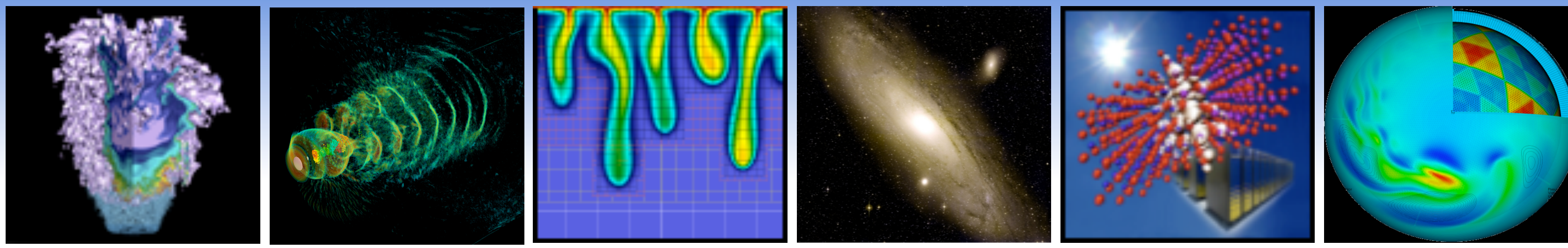


File System Monitoring as a Window Into User I/O Requirements

Andrew Uselton

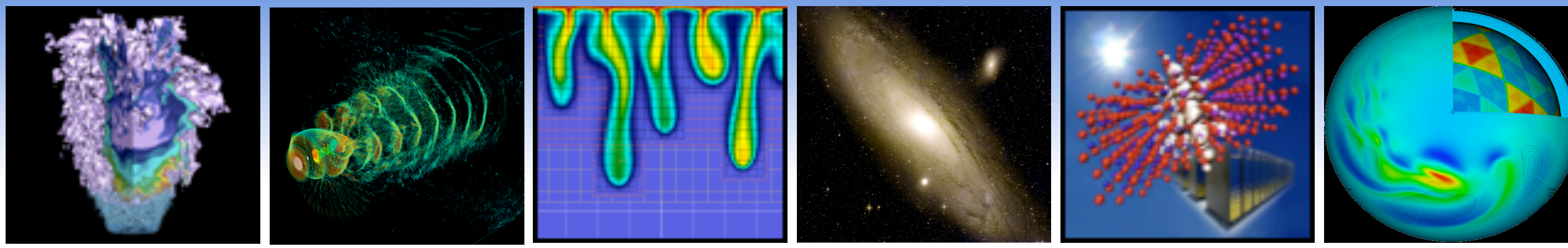
NERSC/Lawrence Berkeley Lab

May 26, 2010

U.S. DEPARTMENT OF ENERGY | Office of Science

NERSC

BERKELEY LAB



File System Monitoring as a Window Into User I/O Requirements

with

Katie Antypas, NERSC/Lawrence Berkeley Lab

Daniela Ushizima, CRD/Larwence Berkeley Lab

Jeff Sukharov, Univ of California, Davis

May 26, 2010

Overview

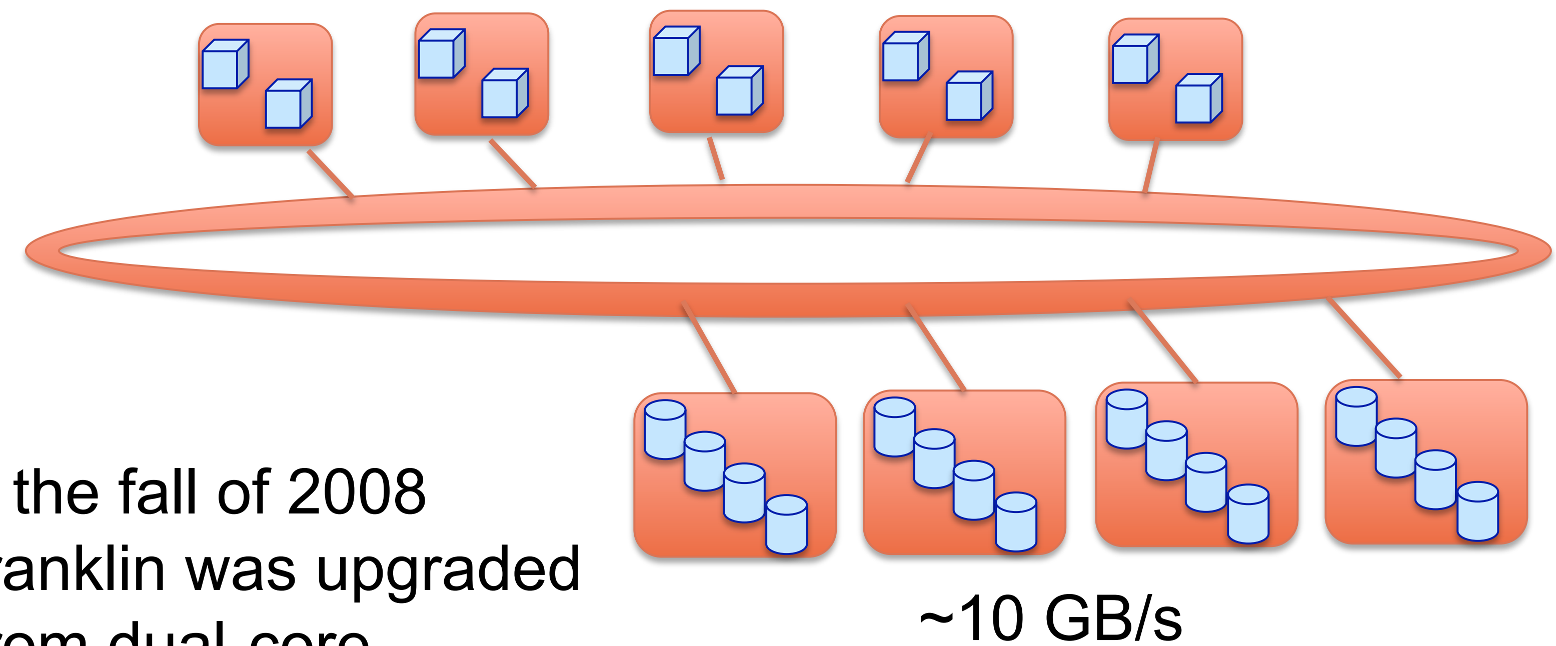
- Franklin's (NERSC's Cray XT) I/O subsystem was upgraded in Spring 2009 from 10 GB/s to 2 x 17 GB/s [Antypas, CUG'09]

- On the resulting two file systems the "Big I/O" users were directed to */scratch2*

- We now have a year's data on the workload characteristics to document how the workload differs on the two file systems.

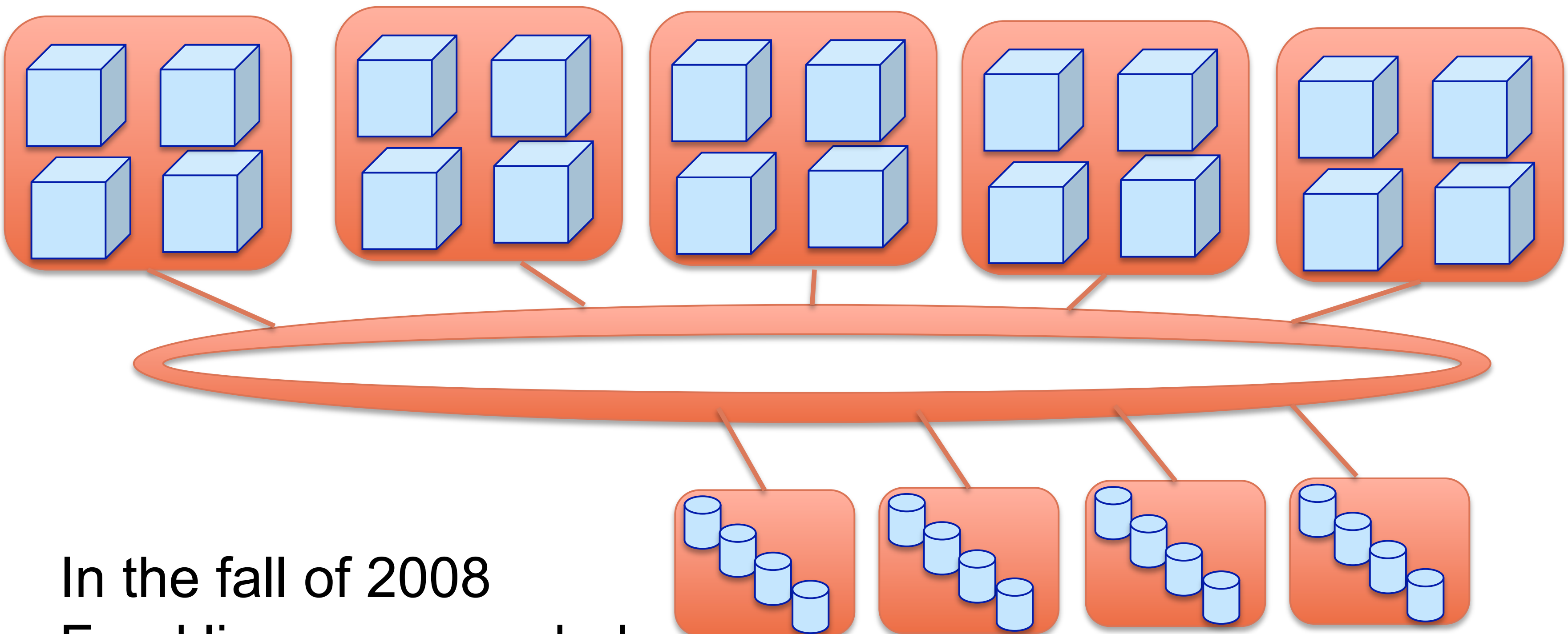
- We have the beginnings of a framework for rigorously evaluating the effectiveness of dividing the NERSC workload across the two file systems.

Franklin compute nodes were upgraded



In the fall of 2008
Franklin was upgraded
From dual-core
to quad-core

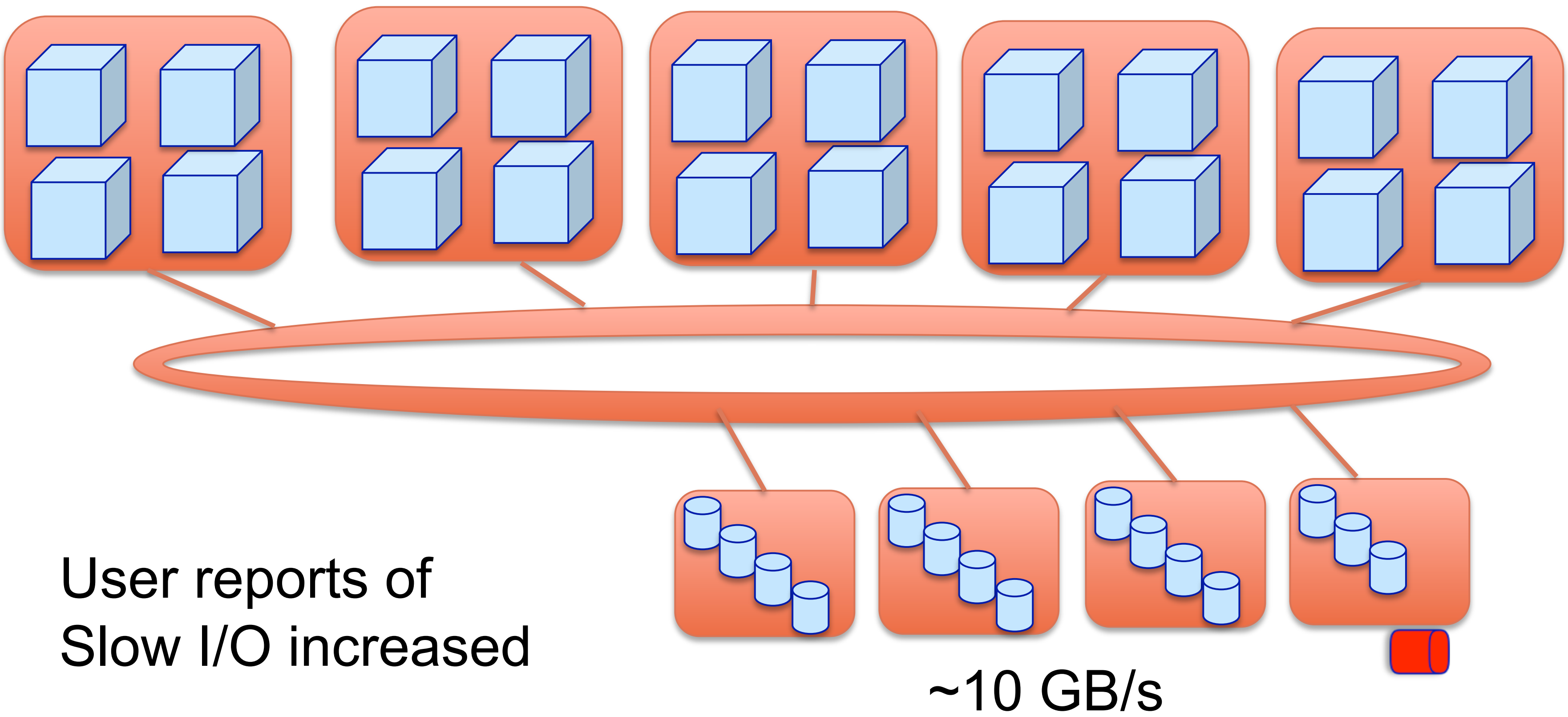
Franklin's compute nodes were upgraded



In the fall of 2008
Franklin was upgraded
From dual-core
to quad-core

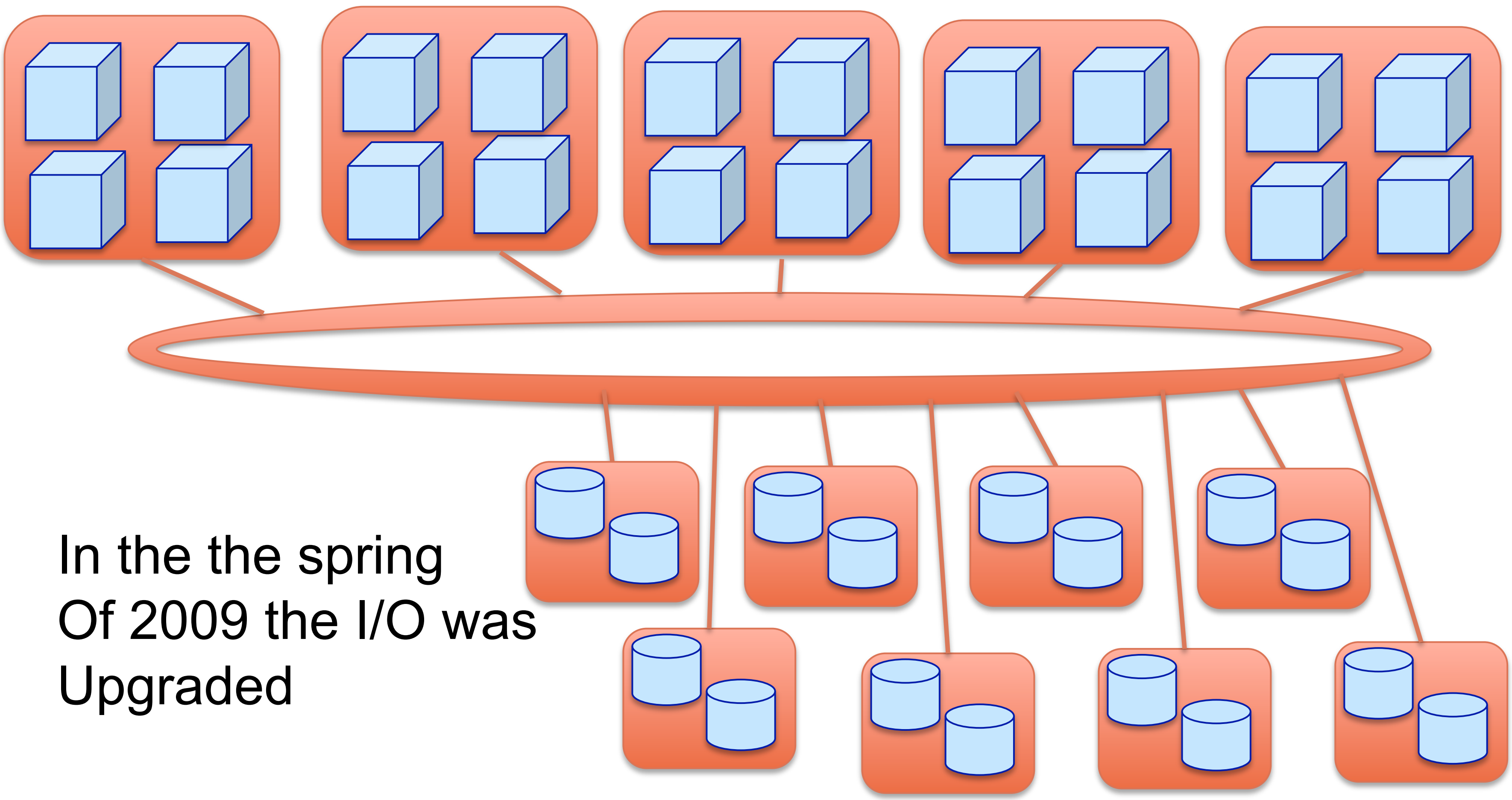
~10 GB/s

I/O appeared to be under-provisioned



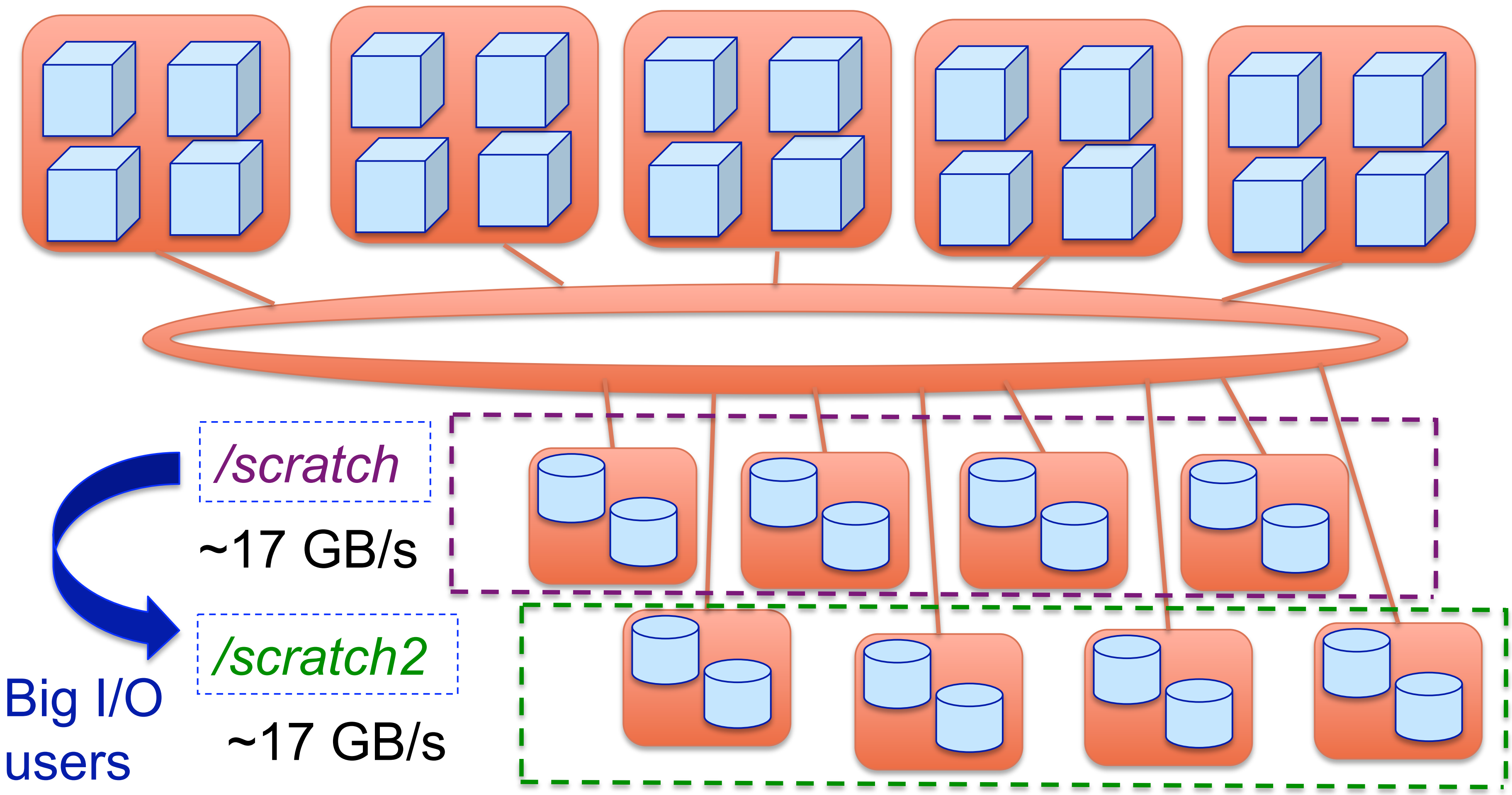
User reports of
Slow I/O increased

I/O was upgraded



In the the spring
Of 2009 the I/O was
Upgraded

Creating two scratch file systems

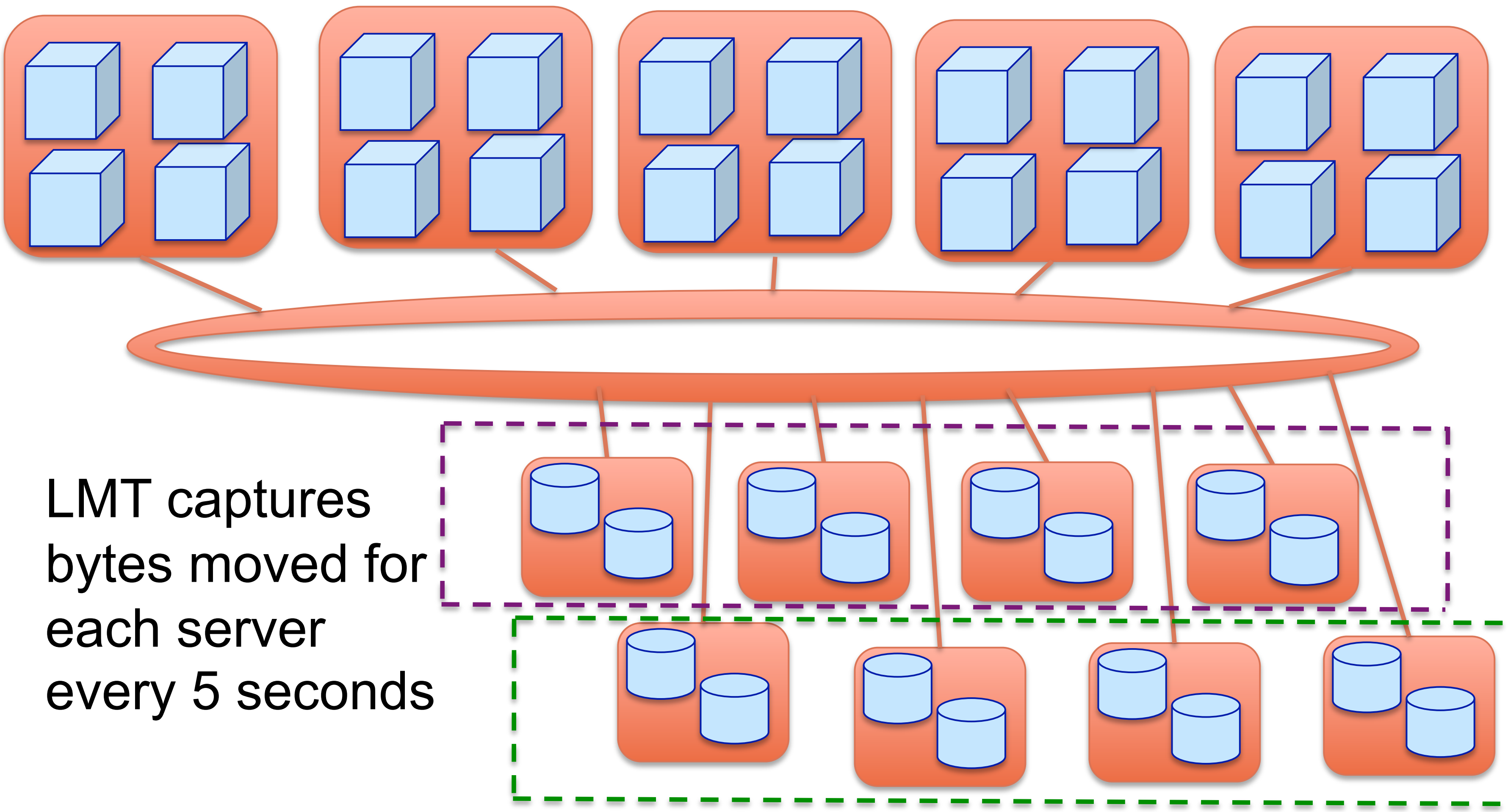


Gathering I/O Statistics

•LMT – server side I/O performance monitoring

• IOR – Parallel file system I/O performance benchmark

The Lustre Monitoring Tool: detailed server I/O data [Uselton, CUG'09]

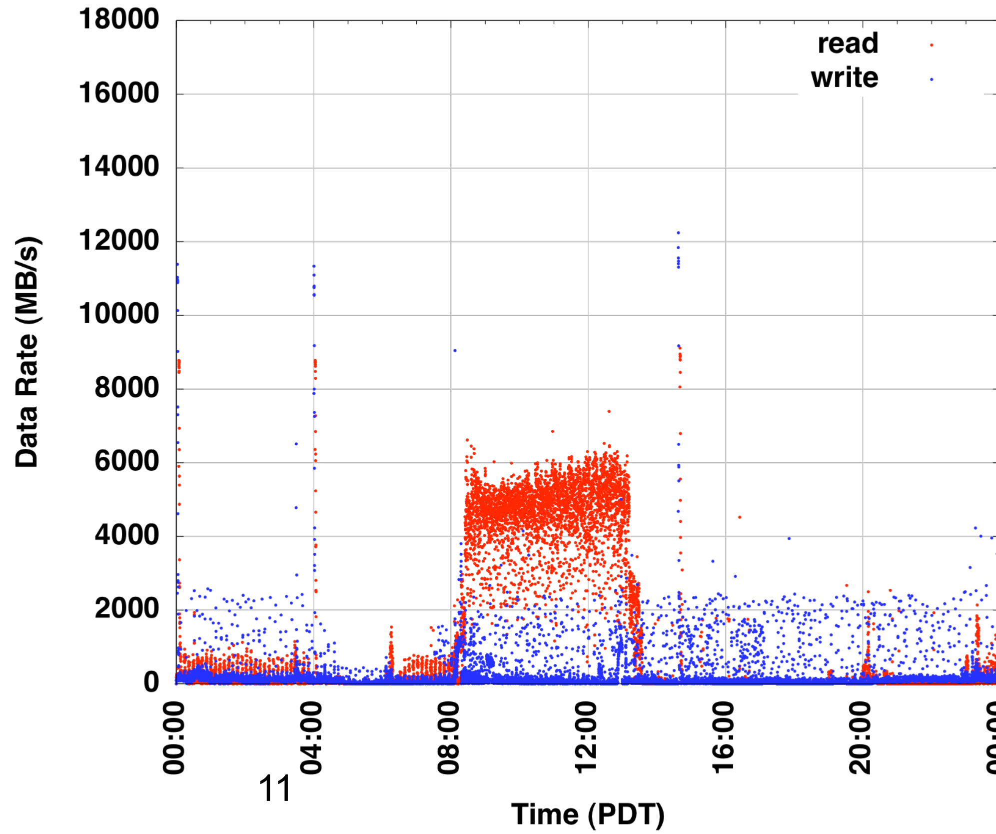


LMT captures bytes moved for each server every 5 seconds

LMT provides detailed I/O performance data

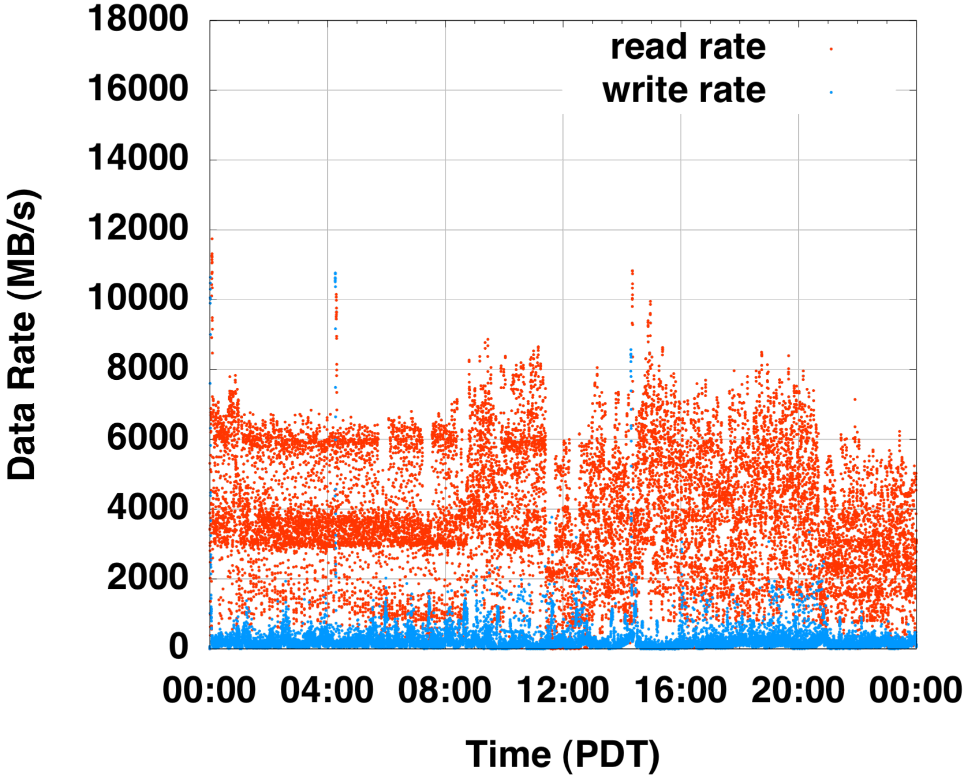
- Captures all read and write I/O to scratch file systems
- Samples every five seconds
- Sever-side data is anonymous in that it doesn't record which application originated it.

May 22, 2010
Example day of I/O on the */scratch* file system

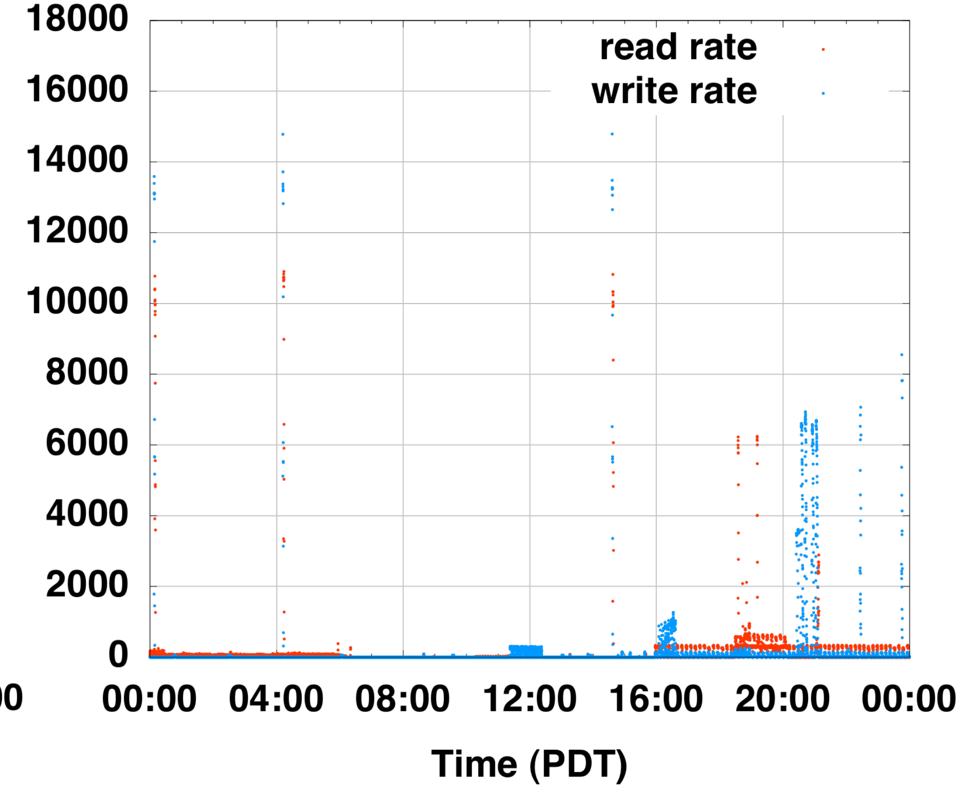


24 hours of LMT data

Viewing time series data doesn't scale.



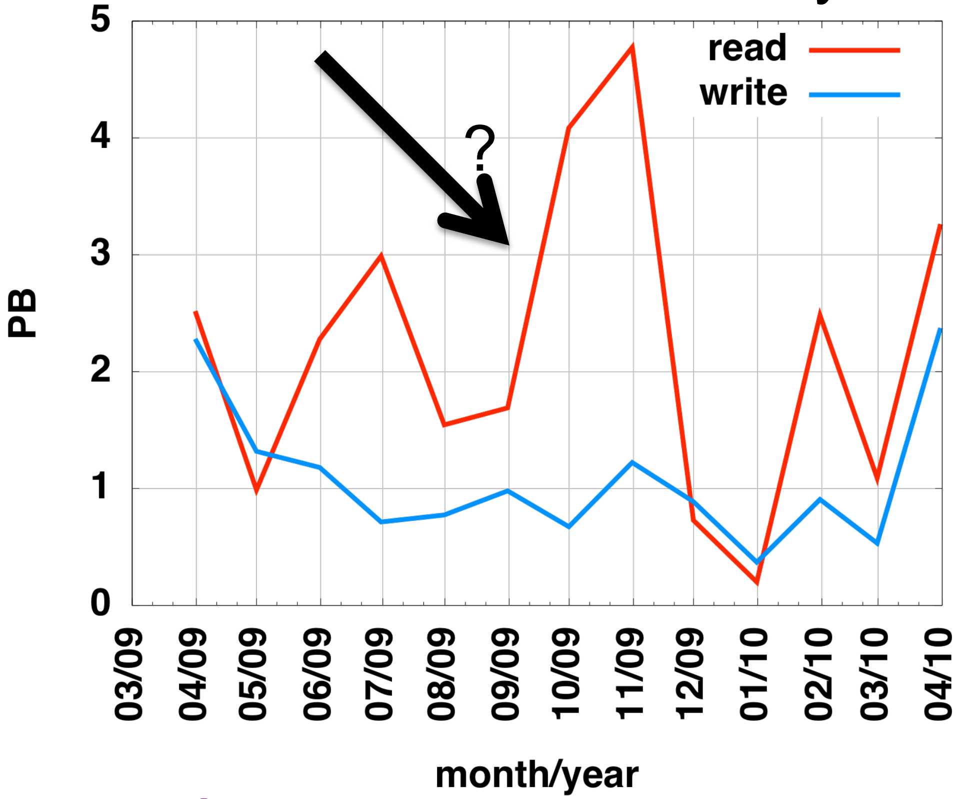
/scratch



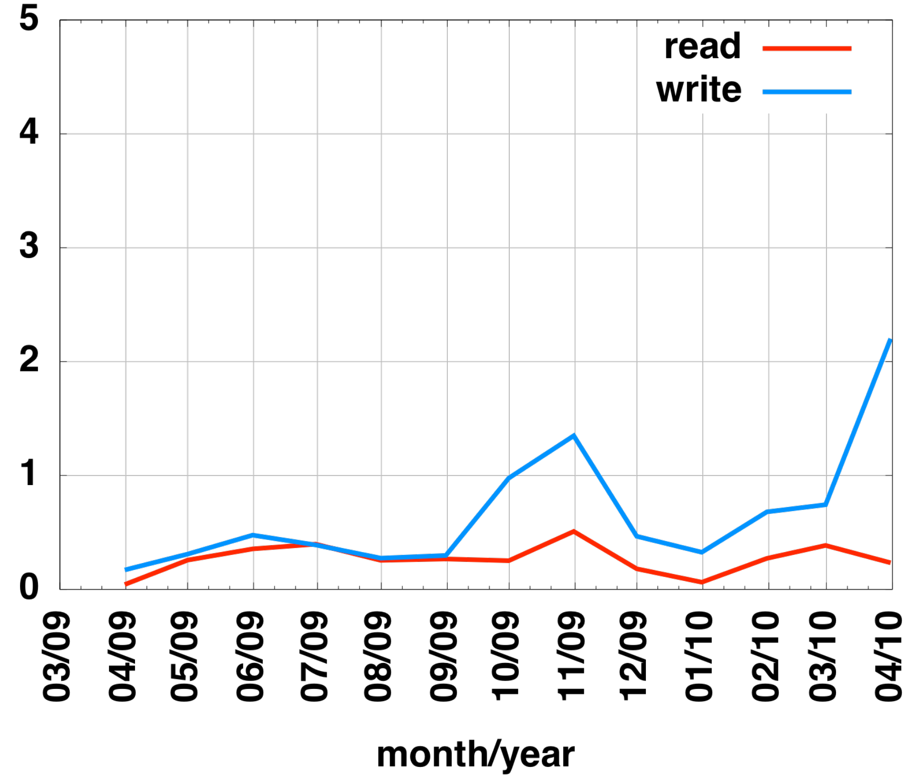
/scratch2

"Big I/O" was directed to */scratch2*

One year of LMT data



/scratch - read workload



/scratch2 - write workload

"Big I/O" == I/O intensive,
But most I/O is still on */scratch*

IOR test probe

IOR is a parallel I/O benchmark

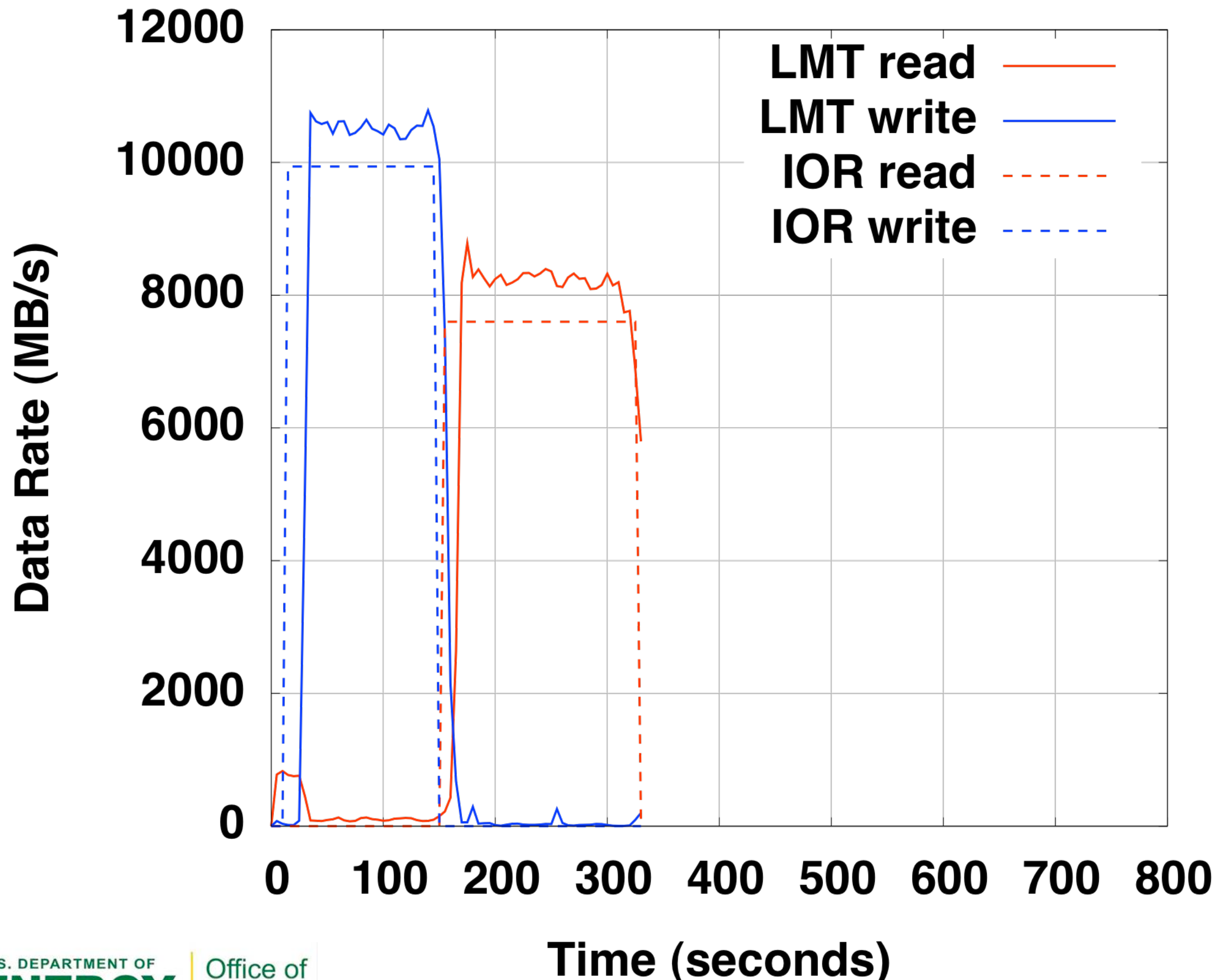
Reports:

- b_r = bytes read
- t_r = time for read I/O
- $r_r = b_r/t_r$ = read rate

- b_w = bytes written
- t_w = time for write I/O
- $r_w = b_w/t_w$ = write rate

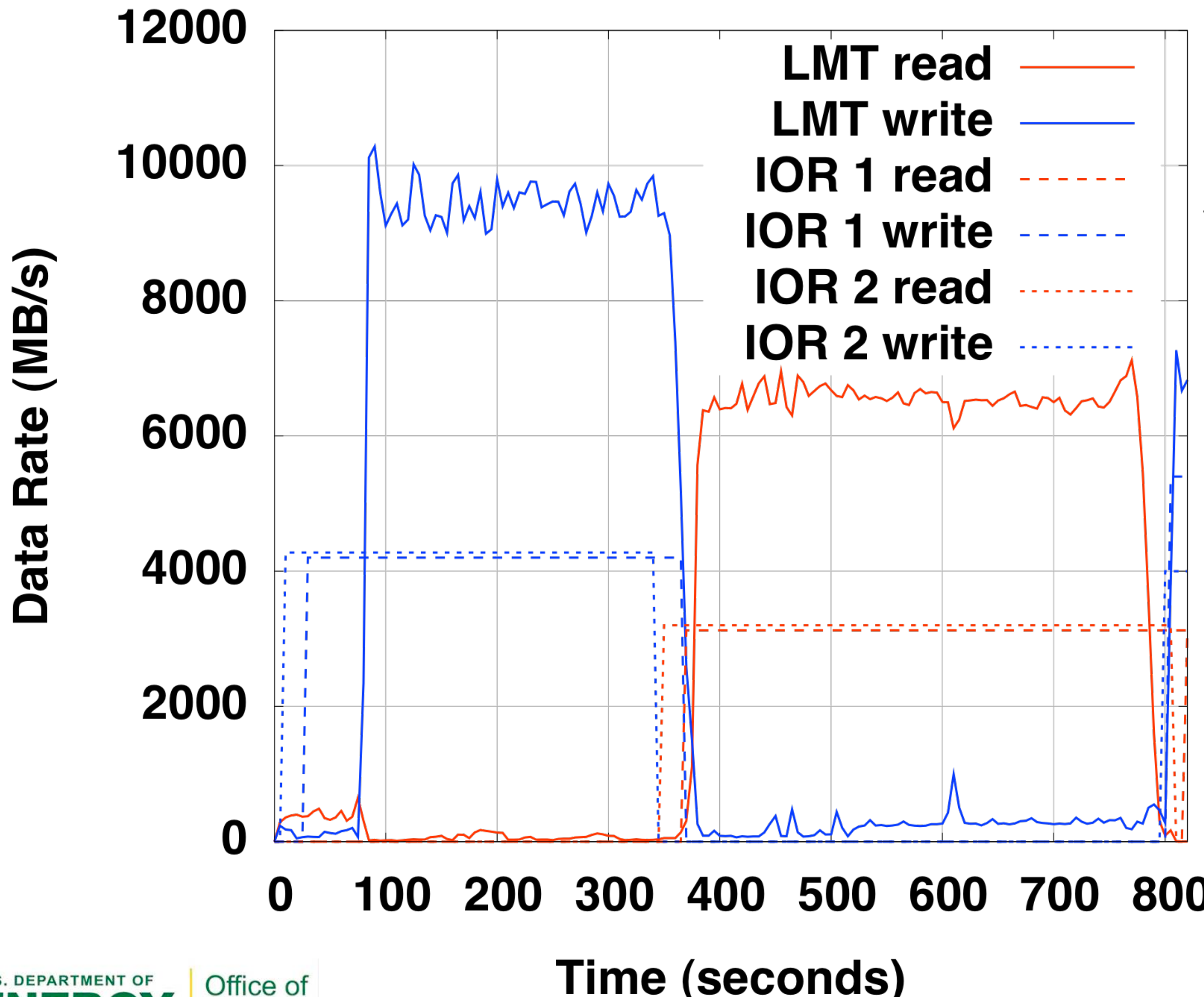
A regularly scheduled IOR test act as a standard probe of file system performance and gives an indication of the level of activity on the system

IOR test on dedicated system



Dedicated Testing Time Ensures a Quiet system

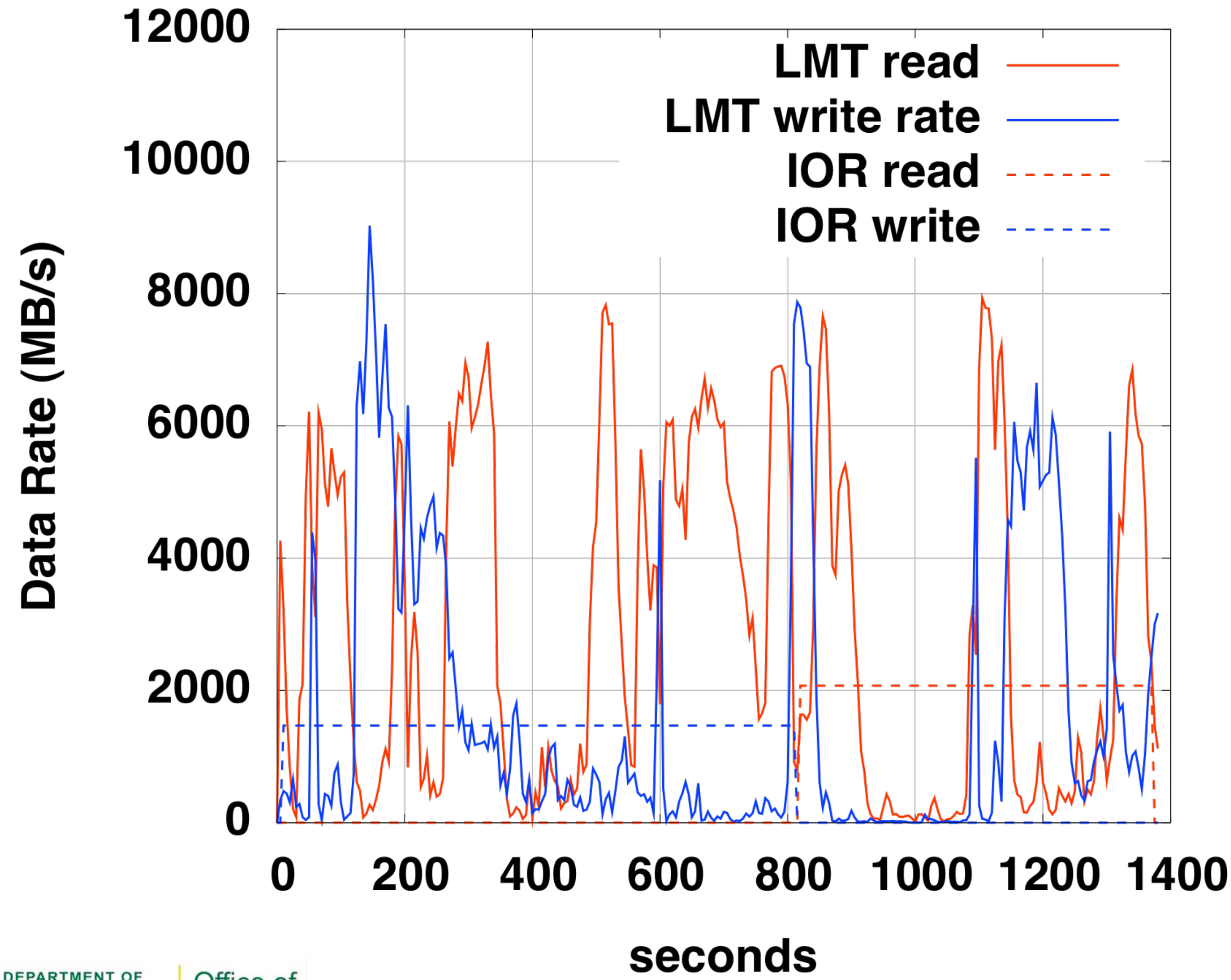
Two IOR tests



Run sequentially would take the same time, but occupy half as many CPUs.

Contention takes from compute!

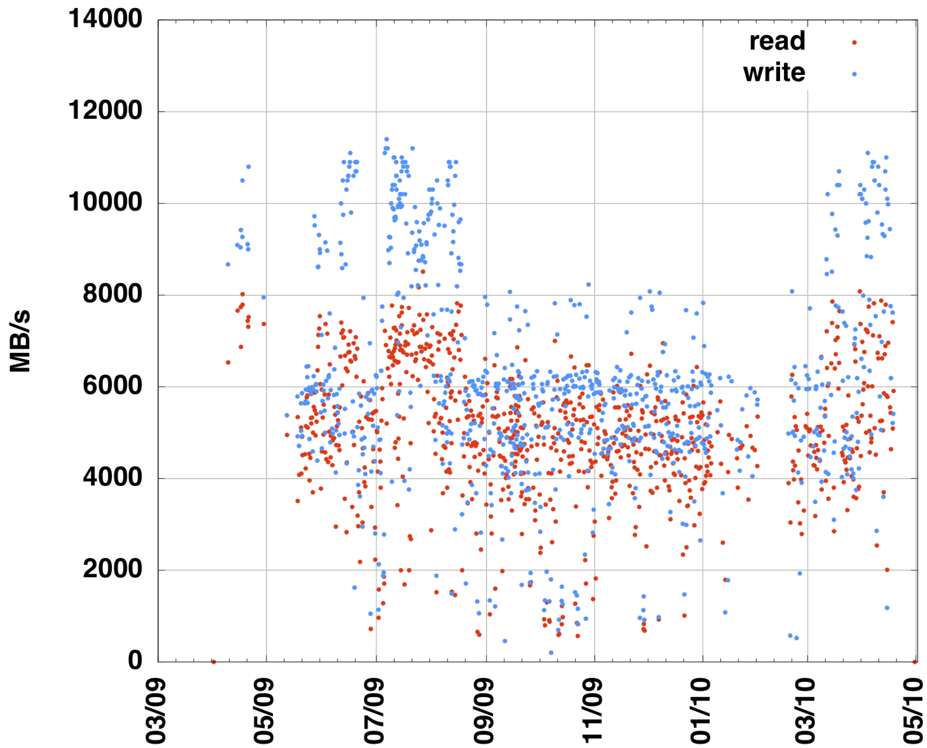
IOR test probe sees contention "in the wild"



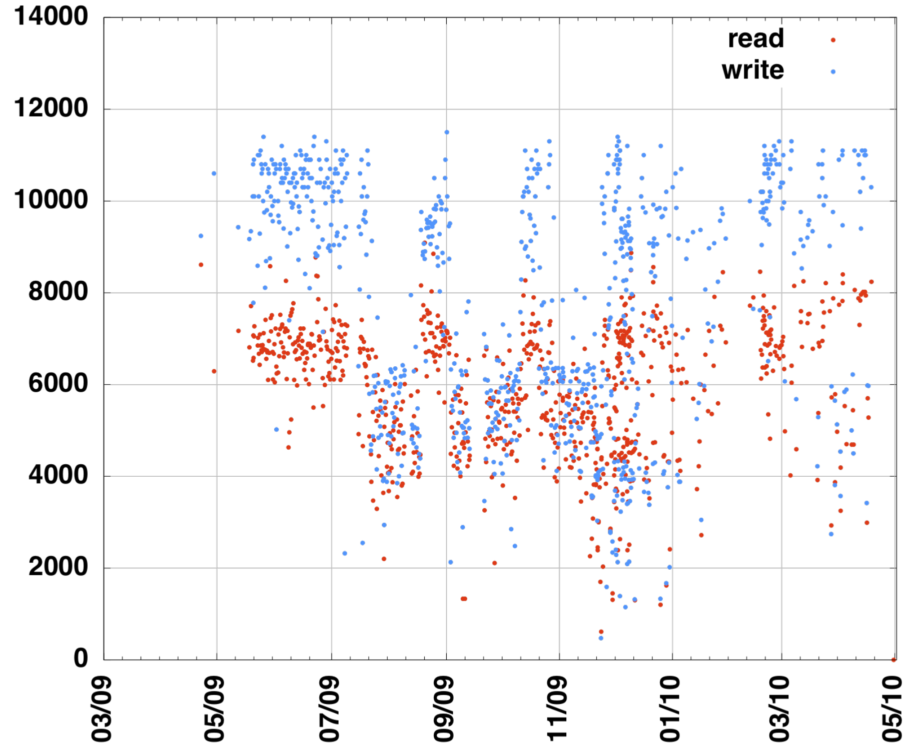
Production time testing shows the effect of contention

Test probe runs 3 times a day

IOR shows variability during a year of testing.



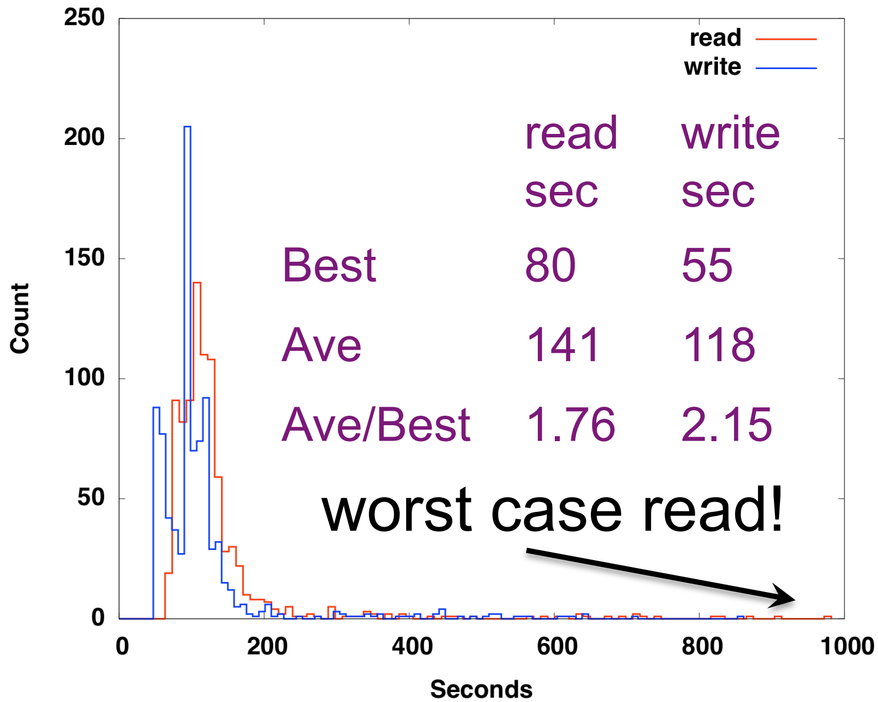
/scratch



/scratch2

Test probe shows more contention on */scratch*

Contention follows a long-tailed distribution



/scratch

/scratch2

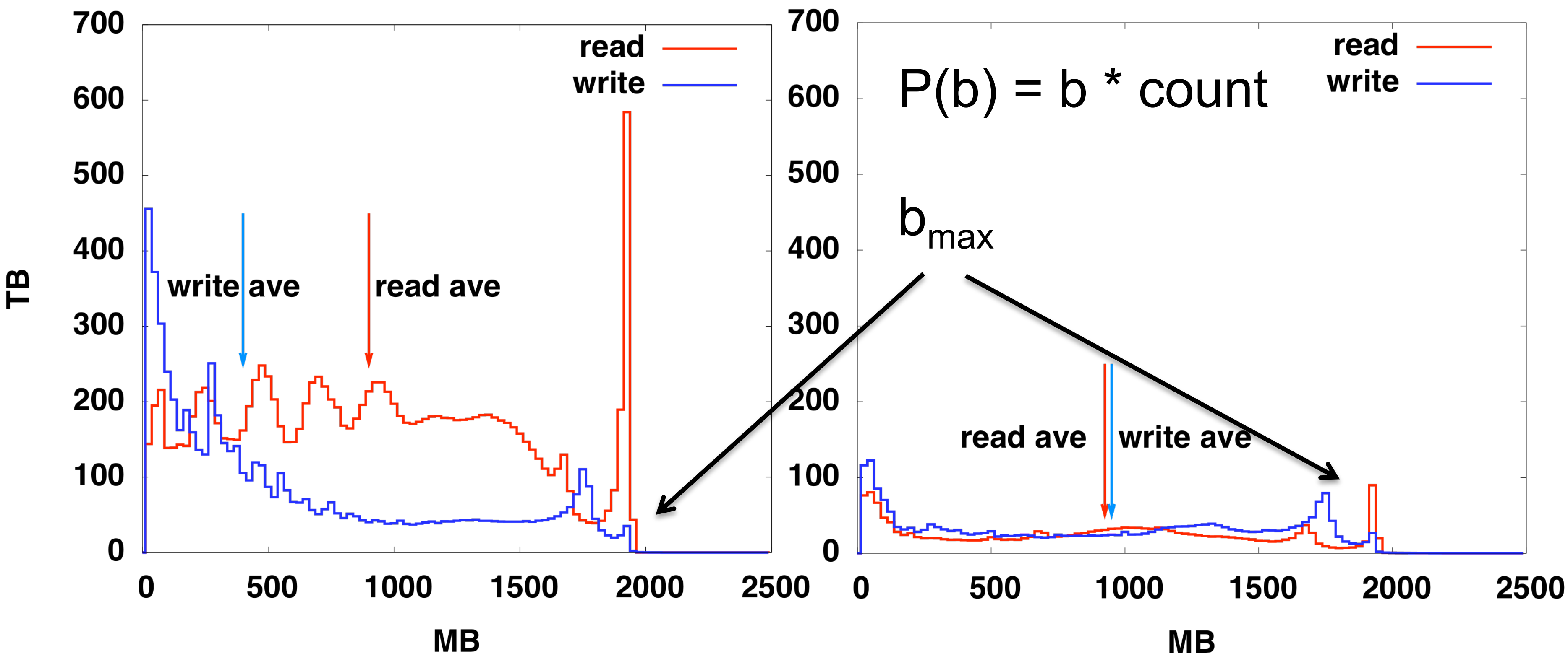
Analyzing I/O Statistics

- Power Spectrum – weighted histogram

- Auto-correlation – If you know the weather now will you know it in an hour?

Power Spectrum for a year of data

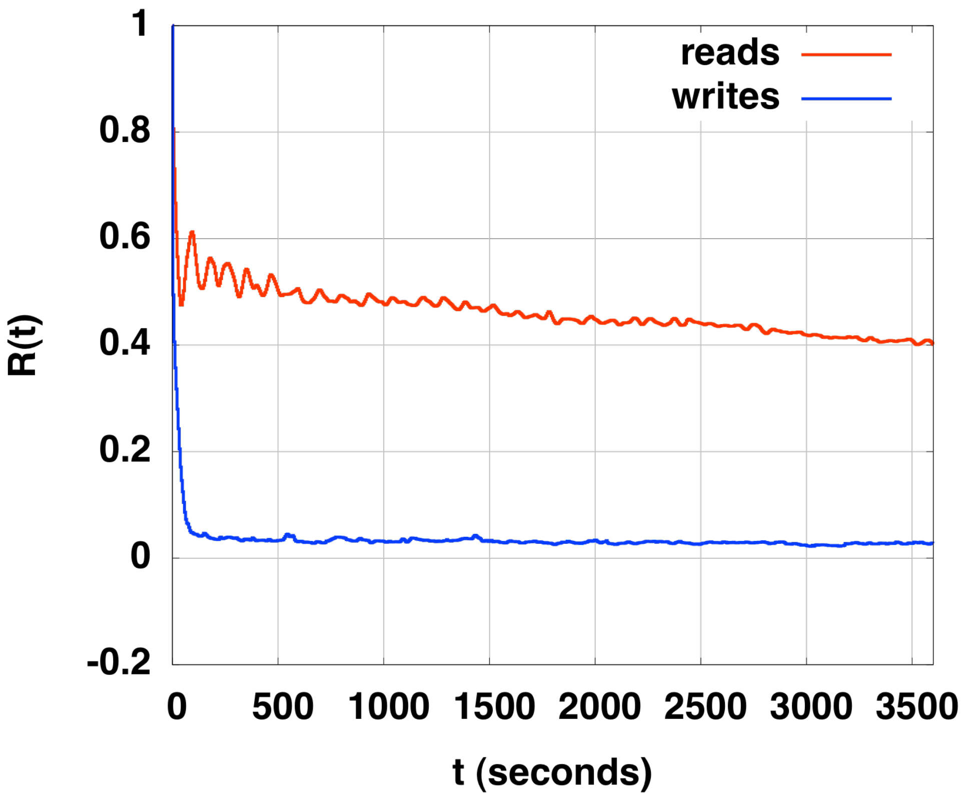
Samples every 5 seconds, max I/O rate about 400 MB/s (per server), so max transfer is about b_{max} = 2000 MB per sample



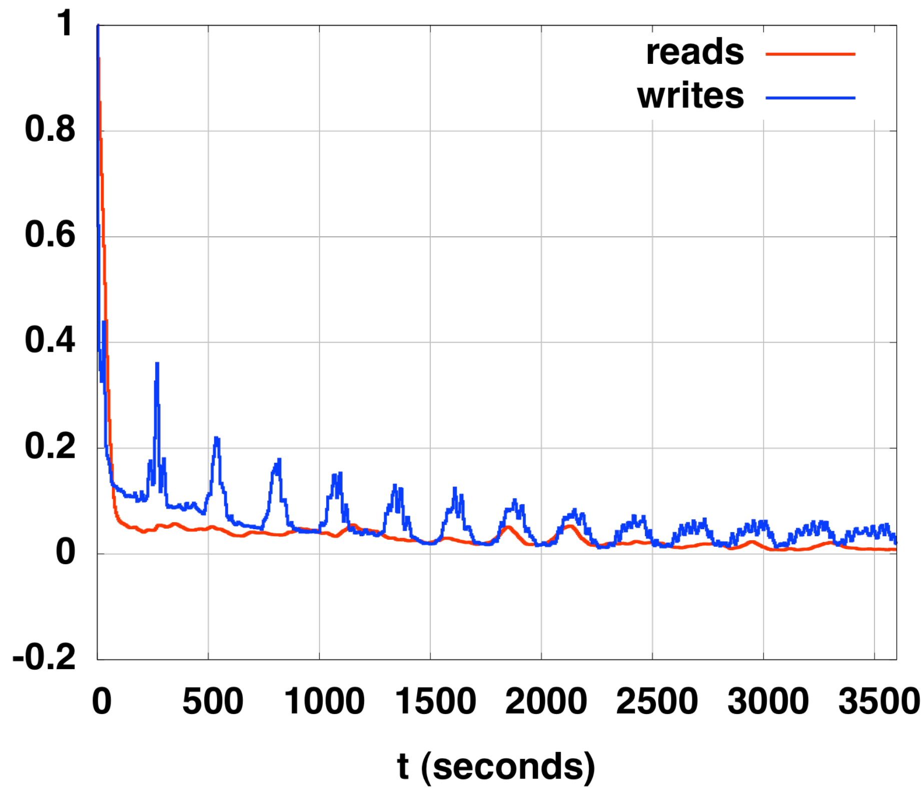
/scratch *scratch2*

The Auto-correlation function for April 2010

One month of data for lags out to one hour



/scratch

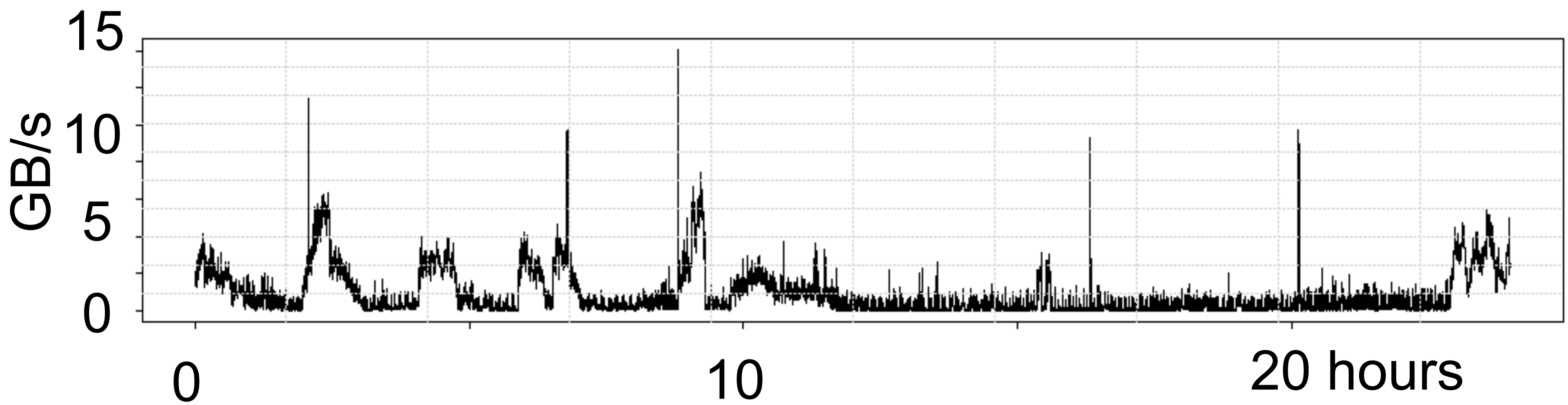


/scratch2

But is a two file system strategy the best policy?

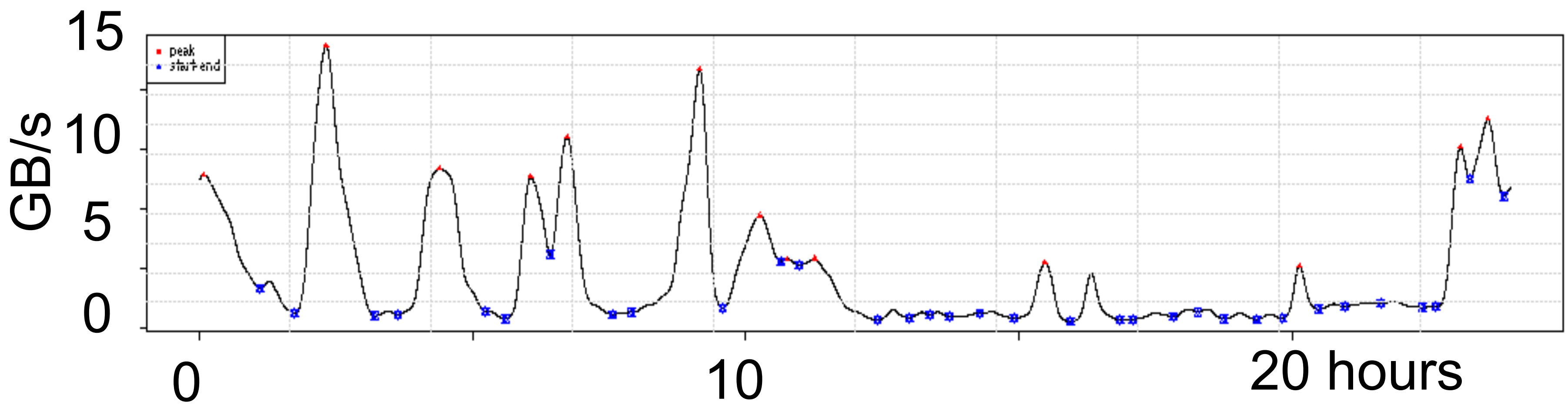
- Our metric has been user satisfaction
- An alternative metric is to count CPU hours lost to contention
- To measure that requires connecting specific I/O to specific jobs
    - Integrated Performance Monitoring (IPM) can do that
    - But it is not always available
- We propose to infer the connection between LMT data and the job log via spectral analysis

We can identify job I/O events



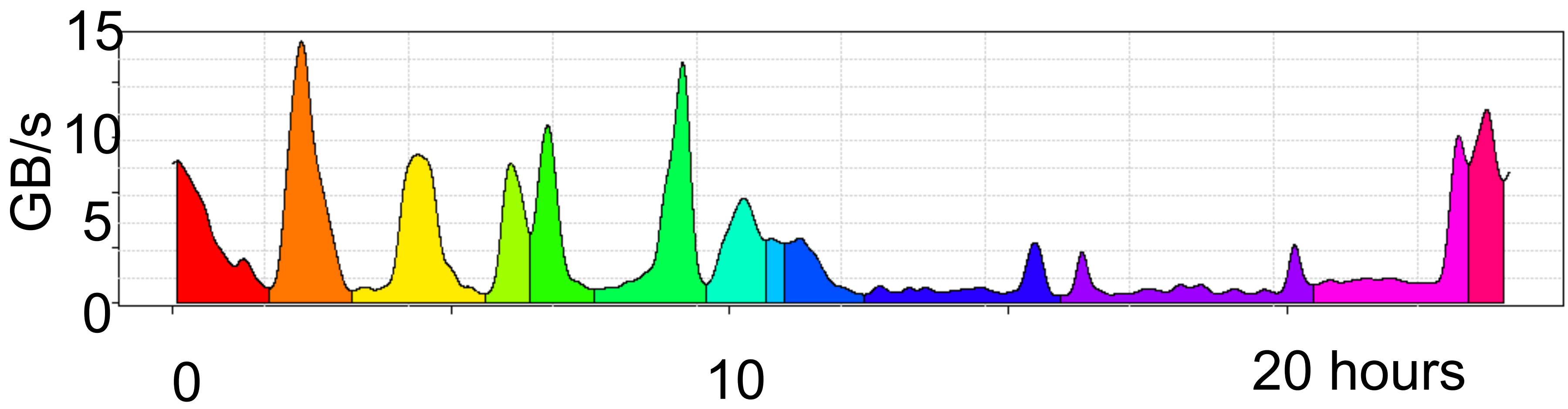
- 24 hours of LMT read data treated as a time series

We can identify job I/O events



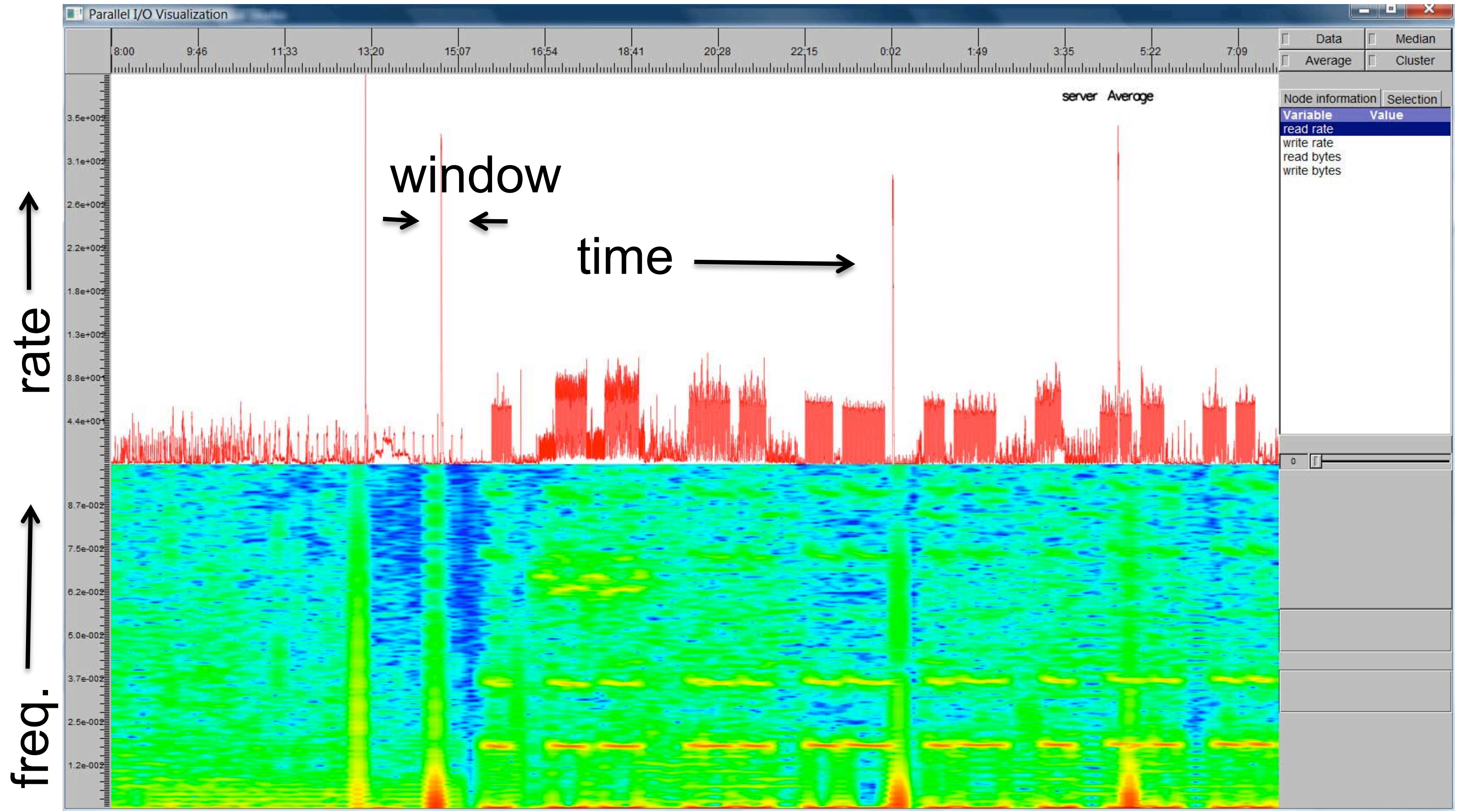
- 24 hours of LMT read data treated as a time series
- Smoothed with a Gaussian filter
- Pick out the peaks and the troughs

We can identify job I/O events



- 24 hours of LMT read data treated as a time series
- Smoothed with a Gaussian filter
- Pick out the peaks and the troughs
- These are candidate “events”
- Further analysis needed

Fourier Analysis



Short term Fourier transform (STFT)

A Metric for I/O System Utilization

- Identify the source (job) of I/O
  - We don't have to get it all
- Establish the occurrence distribution d
  - This will probably depend on job size n, d(n)
  - and on its duration t, d(n, t)
  - this is the I/O workload!
- Calculate the collision probability density $P_d(n, t)$
  - Wasted CPU = $\Sigma_{n,t}\, n \bullet t \bullet P_d(n, t)$

Conclusions

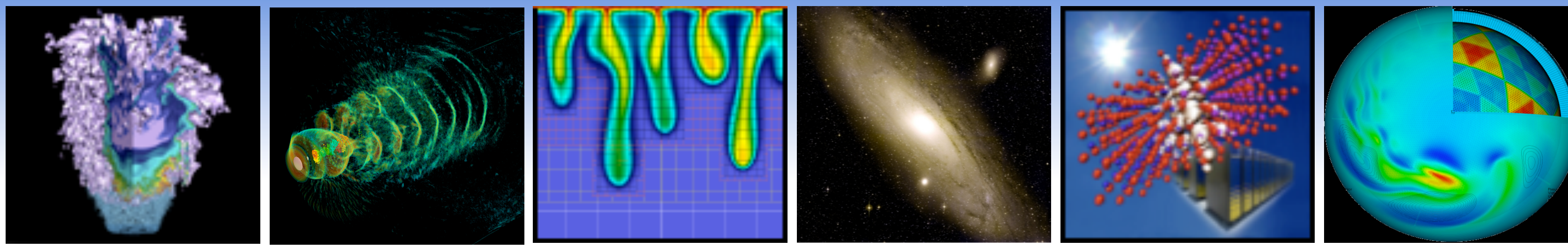
- Monitoring shows the workloads are different
  - LMT: Reads dominate on */scratch* and writes dominate on */scratch2.*
  - IOR: A test probe shows more contention on */scratch,*
  - Power Spectrum: The */scratch* workload represents smaller writes, and
  - Auto-correlation: */scratch* has high prevalence of reads.
- But is it the best we can do?
- Future work - Spectral analysis of the data may help:
  - Identify the details of the workload
  - Tell us how a given workload would perform on other I/O configurations

Acknowledgements

Office of Advanced Scientific Computing Research in the Department of Energy's Office of Science under contract number DE-AC02-05CH11231, and

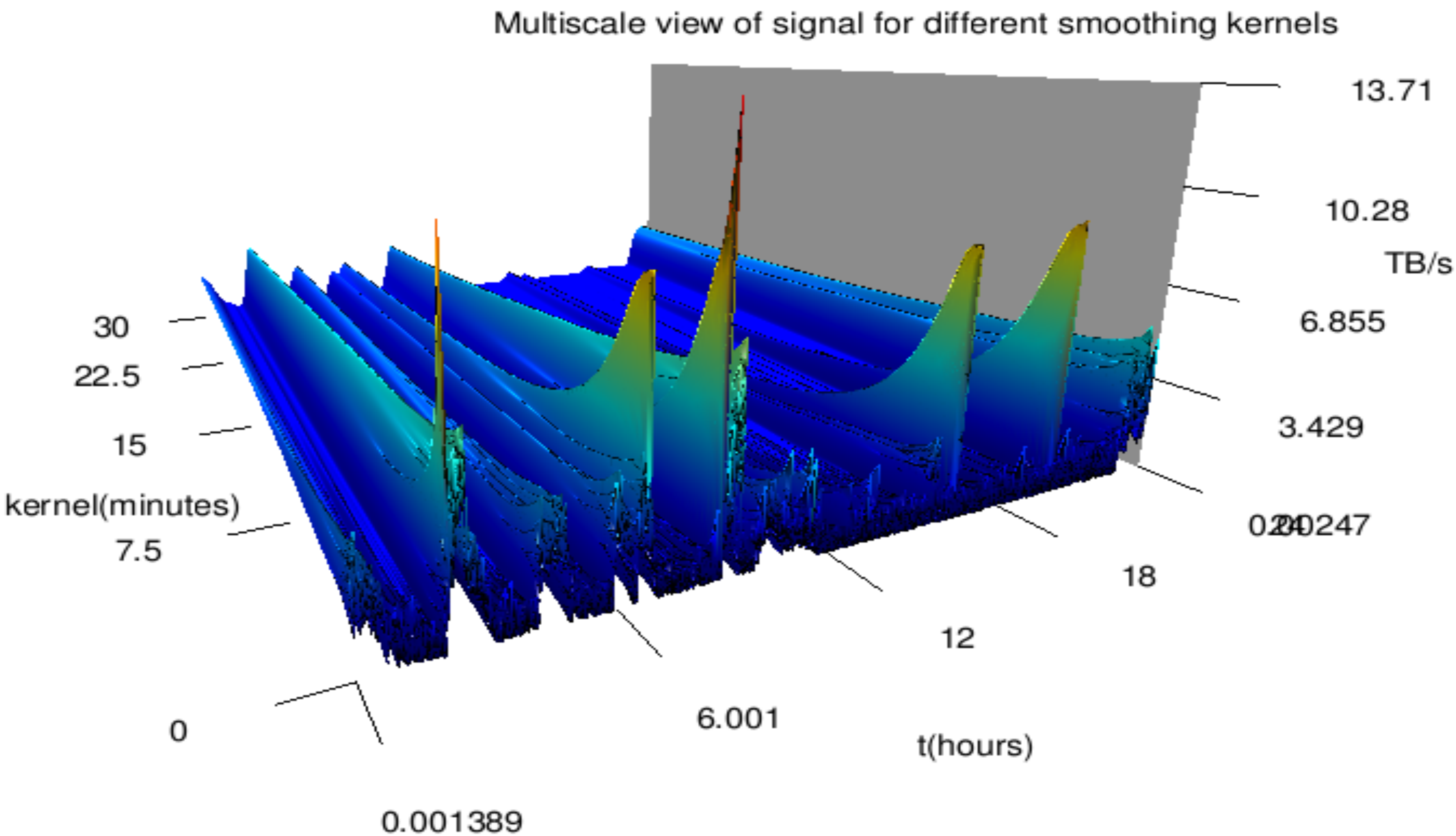
SciDAC Agreement No. DE-FC02-06ER25777, and

U.S. National Science Foundation through grants CCF 0938114

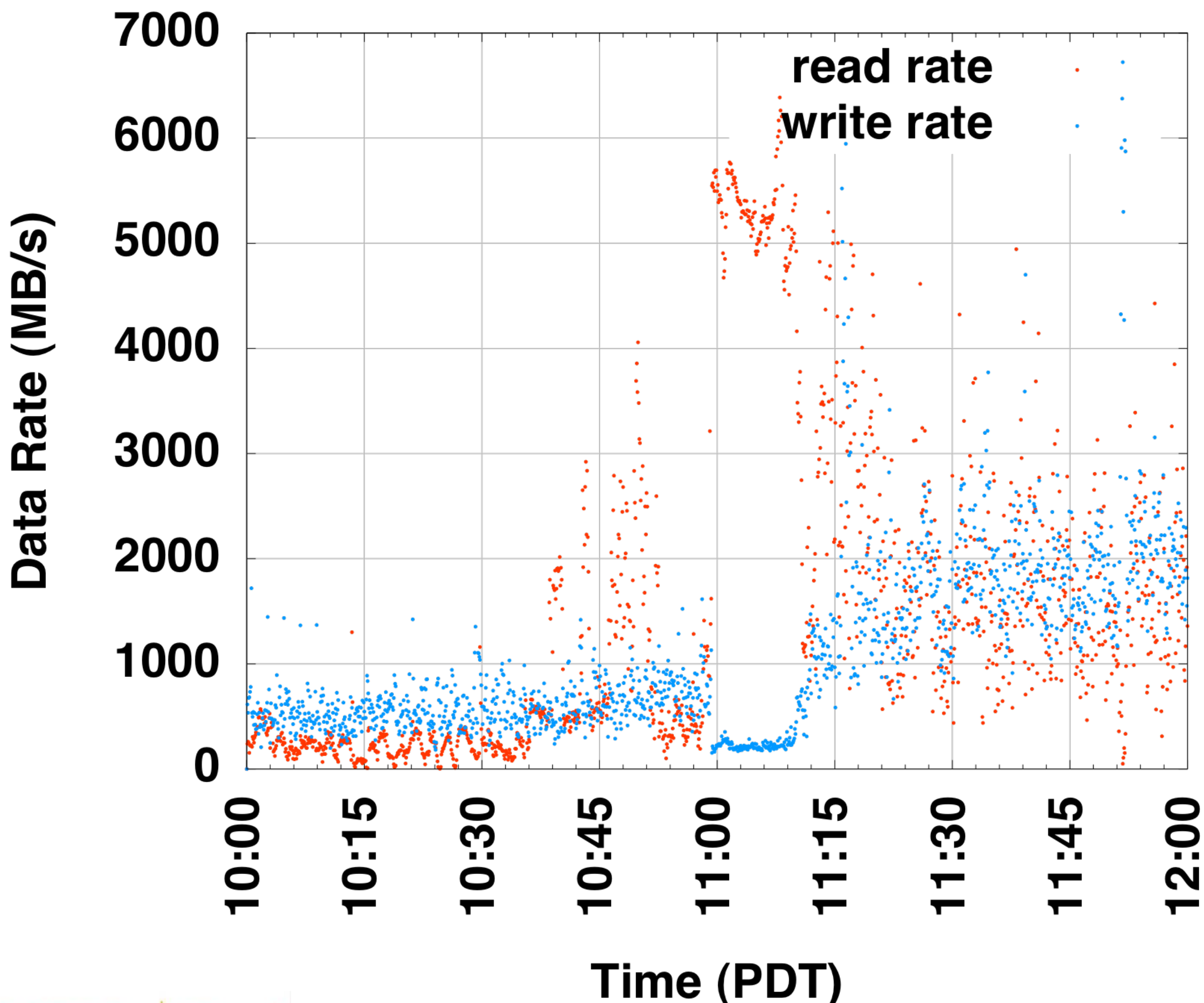


Thank you!

Varying the smoothing function

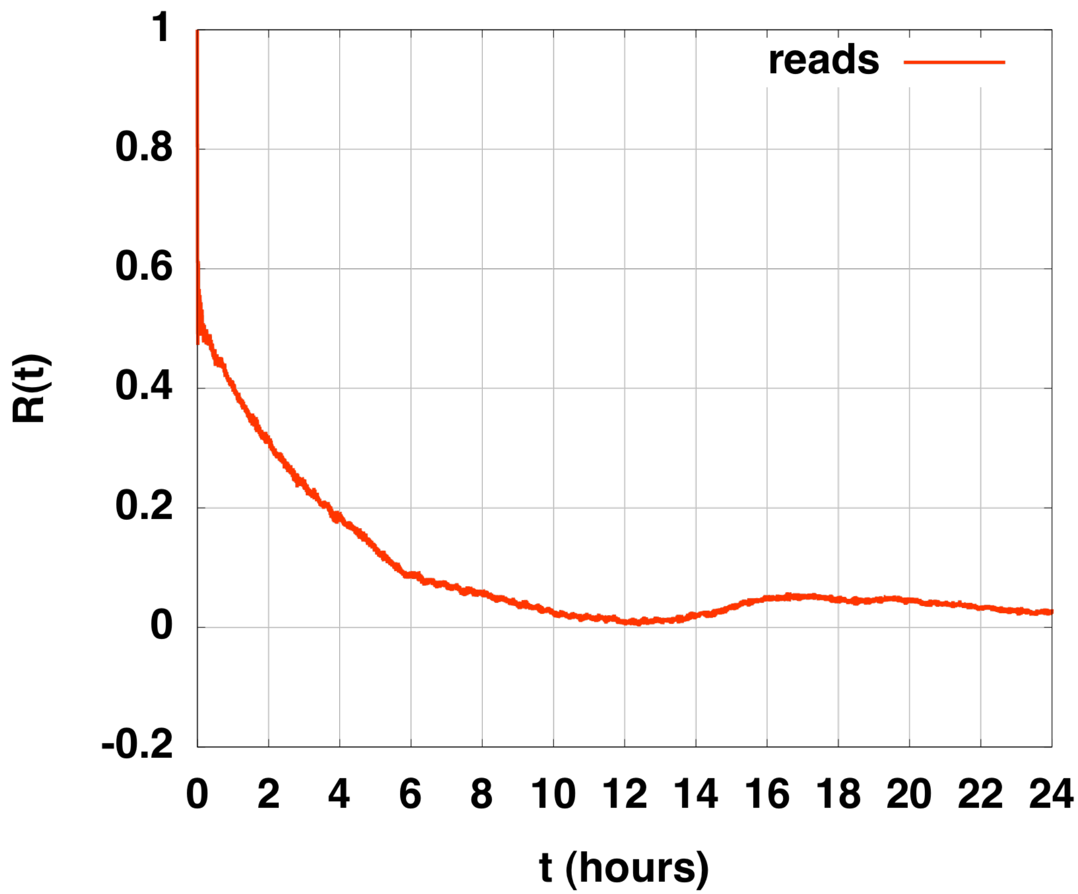


Contention happens all the time



Two hours of data show a common contention pattern

Auto-correlation with lags out to 24 hours



/scratch