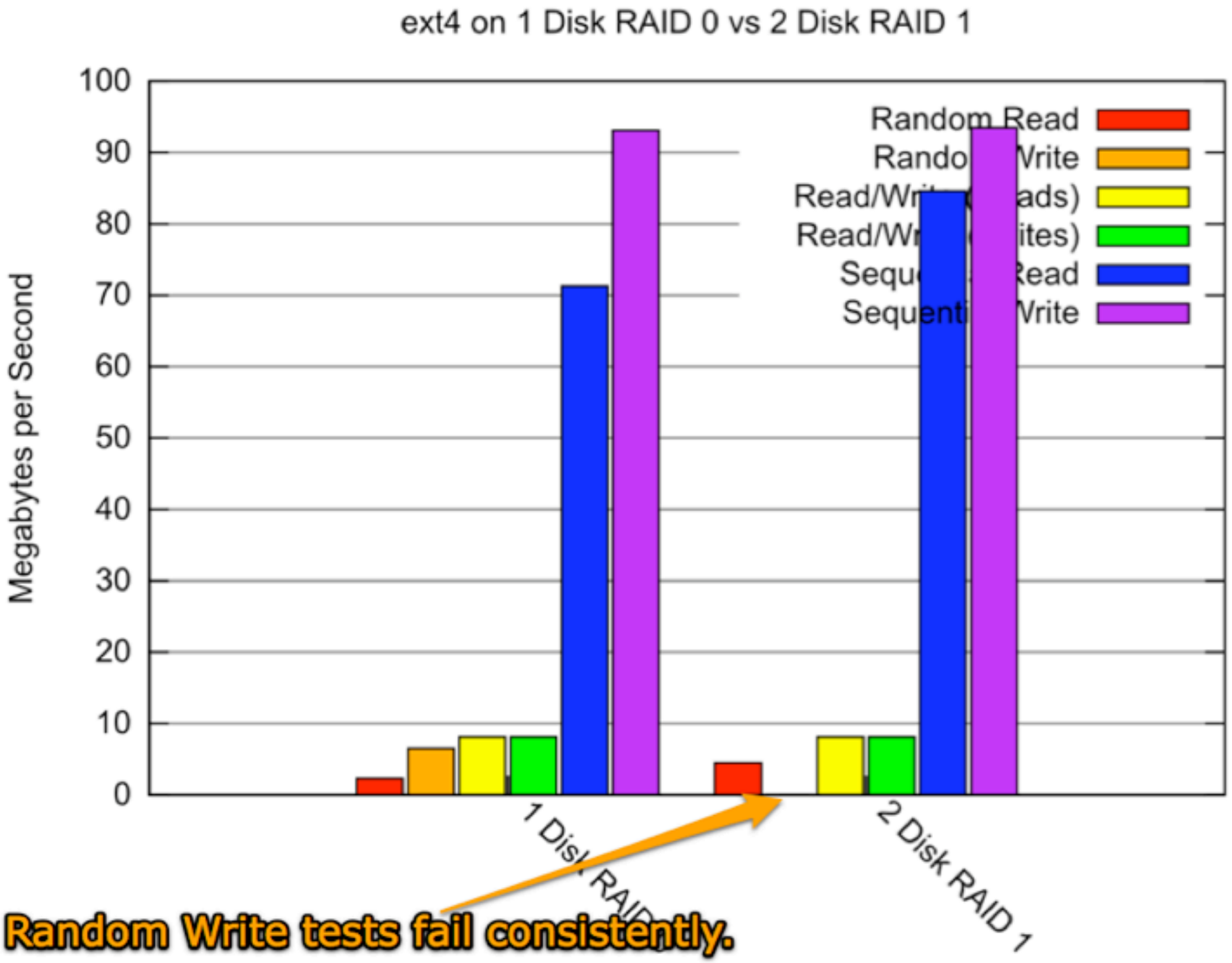

Random Writes
2 Disk RAID 0
Sequential Writes
Random Read
Random Write
Read/Write (Reads)
Read/Write (Writes)
Sequential Read
Sequential Write
Megabytes per Second
160
140
120
100
80
60
40
20
0
ext2
ext3
ext3-journal
ext3-writeback
jfs
reiserfs
xfs
ext4
Read/Write


HOW TO START A FIGHT
AUDIENCE PARTICIPATION
RAID 0 (stripe) versus
RAID 1 (mirroring)
performance?
PERFORMANCE
FIGHT CLUB


ext2 on 1 Disk RAID 0 vs 2 Disk RAID 1
Megabytes per Second
90
80
70
60
50
40
30
20
10
0
Rando Read
Rando Write
Read/Write ads)
Read/W ites)
Sequ Read
Sequenti Write
1 Disk RAID 0
2 Disk RAID 1
No significant change!

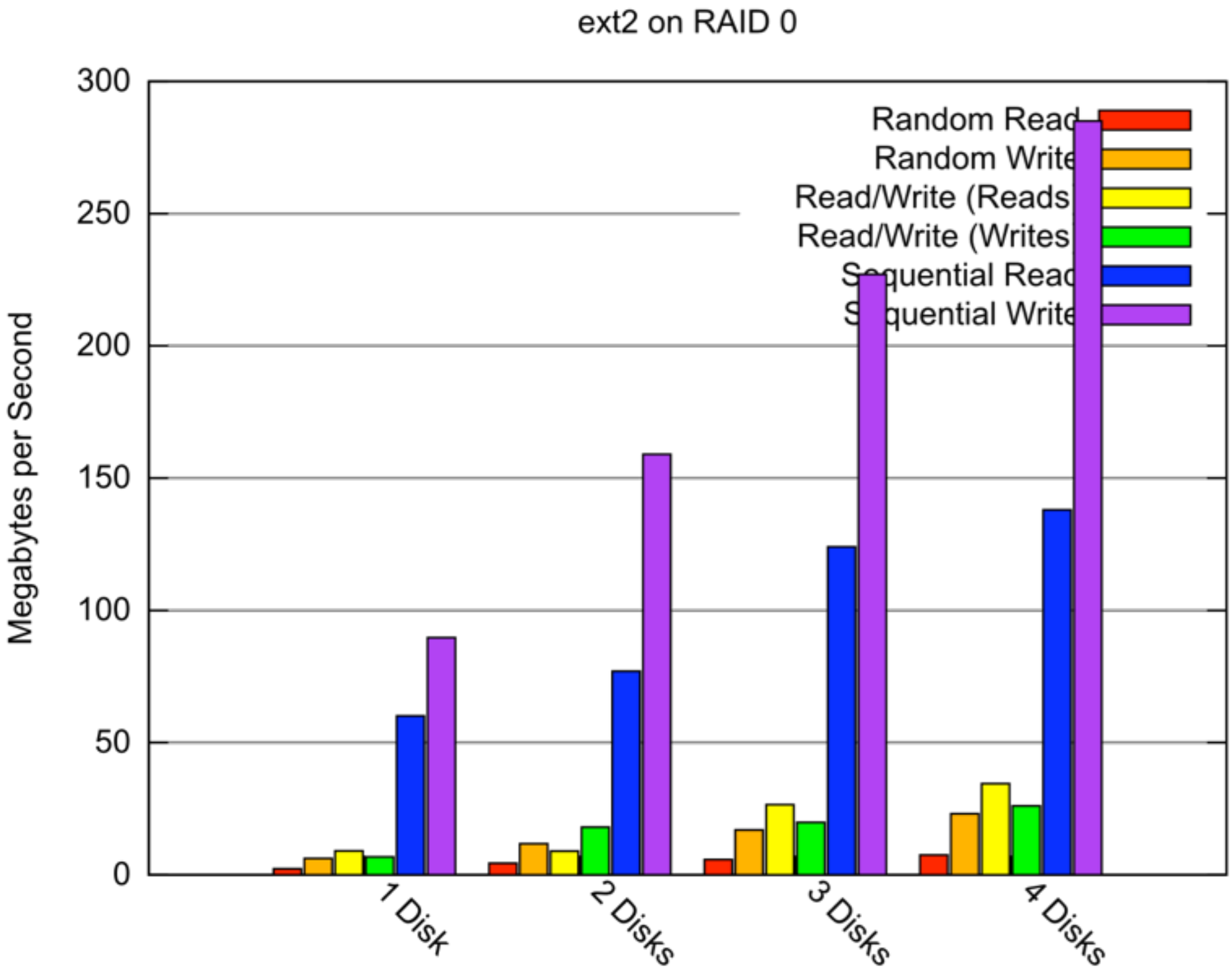


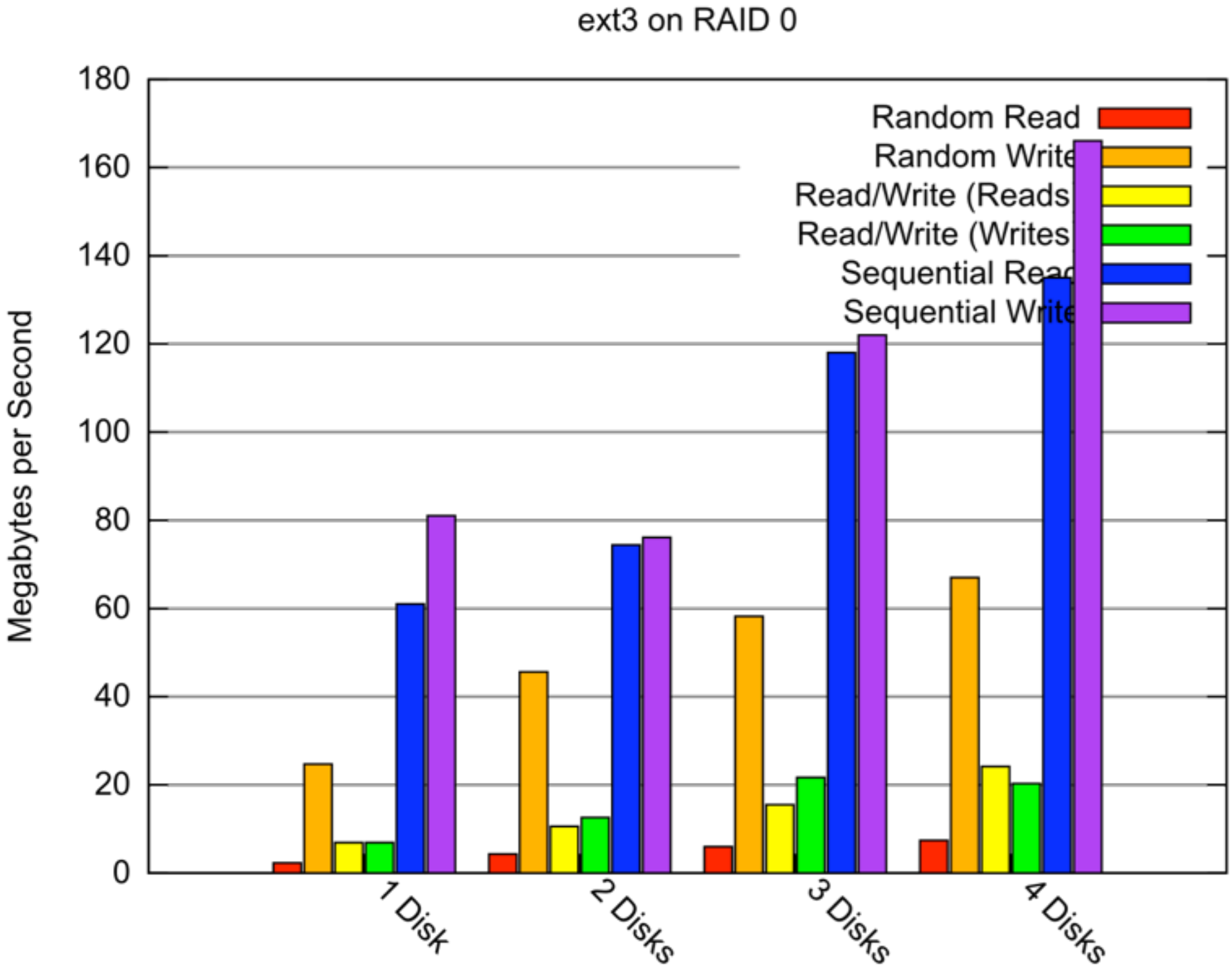
No significant degredation.

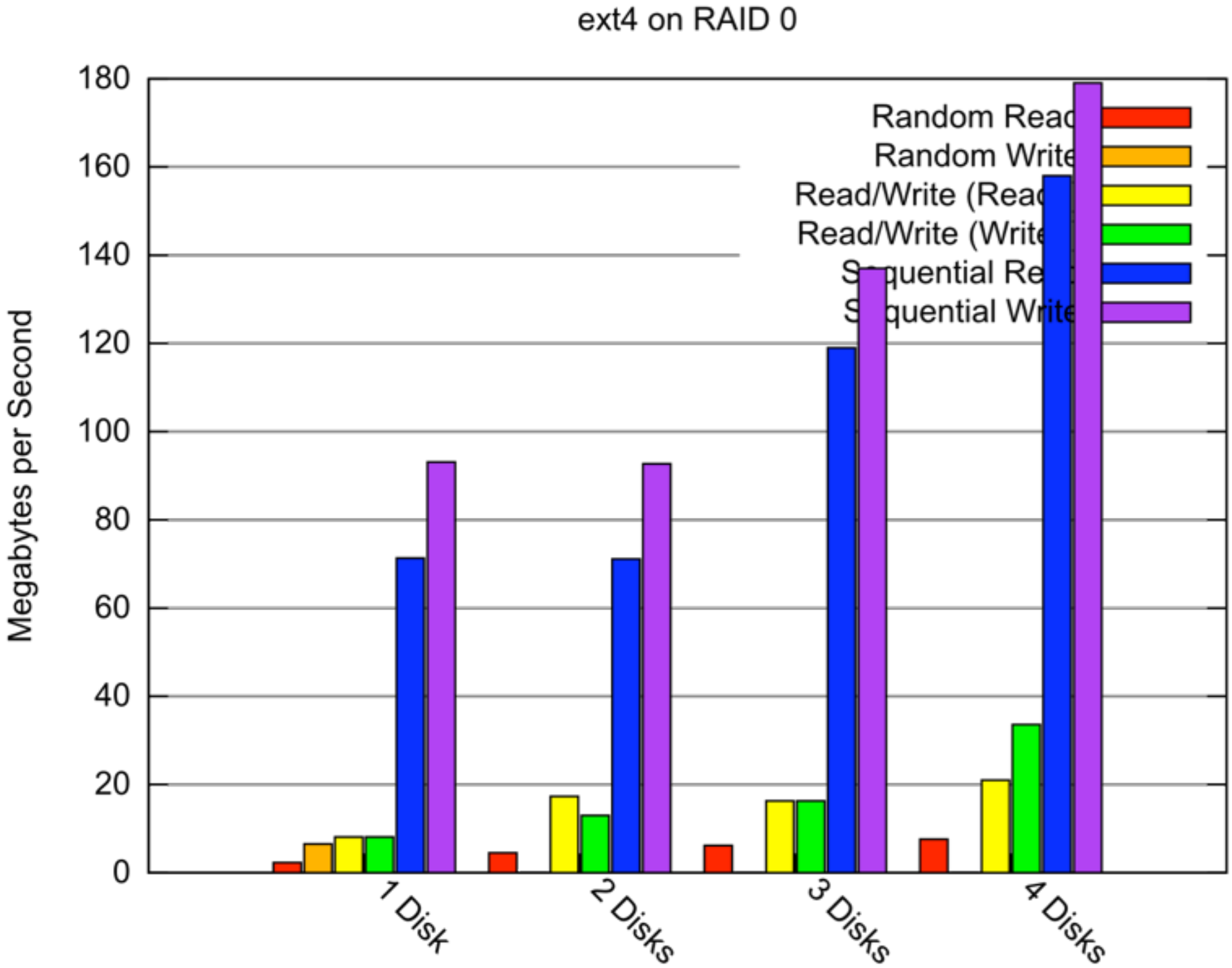

ext4 on 1 Disk RAID 0 vs 2 Disk RAID 1
Megabytes per Second
100
90
80
70
60
50
40
30
20
10
0
Random Read
Rando Write
Read/Wr ads)
Read/Wr ites)
Sequ Read
Sequenti Write
1 Disk RAID 0
2 Disk RAID 1
Random Write tests fail consistently.

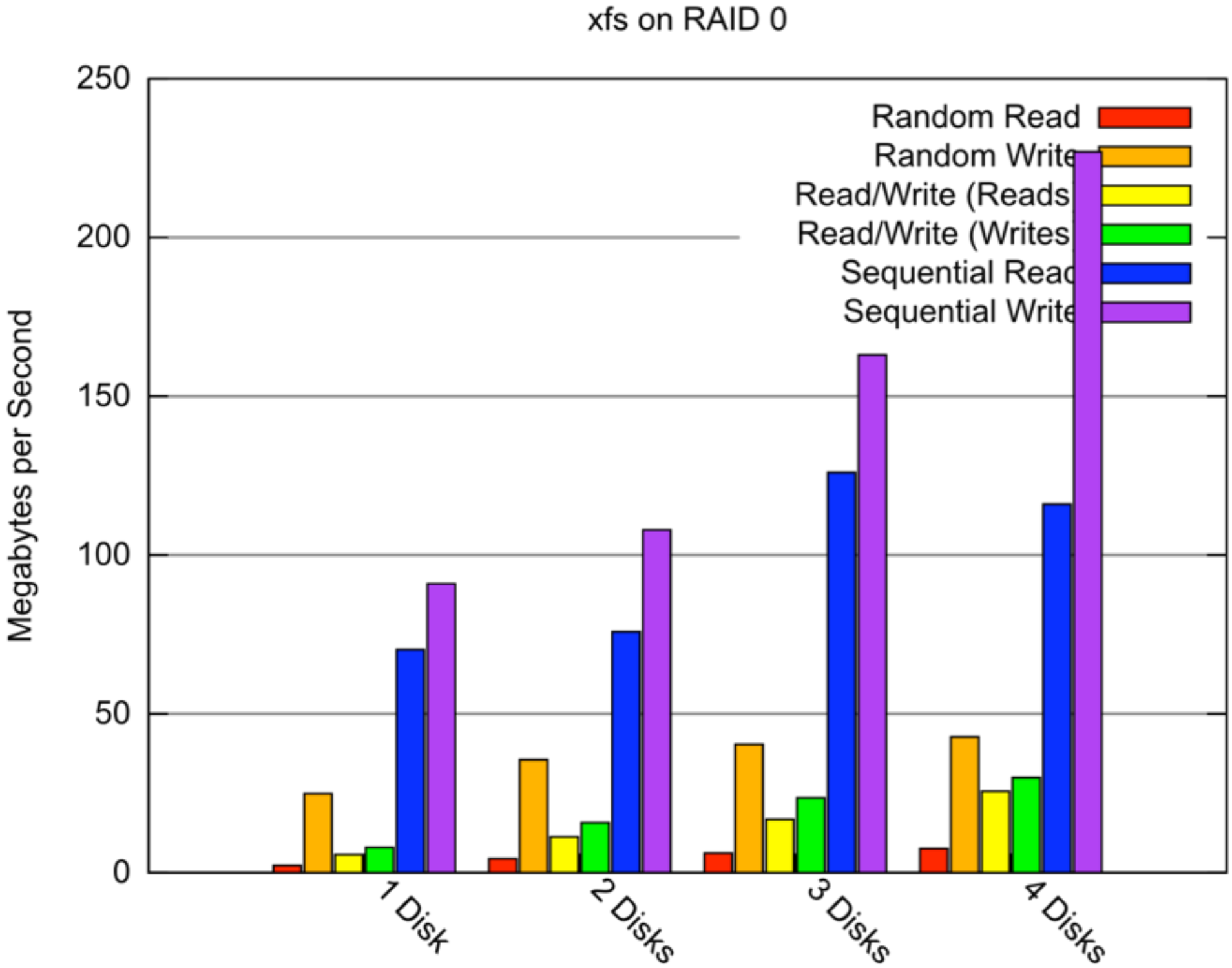
What happens when we:
add disks to a
RAID 0 (stripe) LUN?

~~PERFORMANCE~~

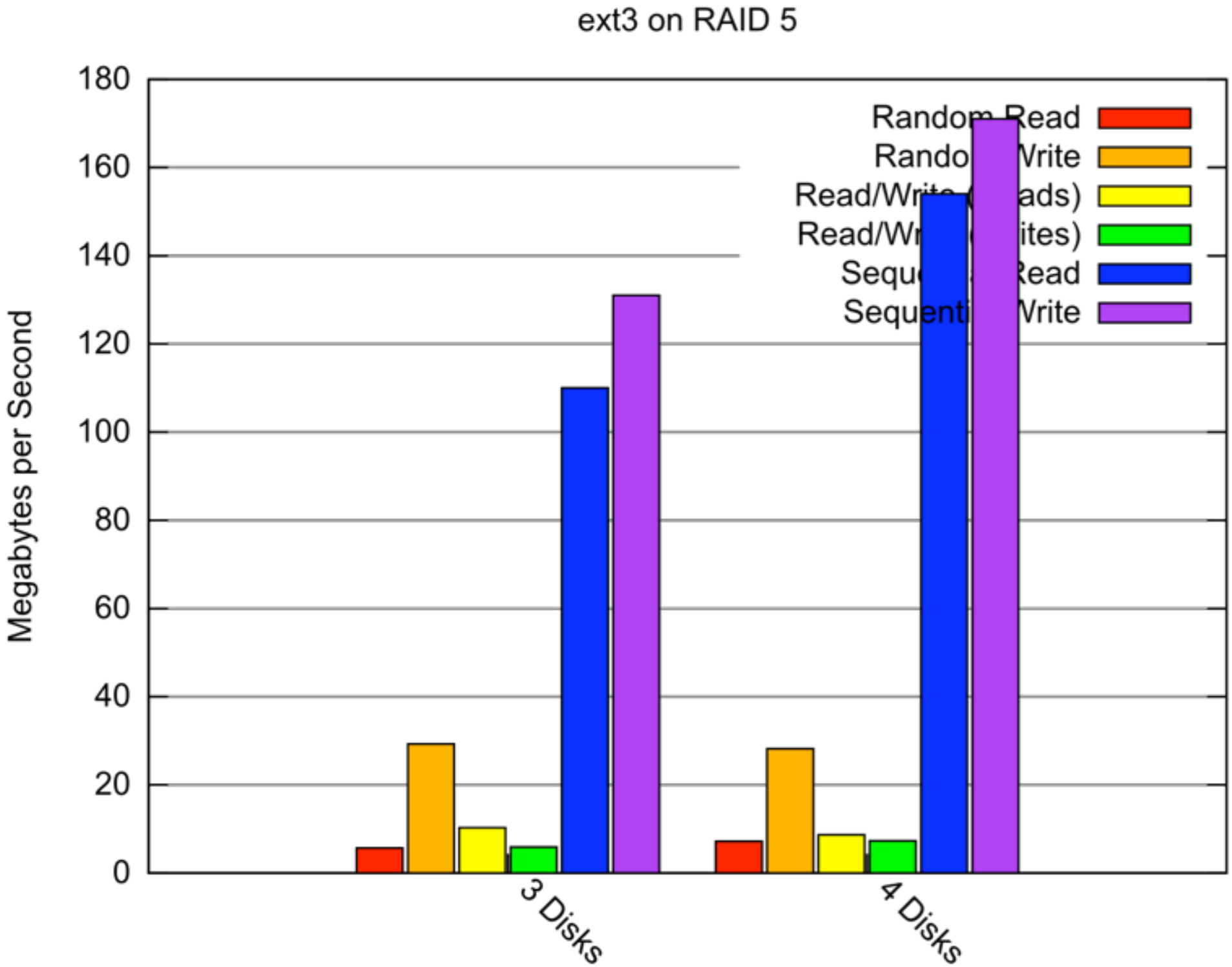

ext2 on RAID 0
Megabytes per Second
300
250
200
150
100
50
0
Random Read
Random Write
Read/Write (Reads
Read/Write (Writes
Sequential Read
Sequential Write
1 Disk
2 Disks
3 Disks
4 Disks

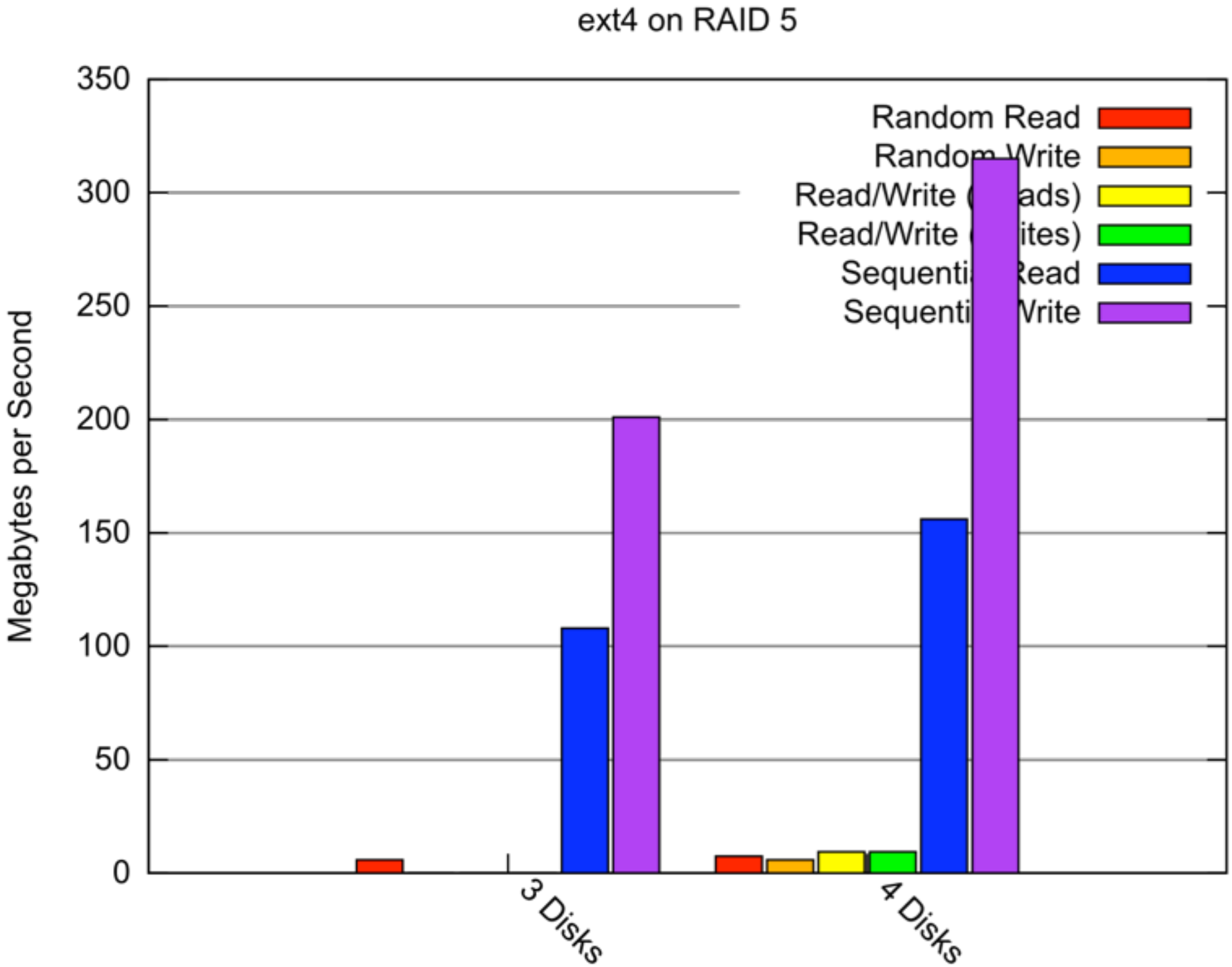

ext3 on RAID 0
Megabytes per Second
180
160
140
120
100
80
60
40
20
0
Random Read
Random Write
Read/Write (Reads
Read/Write (Writes
Sequential Read
Sequential Write
1 Disk
2 Disks
3 Disks
4 Disks

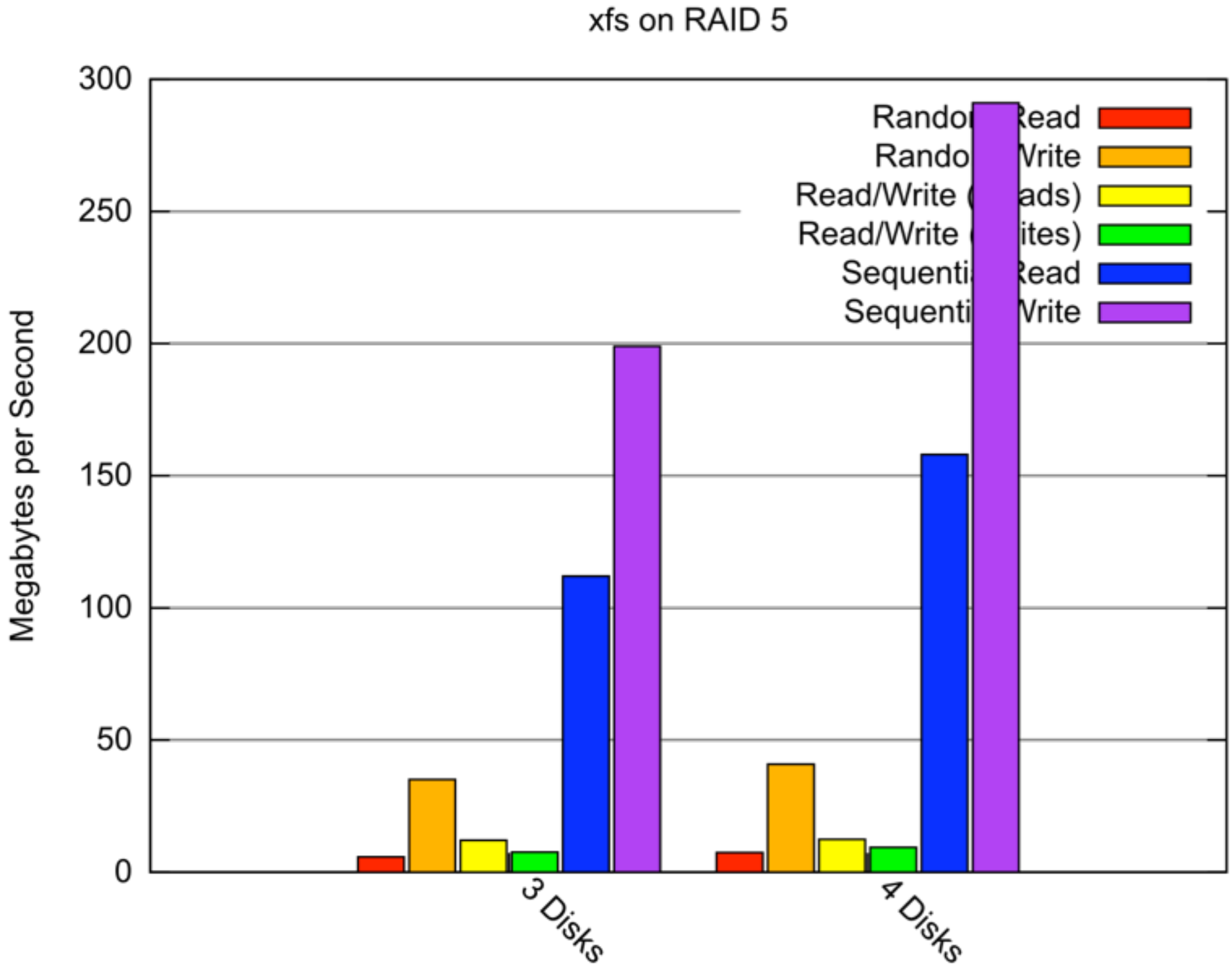

ext4 on RAID 0
Megabytes per Second
180
160
140
120
100
80
60
40
20
0
Random Read
Random Write
Read/Write (Read
Read/Write (Write
Sequential Read
Sequential Write
1 Disk
2 Disks
3 Disks
4 Disks


xfs on RAID 0
Megabytes per Second
250
200
150
100
50
0
Random Read
Random Write
Read/Write (Reads
Read/Write (Writes
Sequential Read
Sequential Write
1 Disk
2 Disks
3 Disks
4 Disks

~~PERFORMANCE~~


ext3 on RAID 5
Megabytes per Second
180
160
140
120
100
80
60
40
20
0
Random Read
Rando Write
Read/Wri ads)
Read/Wr ites)
Sequ Read
Sequenti Write
3 Disks
4 Disks

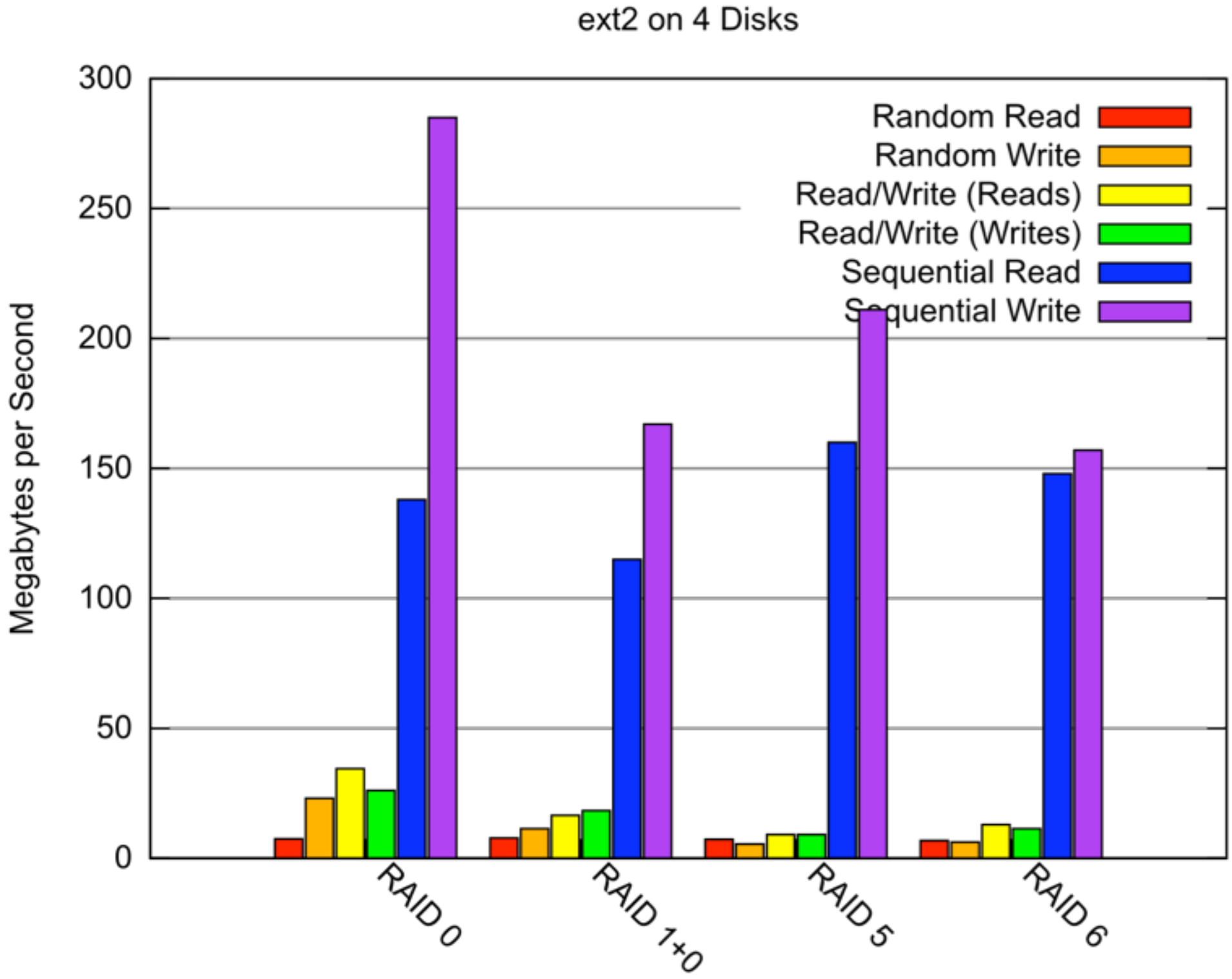

ext4 on RAID 5
Megabytes per Second
350
300
250
200
150
100
50
0
Random Read
Random Write
Read/Write (Reads)
Read/Write (Writes)
Sequential Read
Sequential Write
3 Disks
4 Disks

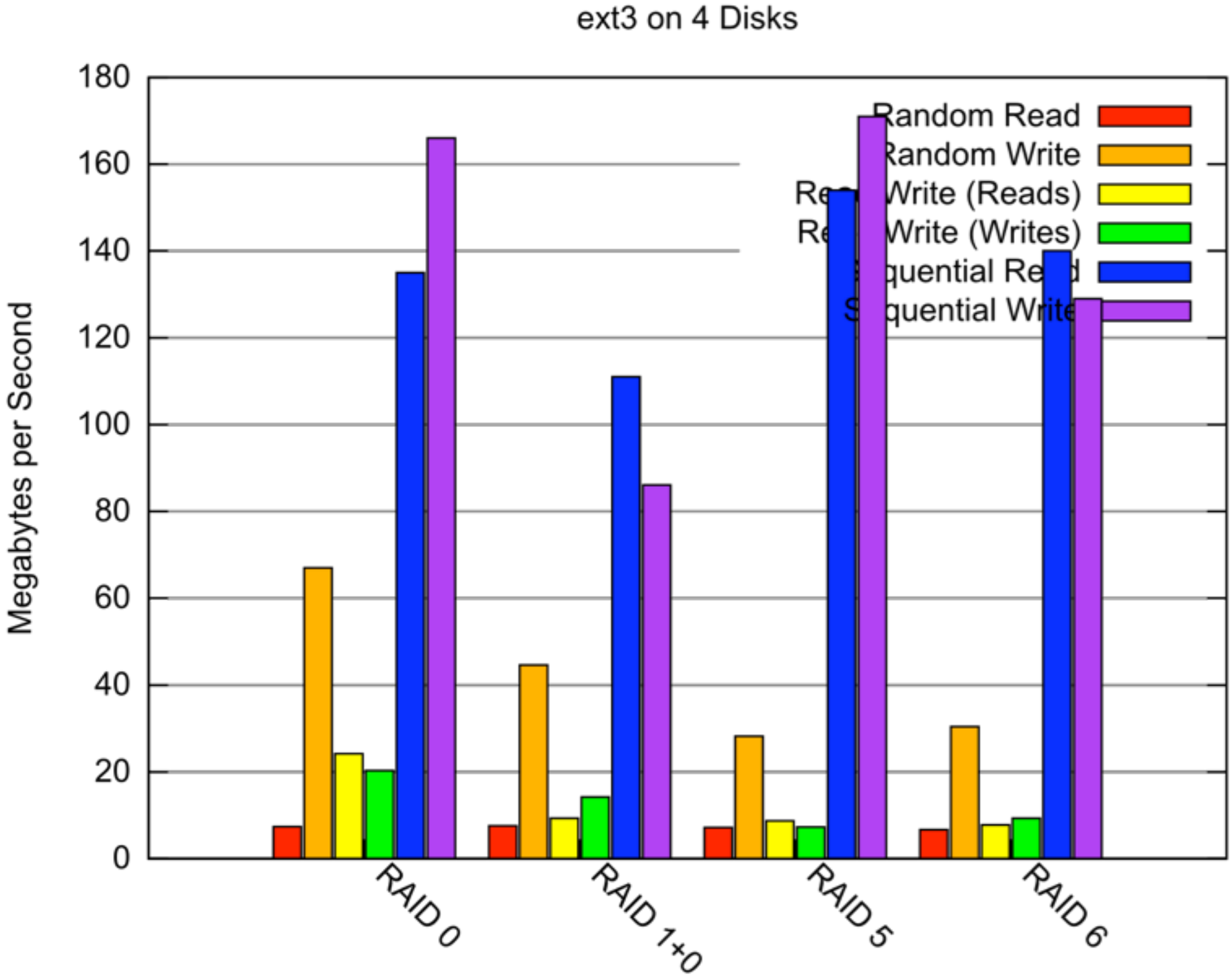

xfs on RAID 5
Megabytes per Second
300
250
200
150
100
50
0
Rando Read
Rando Write
Read/Write (ads)
Read/Write (ites)
Sequenti Read
Sequenti Write
3 Disks
4 Disks

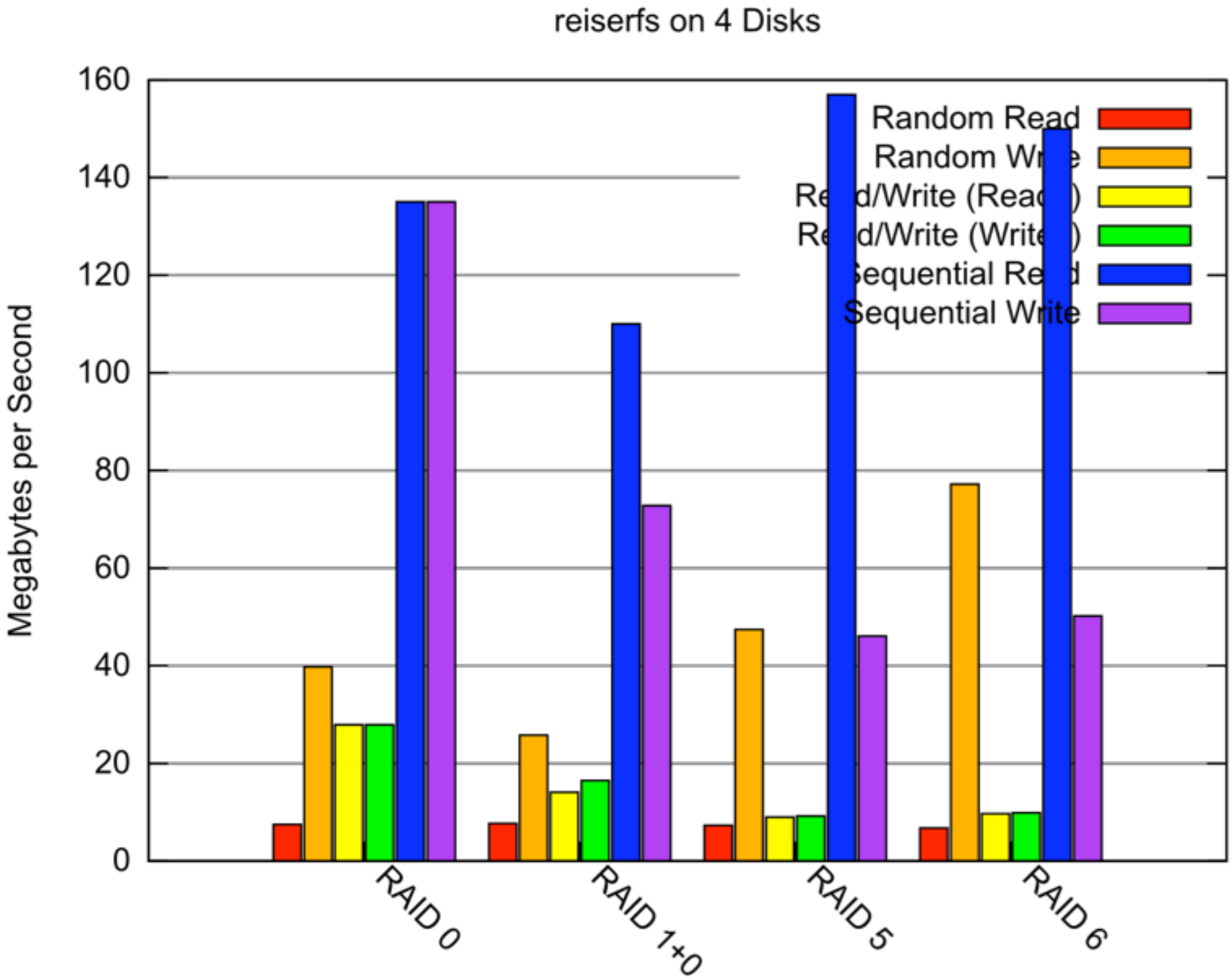
Only have 4 disks?
What should you do?

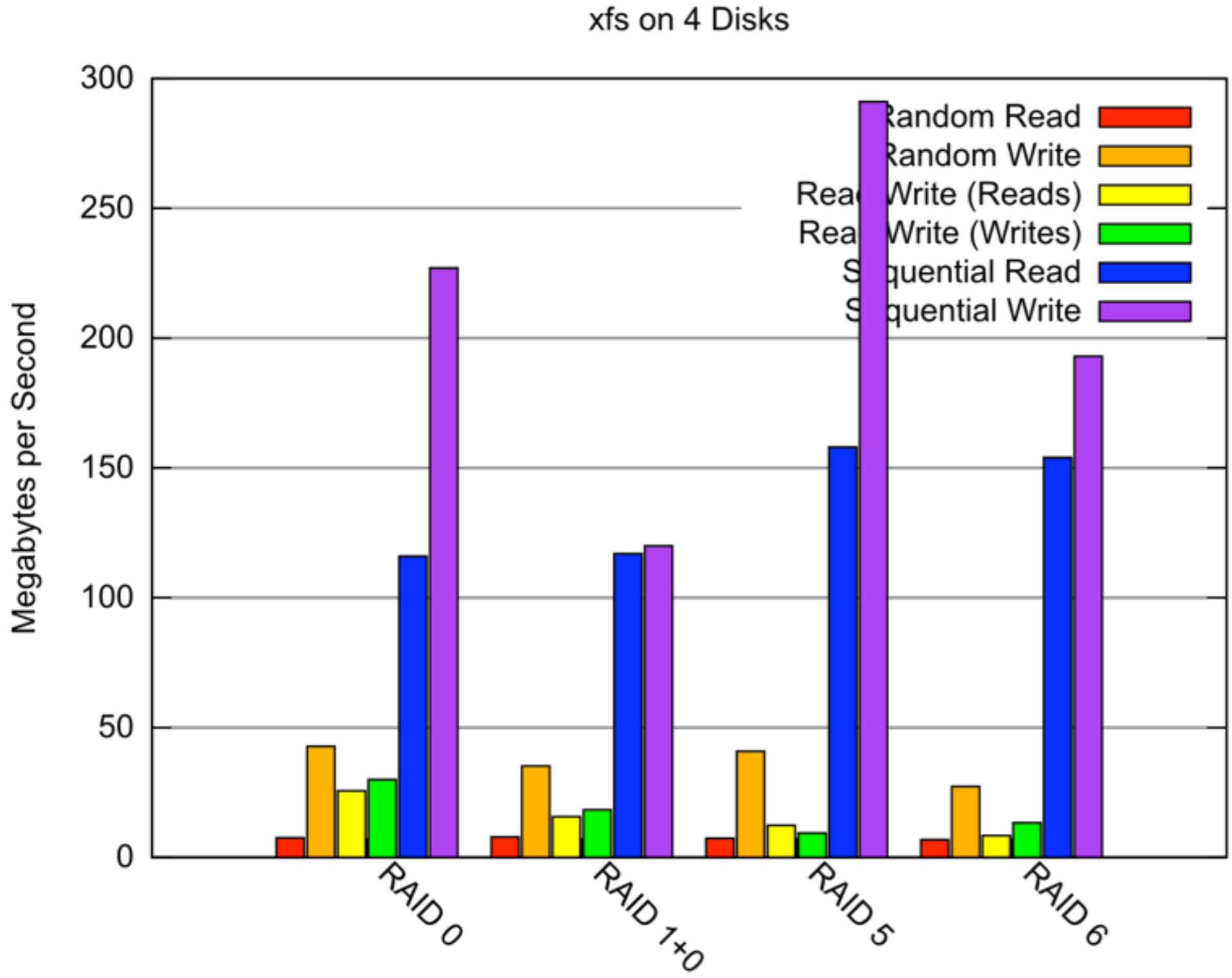
~~FIGHT CLUB~~

~~PERFORMANCE~~


ext2 on 4 Disks
Megabytes per Second
300
250
200
150
100
50
0
Random Read
Random Write
Read/Write (Reads)
Read/Write (Writes)
Sequential Read
Sequential Write
RAID 0
RAID 1+0
RAID 5
RAID 6


ext3 on 4 Disks
Megabytes per Second
180
160
140
120
100
80
60
40
20
0
Random Read
Random Write
Write (Reads)
Write (Writes)
quential Re
quential Write
RAID 0
RAID 1+0
RAID 5
RAID 6


reiserfs on 4 Disks
Megabytes per Second
160
140
120
100
80
60
40
20
0
Random Read
Random Write
Read/Write (Read)
Read/Write (Write)
Sequential Read
Sequential Write
RAID 0
RAID 1+0
RAID 5
RAID 6

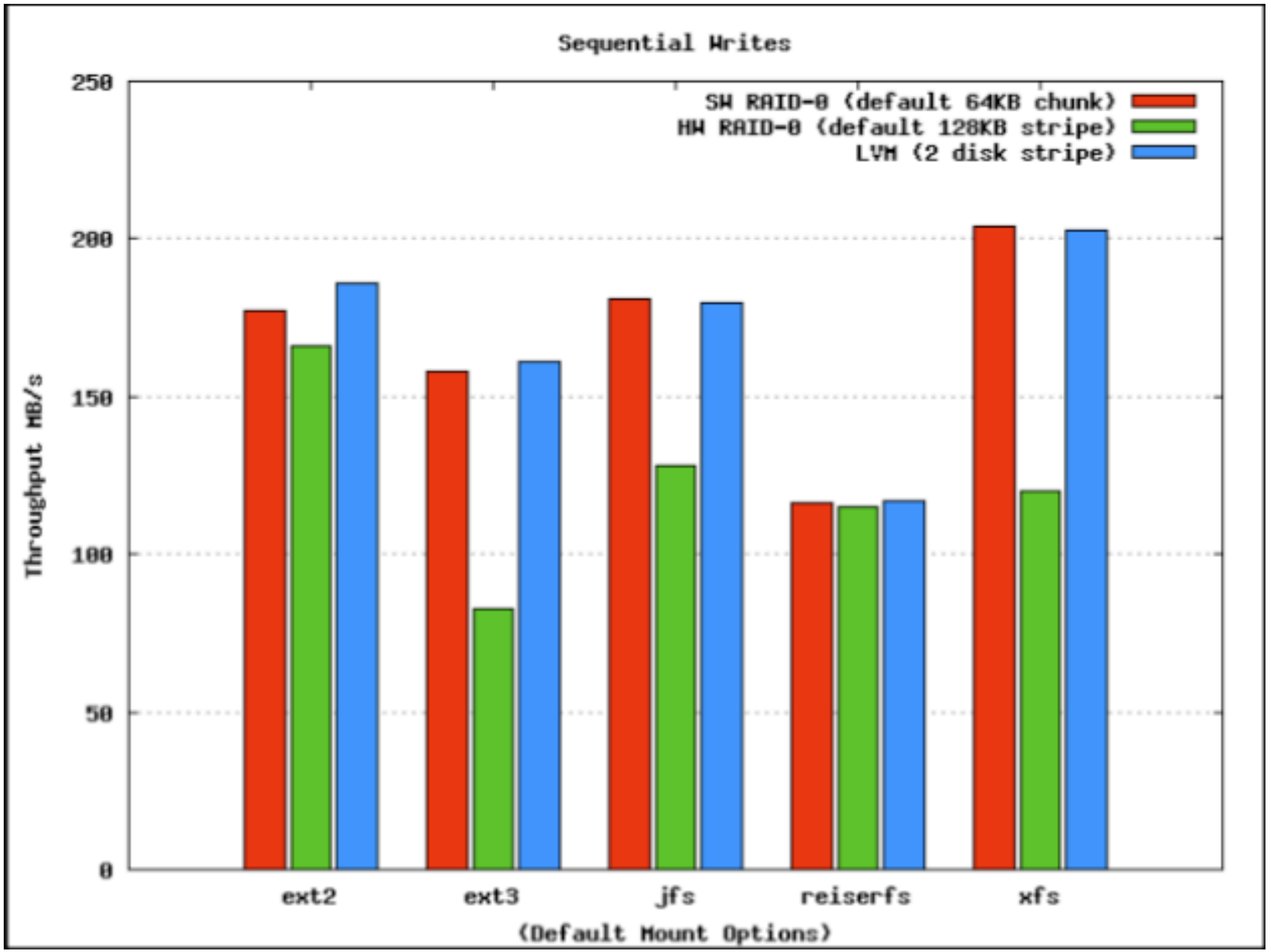

xfs on 4 Disks
Megabytes per Second
300
250
200
150
100
50
0
Random Read
Random Write
Write (Reads)
Write (Writes)
quential Read
quential Write
RAID 0
RAID 1+0
RAID 5
RAID 6

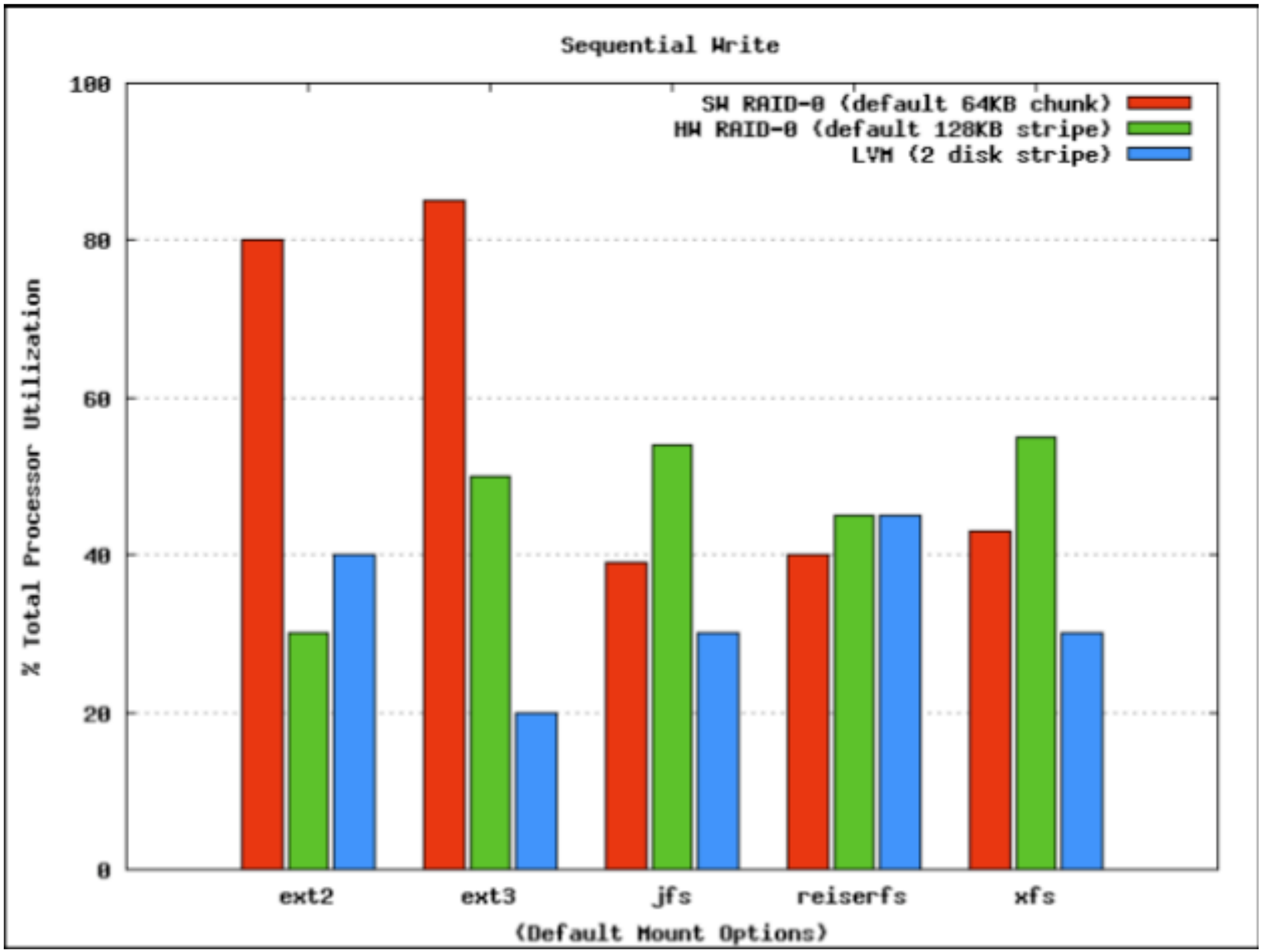
In most cases, RAID 5 out-performs on sequential writes (xlog).

Random writes is only an improvement on xfs and reiserfs.

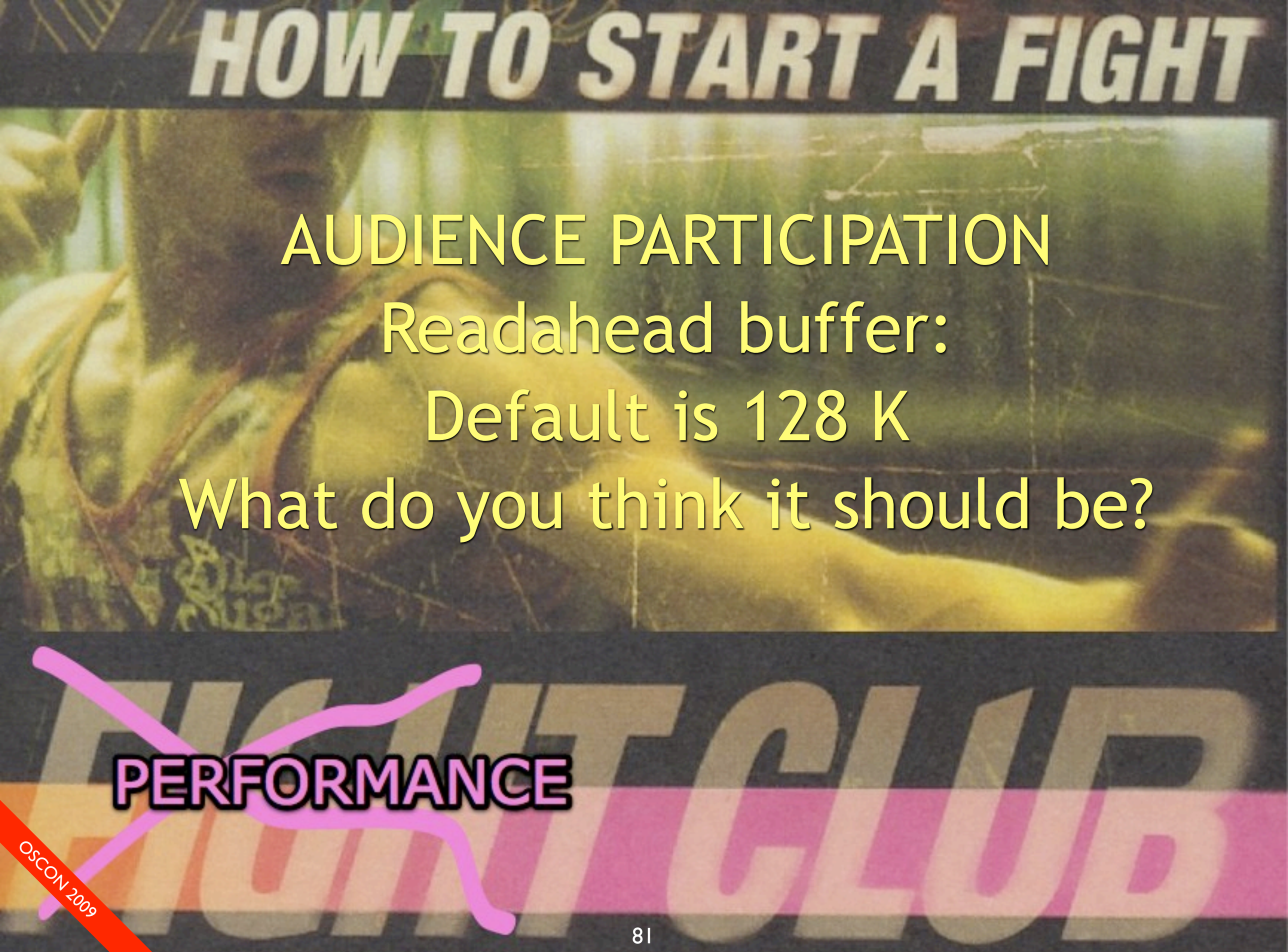
Are software RAID and LVM are slow?

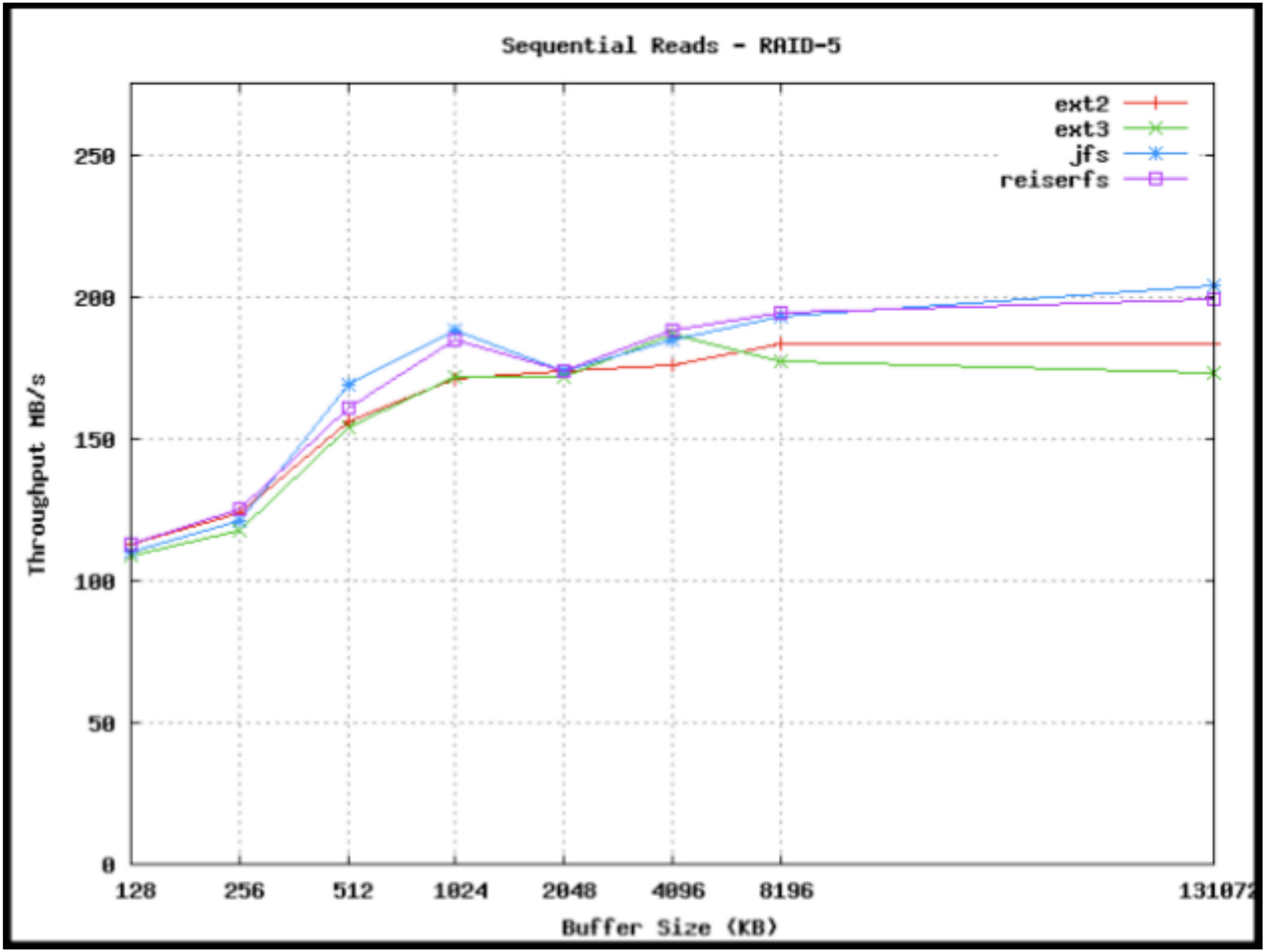
~~PERFORMANCE~~


Sequential Writes
SW RAID-0 (default 64KB chunk)
HW RAID-0 (default 128KB stripe)
LVM (2 disk stripe)
Throughput MB/s
250
200
150
100
50
0
ext2
ext3
jfs
reiserfs
xfs
(Default Mount Options)

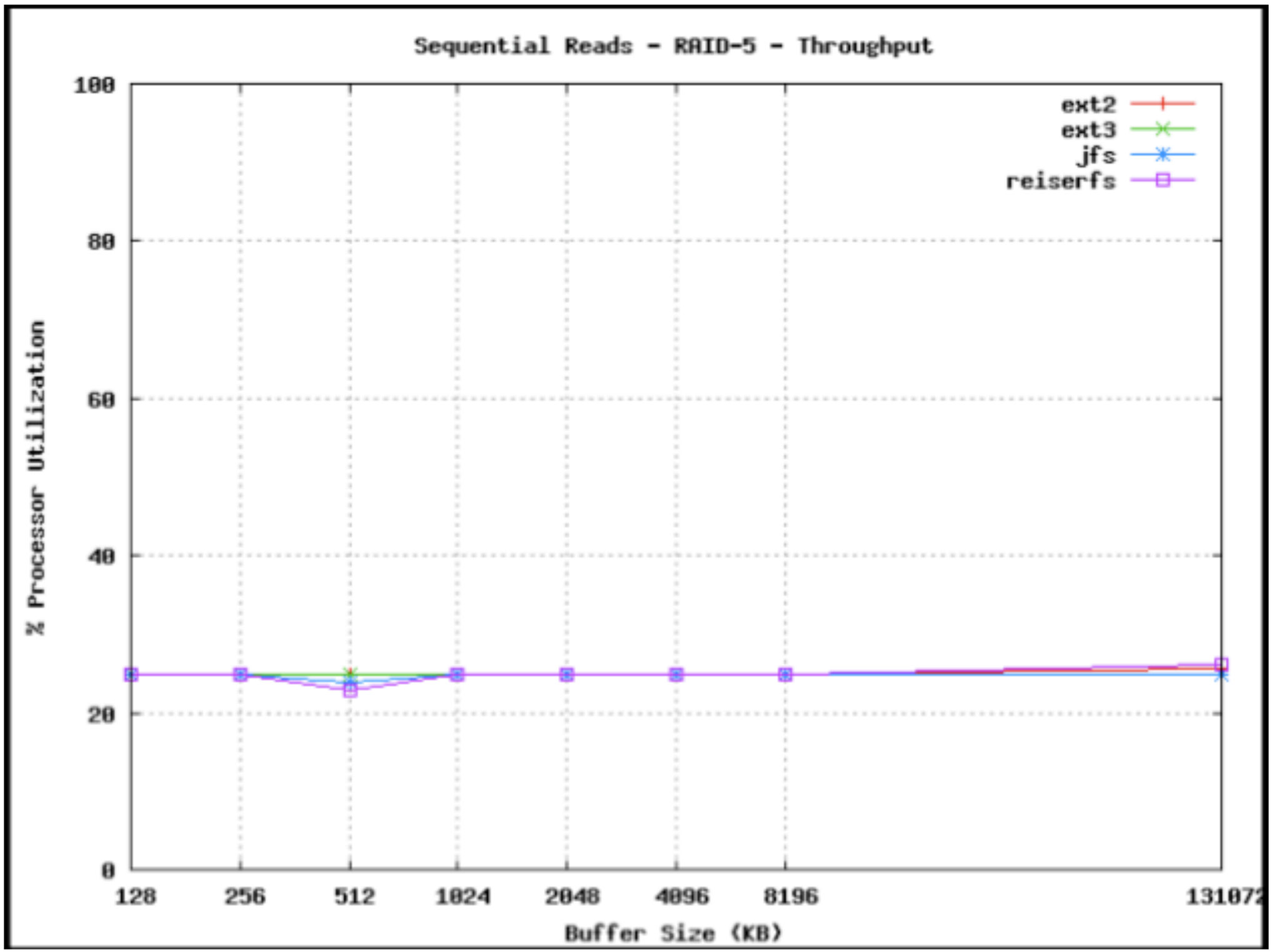

Sequential Write
SW RAID-0 (default 64KB chunk)
HW RAID-0 (default 128KB stripe)
LVM (2 disk stripe)
% Total Processor Utilization
100
80
60
40
20
0
ext2
ext3
jfs
reiserfs
xfs
(Default Mount Options)

The Read-ahead buffer

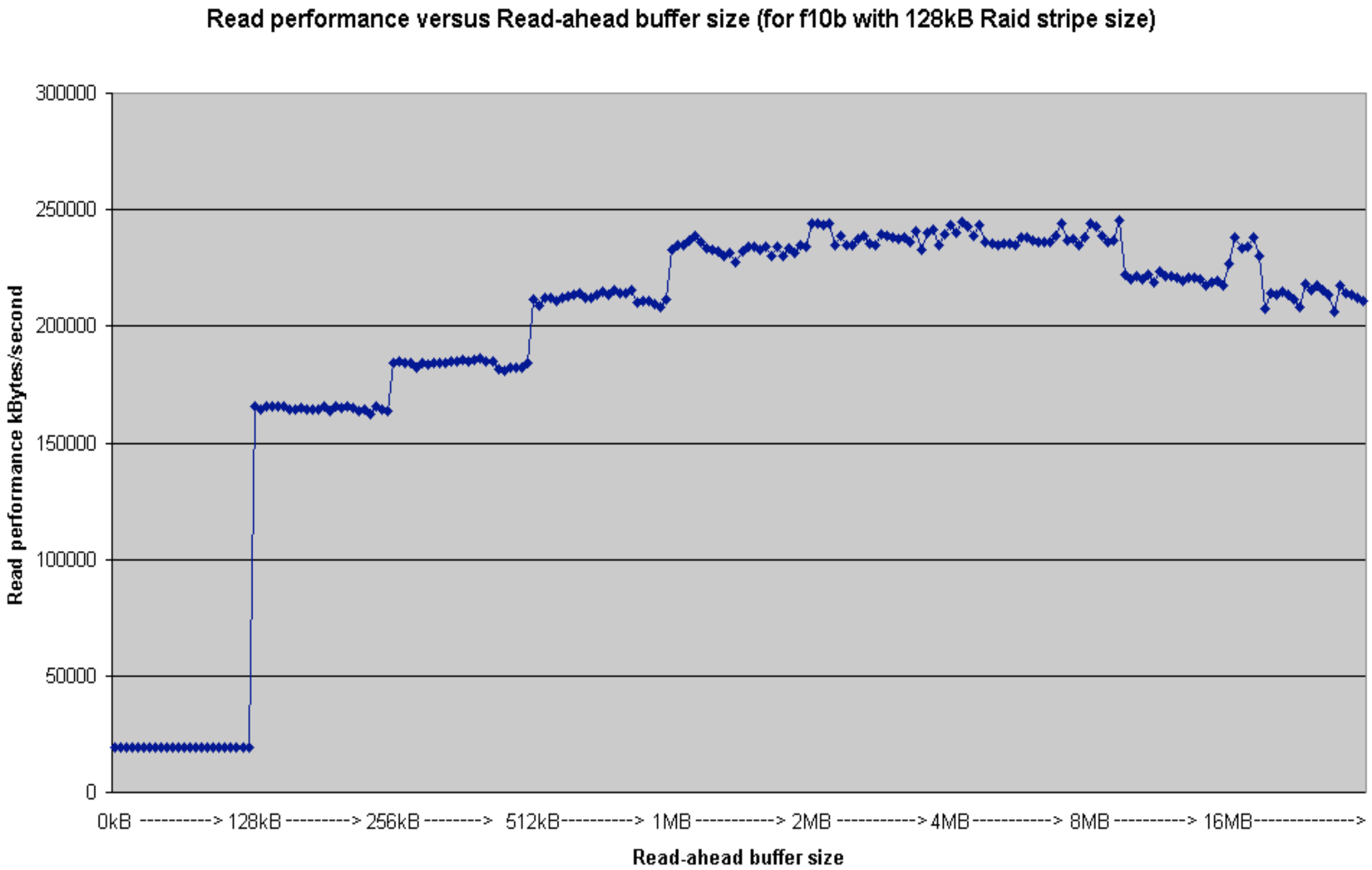

HOW TO START A FIGHT
AUDIENCE PARTICIPATION
Readahead buffer:
Default is 128 K
What do you think it should be?
FIGHT CLUB
PERFORMANCE
OSCON 2009


Sequential Reads - RAID-5
ext2
ext3
jfs
reiserfs
250
200
150
100
50
0
Throughput MB/s
128
256
512
1024
2048
4096
8196
131072
Buffer Size (KB)

And is there a cost to increasing the buffer that much?


Sequential Reads - RAID-5 - Throughput
% Processor Utilization
100
80
60
40
20
0
128
256
512
1024
2048
4096
8196
131072
Buffer Size (KB)
ext2
ext3
jfs
reiserfs

http://moourl.com/readaheadconfirm



OLTP workload

- DBT-2 toolkit
  (Fair-use derivative of TPC-C)
- Used 35 drives ultimately
- pgtune:
  http://pgfoundry.org/projects/pgtune/

Out of the Box - From a 25 disk RAID 0 device[1]

Transaction	%	Response Time (s) Average :	90th %	Total	Rollbacks	%
Delivery	3.99	11.433 :	12.647	45757	0	0.00
New Order	45.24	10.257 :	11.236	518945	5224	1.02
Order Status	4.00	9.998 :	11.023	45926	0	0.00
Payment	42.81	9.983 :	11.022	491102	0	0.00
Stock Level	3.95	9.855 :	10.837	45344	0	0.00

```
8574.99 new-order transactions per minute (NOTPM)
59.3 minute duration
0 total unknown errors
1041 second(s) ramping up
```

This result is from before we ran pgtune to show if it'll help.

[1] http://207.173.203.223/~markwkm/community6/dbt2/baseline.1000.2/

pgtune - From a 25 disk RAID 0 device[2]

Transaction	%	Response Time (s) Average :	90th %	Total	Rollbacks	%
Delivery	3.99	8.715 :	10.553	48961	0	0.00
New Order	45.22	8.237 :	9.949	554565	5425	0.99
Order Status	3.95	8.037 :	9.828	48493	0	0.00
Payment	42.84	8.026 :	9.795	525387	0	0.00
Stock Level	3.99	7.829 :	9.563	48879	0	0.00

```
9171.46 new-order transactions per minute (NOTPM)
59.3 minute duration
0 total unknown errors
1041 second(s) ramping up
```

This result is from after running pgtune 0.3.

[2] http://207.173.203.223/~markwkm/community6/dbt2/pgtune.1000.100.3/

7% improvement! :)

For more info...

- See Mark Wong's blog:
  http://pugs.postgresql.org/blog/92
- Takeaway: for DBT-2, increasing checkpoint_segments had the largest impact (fewer checkpoints :)

Future Work

- OLTP system characterization, sizing (ongoing)
- Daily OLTP regression testing
- More presentations
- P5 - PostgreSQL Portland Performance Pad PRACTICE (done!)

MOAR Hardware?

Thanks again, HP!
MSA70, DL380 in late 2009 ??

Let’s recap…

“RAID5 is the worst choice for a database.” Fast for sequential writes in our tests.

“LVM incurs too much overhead to use. Software RAID is slower.” For reads – throughput is about the same, but saw higher CPU.

“Turning off 'atime' is a big performance gain.” Not in our tests. But, 2-3% for “free”.

“Journaling filesystems will have worse performance than non-journaling filesystems.” Turn the data journaling off on ext3, and you do see better performance, but there are edge cases and performance differences we could not explain.

“Striping doubles performance.” Performance is better, but no where near double. Why?

“Your read-ahead buffer is big enough.”

Your read-ahead buffer IS NOT big enough. Make it 8MB. And can we make that the default?

Thank you!

Results:

http://wiki.postgresql.org/wiki/HP_ProLiant_DL380_G5_Tuning_Guide

http://moourl.com/fsperf

Selena Deckelmann
selena@postgresql.org
twitter: @selenamarie