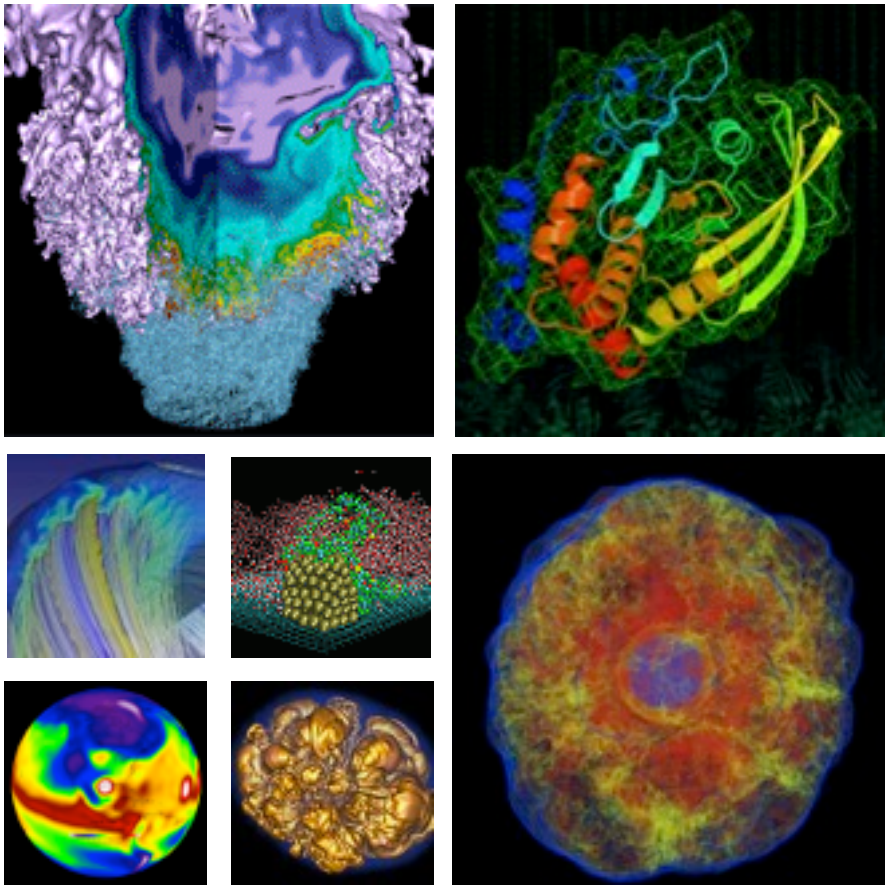
Balanced Design for HPC I/O Capability



Andrew Uselton

April 12, 2013

Overview

I have a budget for an HPC system. How much of it should go to the I/O infrastructure?

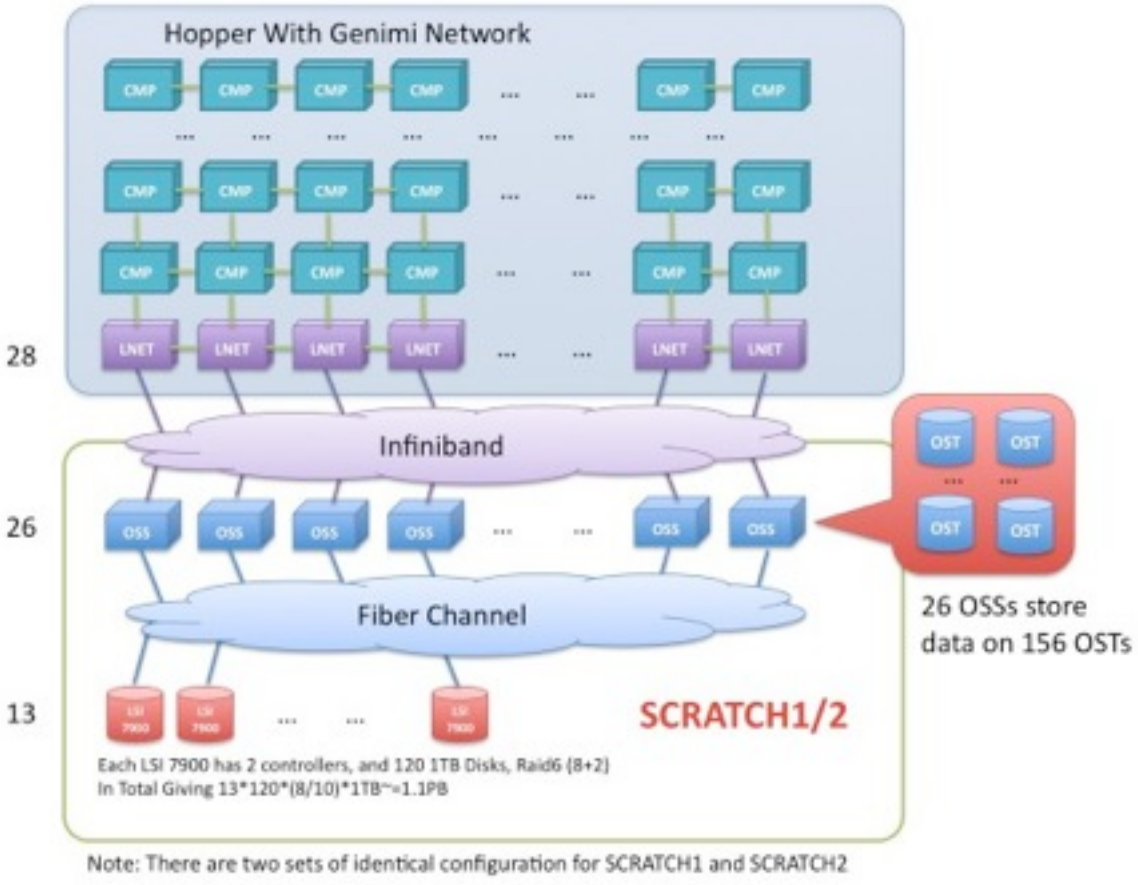
- Determining the workload
- A cost model for HPC systems
- Calibrating the model
- Projecting to new architectures

The Hopper HPC System

- Hopper is a Cray XE6 deployed at NERSC since 2010
- It has 153,216 cores and 212 TB of memory

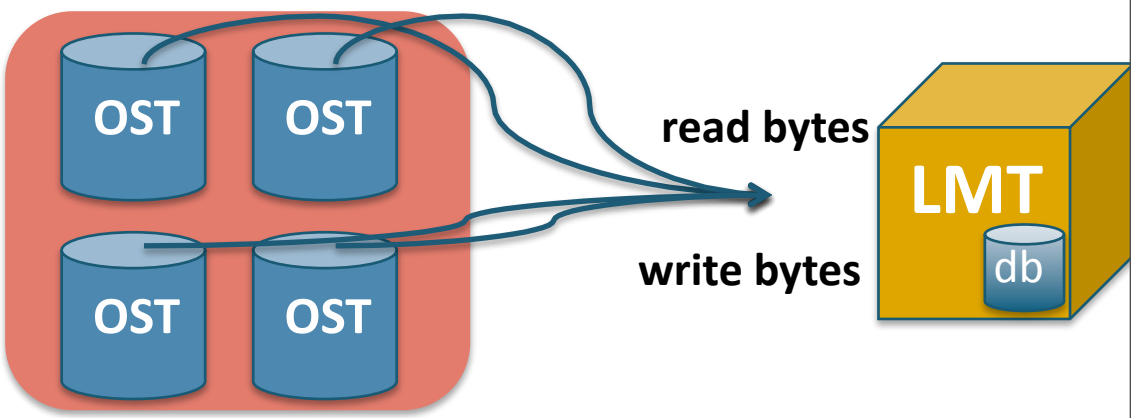
The "Scratch" Resource

- Hopper has 2 PB of disk arranged as two "scratch" file systems
- Each has 156 RAID arrays served by 26 Object Storage Servers (OSSs)



The Lustre Monitoring Tool

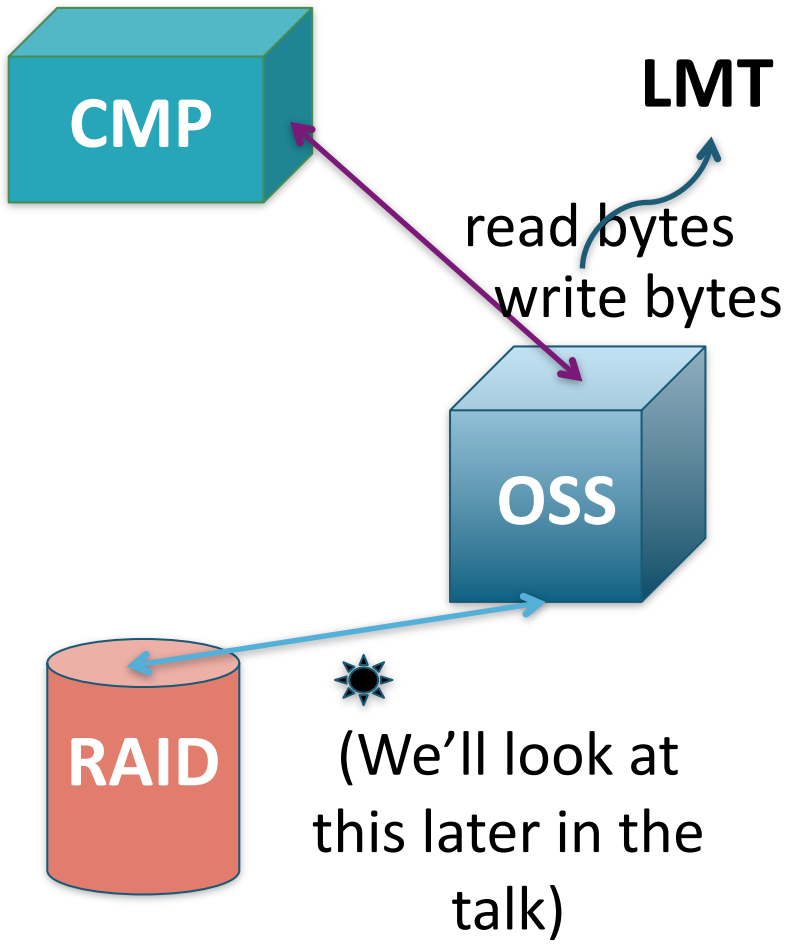
- Each RAID array acts as an Object Storage Target (OST)
- The Lustre Monitoring Tool (LMT) gathers data every five seconds



- 'Read bytes' ('write bytes') for the file system is the sum of the 156 values on the OSTs

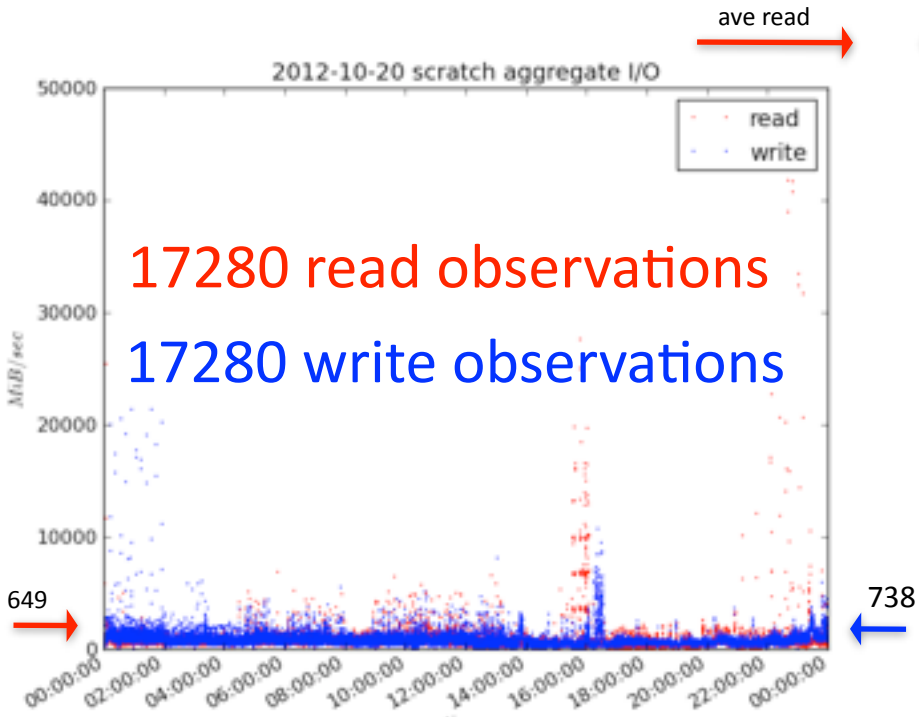
LMT Provides the Raw Data

- "Read bytes" and "write bytes" gathered by LMT represents data movement between the compute nodes (Lustre clients) and the servers (OSSs).
- There is also data movement between the servers and the RAID arrays.

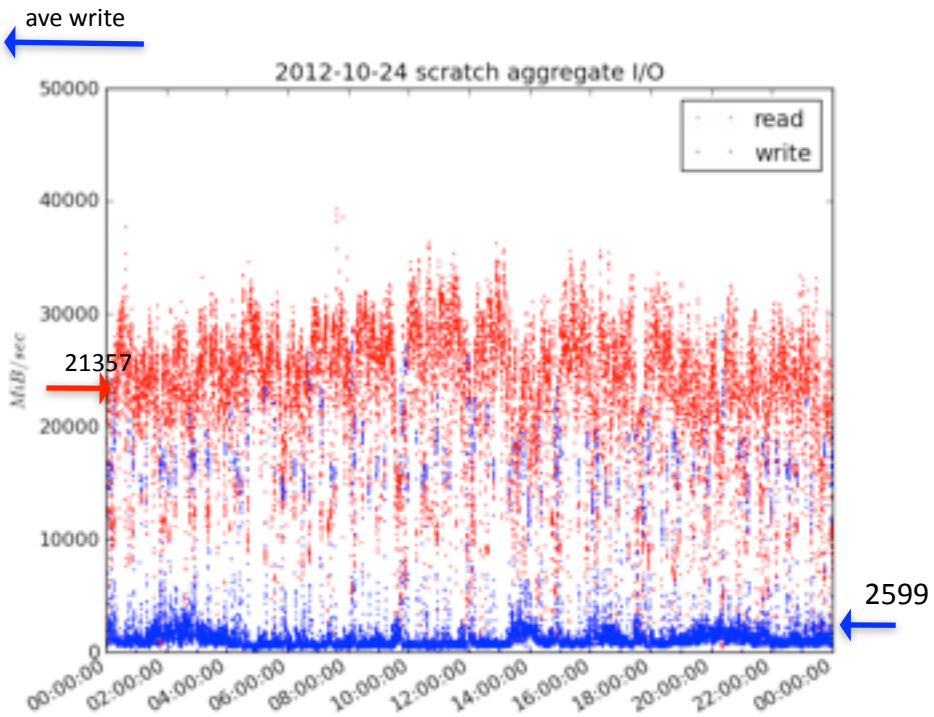


Read and Write Behavior

- A typical day



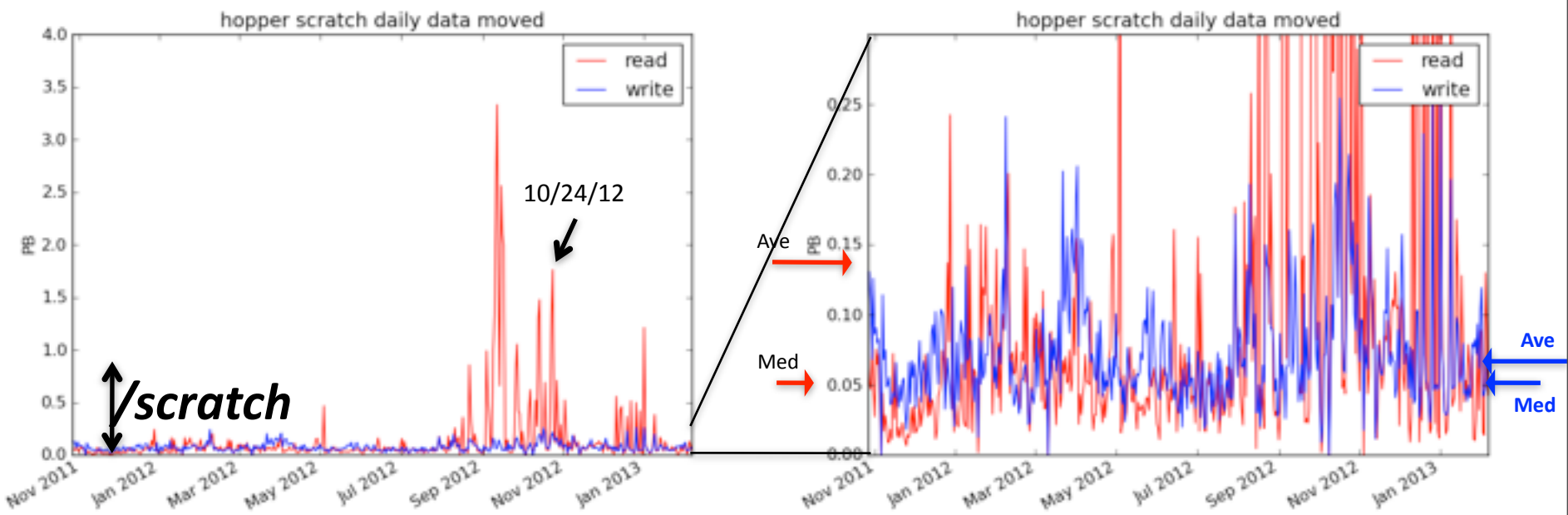
- **An extreme day**



- **This is the raw data showing how fast the file system is moving bytes in and out of scratch space.**

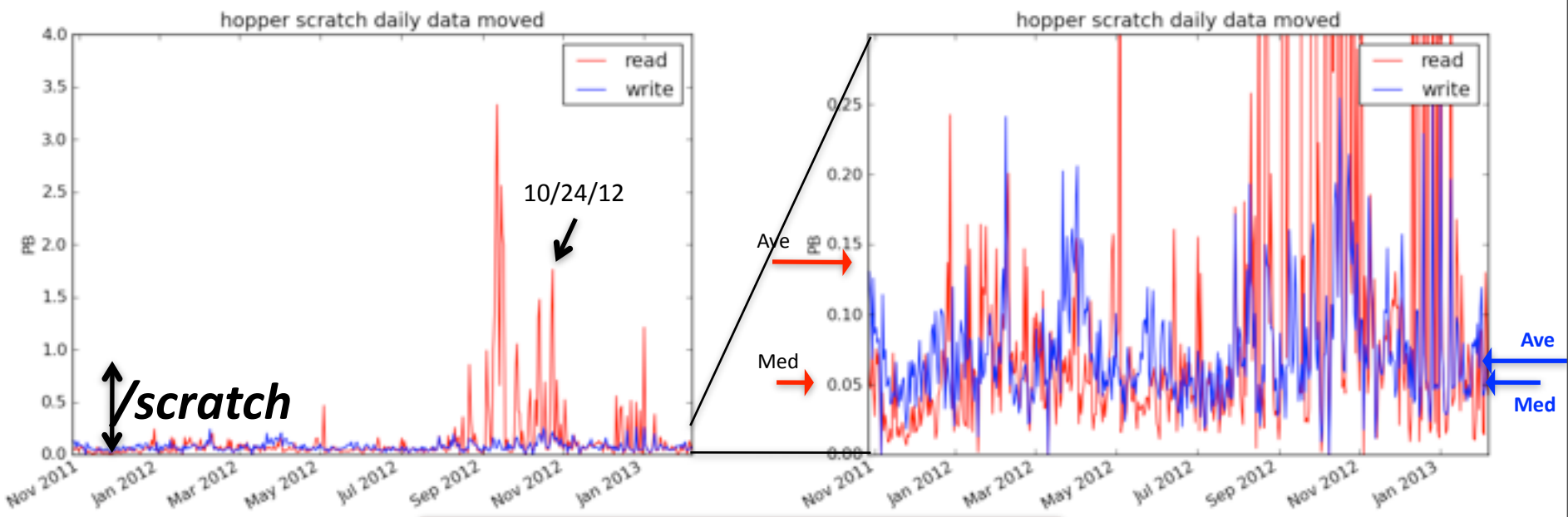
Data Rate as a Proxy for Load

- 488 days of observations on the amount of I/O moved
- The */scratch* file system is 1 PB
- The median conveys "typical" better than the average



Data Rate as a Proxy for Load

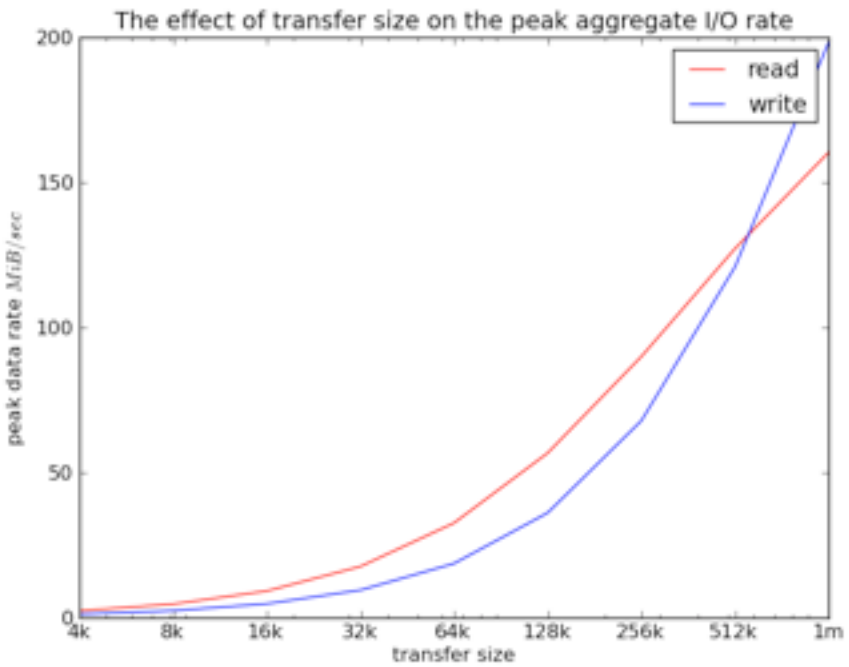
- 488 days of observations on the amount of I/O moved
- The */scratch* file system is 1 PB
- The median conveys "typical" better than the average



Is this the whole story?

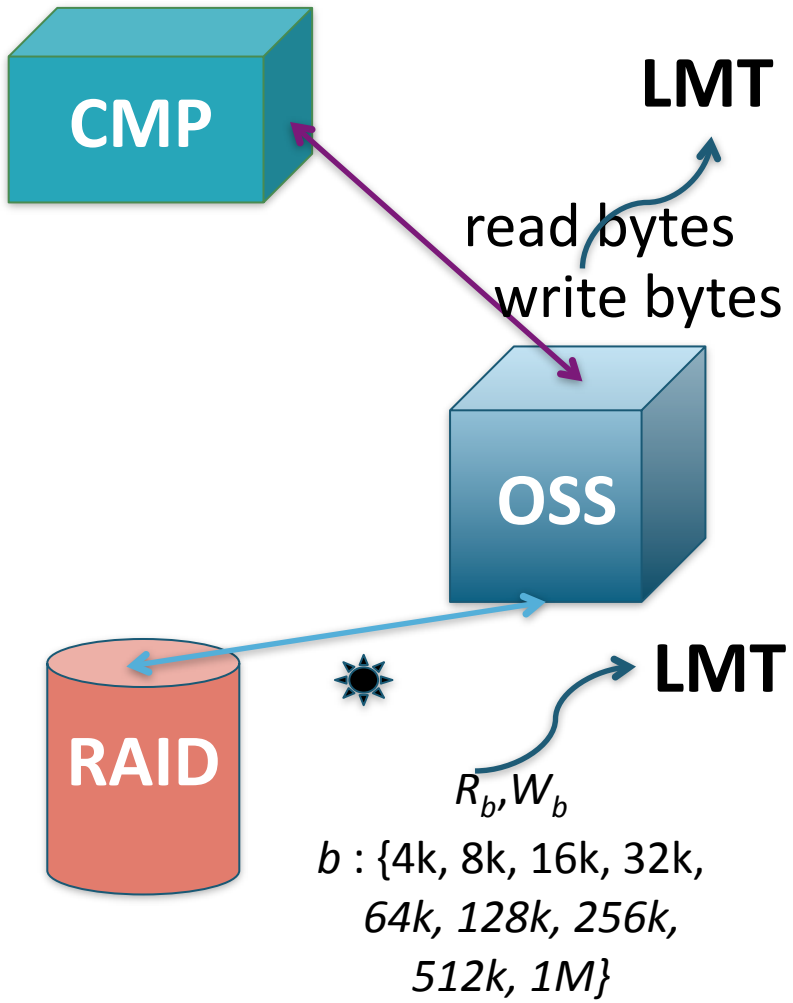
Small I/Os Degrade Performance

- A low data rate could be due to a high load with small I/Os
- We can measure the effect in controlled experiments
- We can gather info on I/O sizes from /proc



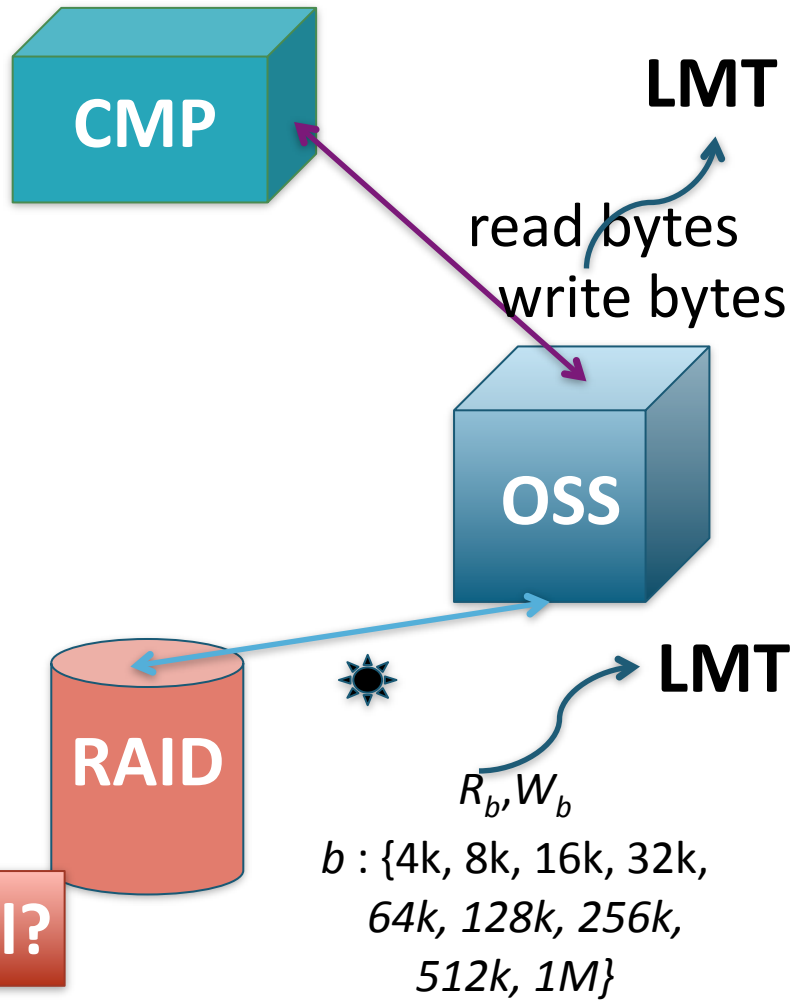
The LMT Extension for Disk I/Os

- In addition to “read bytes” and “write bytes”, LMT now captures a histogram every five seconds for the number of I/Os to and from disk
- The counts are binned in nine sizes by powers of two from 4kiB up to 1MiB



The LMT Extension for Disk I/Os

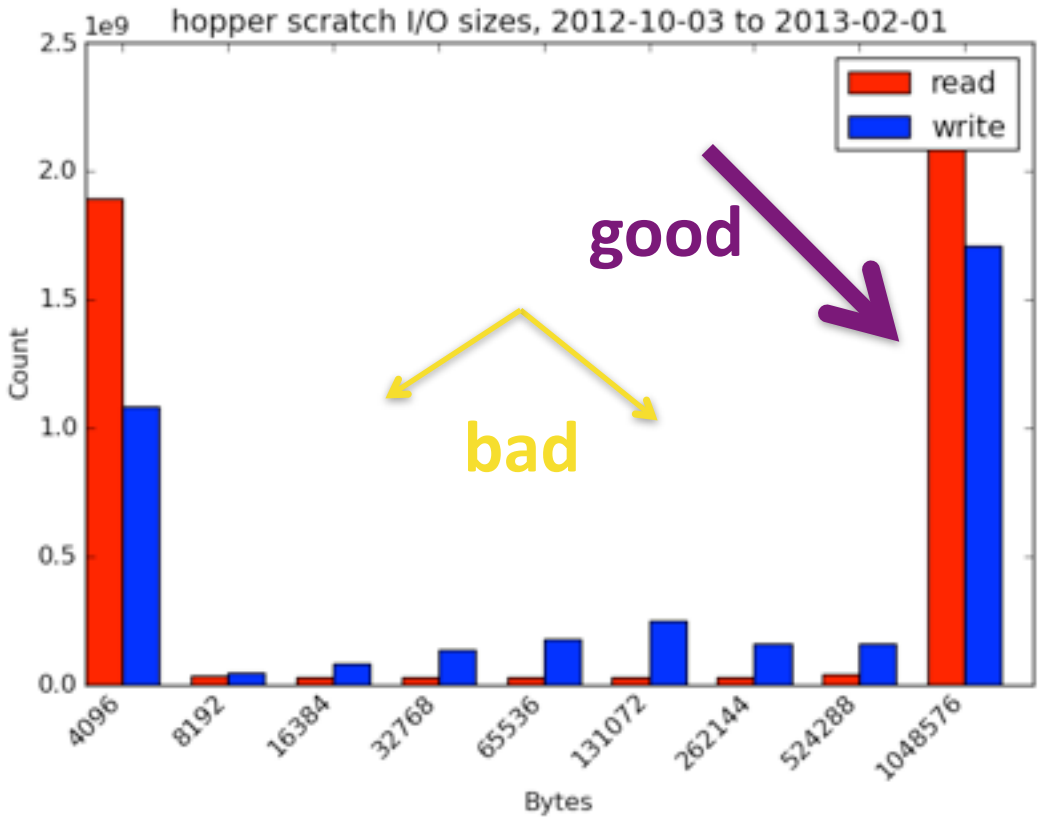
- In addition to "read bytes" and "write bytes", LMT now captures a histogram every five seconds for the number of I/Os to and from disk
- The counts are binned in nine sizes by powers of two from 4kiB up to 1MiB



But how much of our I/O is small?

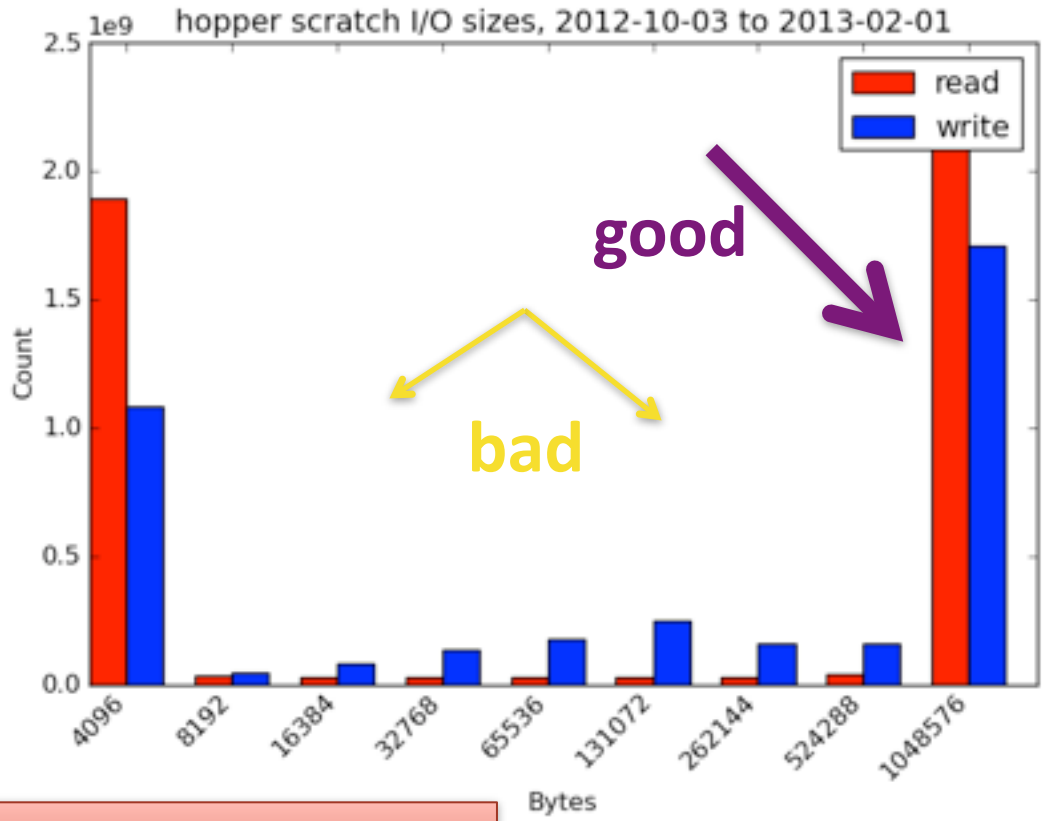
The Influence of small I/Os

- We have about 120 days of data from the LMT extension
- For writes, and especially for reads, the largest and smallest size I/Os dominate



The Influence of small I/Os

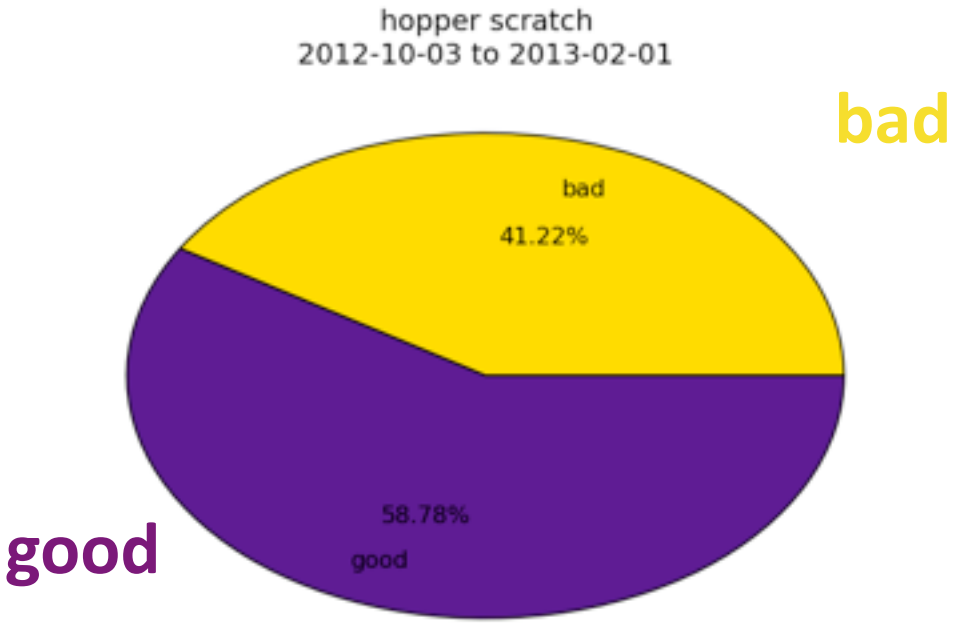
- We have about 120 days of data from the LMT extension
- For writes, and especially for reads, the largest and smallest size I/Os dominate



Anything below the optimum is bad!

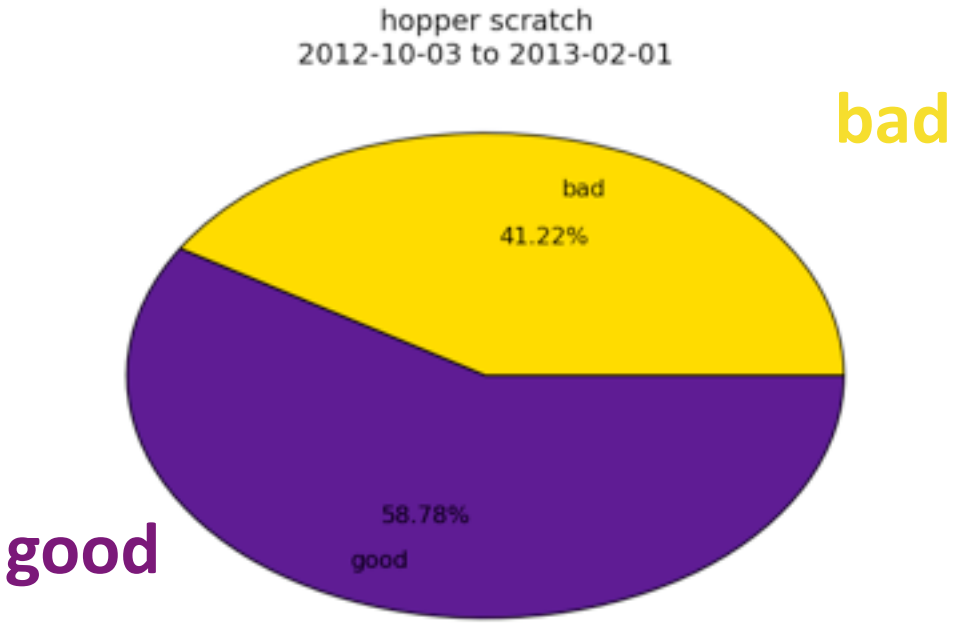
Good I/O Versus Bad I/O

- The proportion of I/O transactions that are optimum (good!) and sub-optimum (bad!)



Good I/O Versus Bad I/O

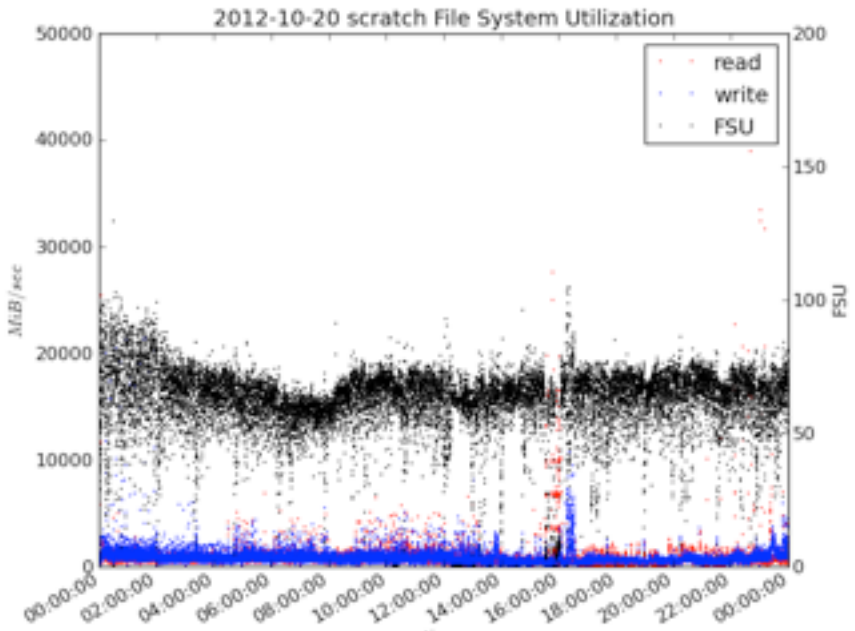
- The proportion of I/O transactions that are optimum (good!) and sub-optimum (bad!)

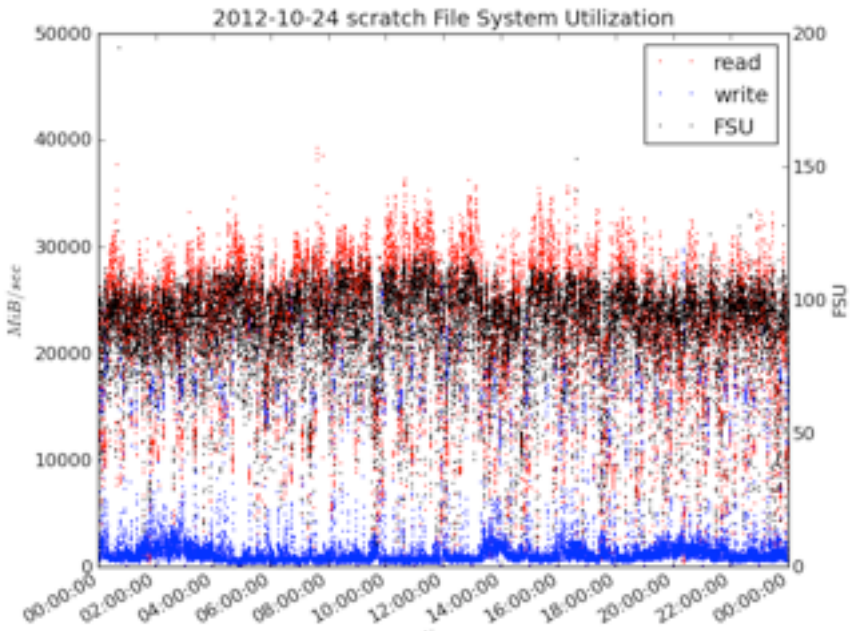


We can use this insight and the I/O size data to create a new metric for file system utilization (FSU)

Now look at those same two days…

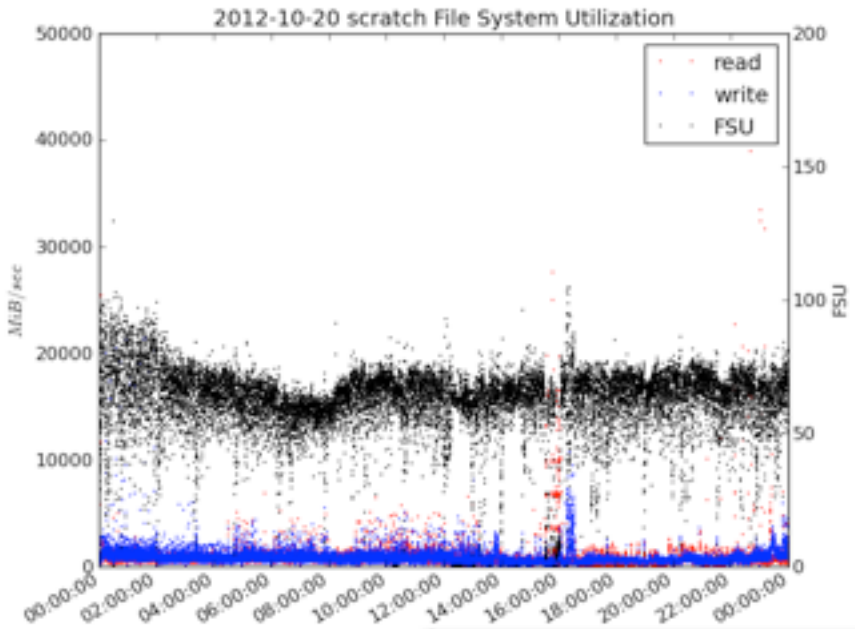
- The “busy” day on the right shows an FSU near 100
- The “typical” day that we thought was quiet has an FSU around 60 or 70. It was actually quite busy!

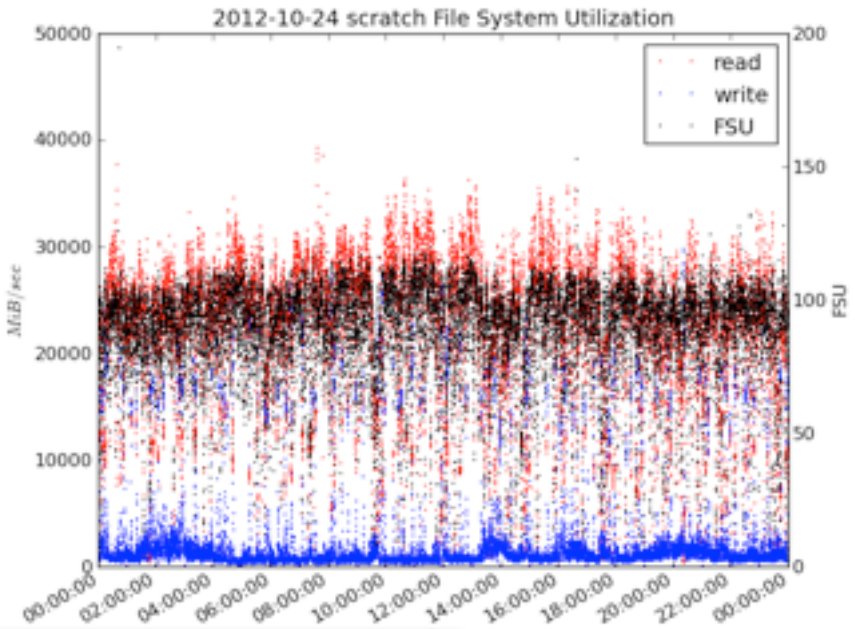




Now look at those same two days…

- The “busy” day on the right shows an FSU near 100
- The “typical” day that we thought was quiet has an FSU around 60 or 70. It was actually quite busy!

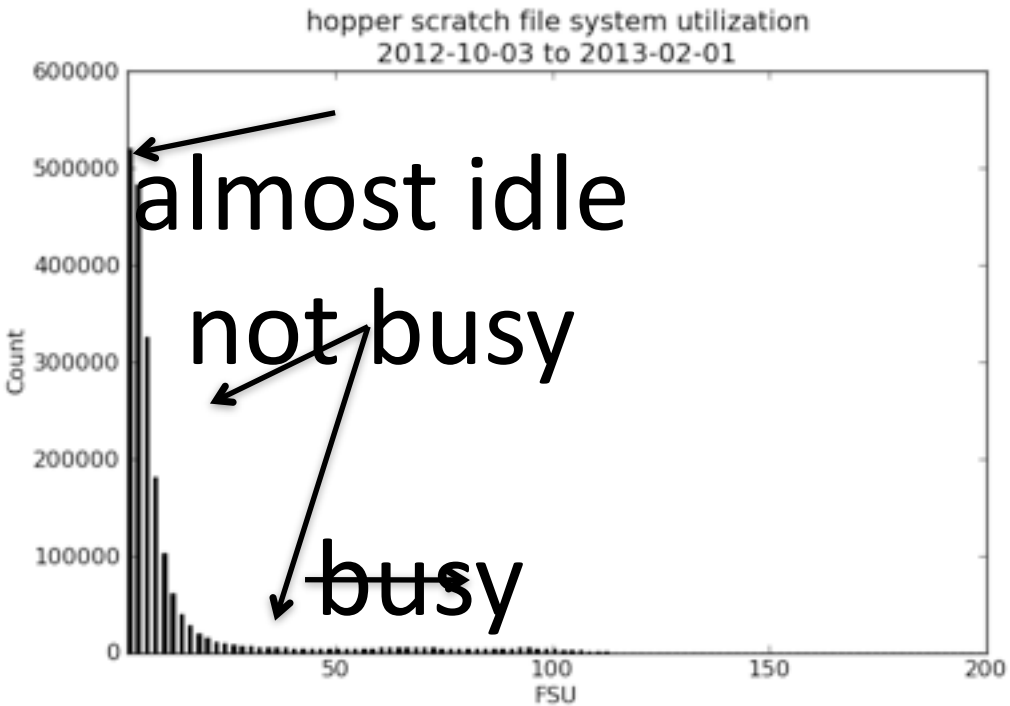




How often does this happen?

How often is */scratch* busy and not

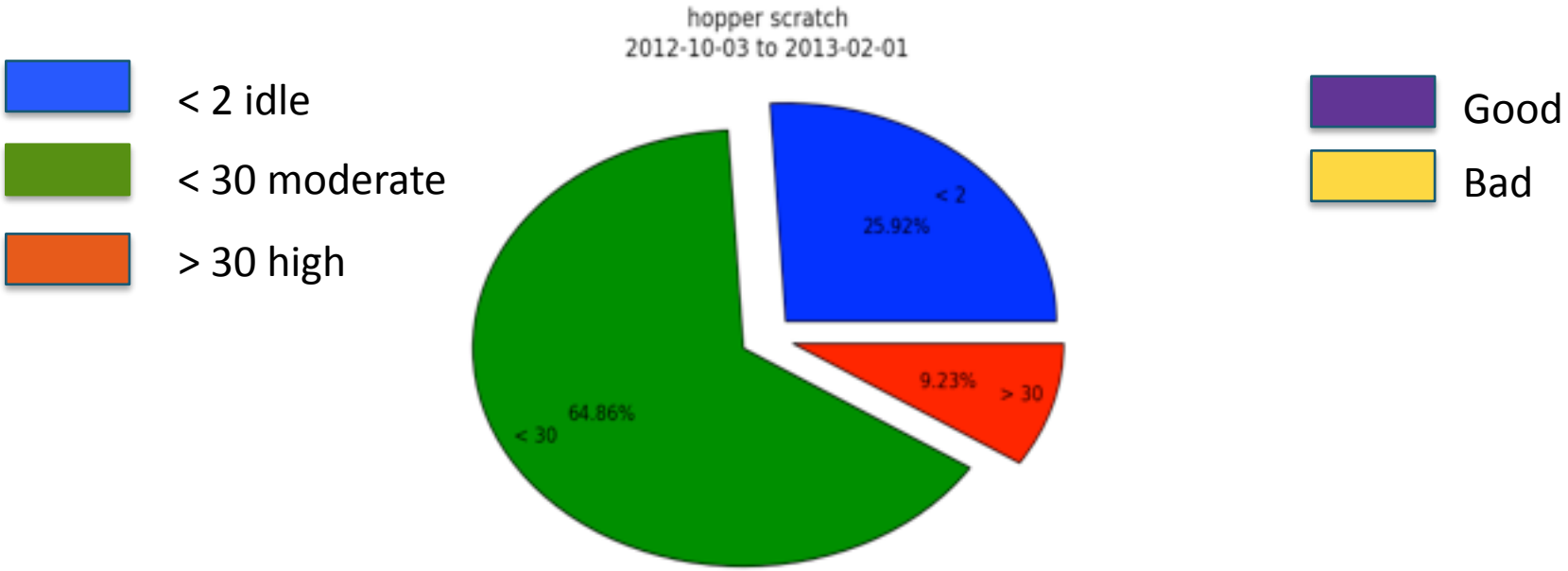
- The distribution of FSU values is dominated by values in the bin for 'less than 2' and has a long tail



- **17,280 observations/day for 120 days**
- **A quarter of all observations have FSU < 2**
- **Anything above FSU=30 is busy**

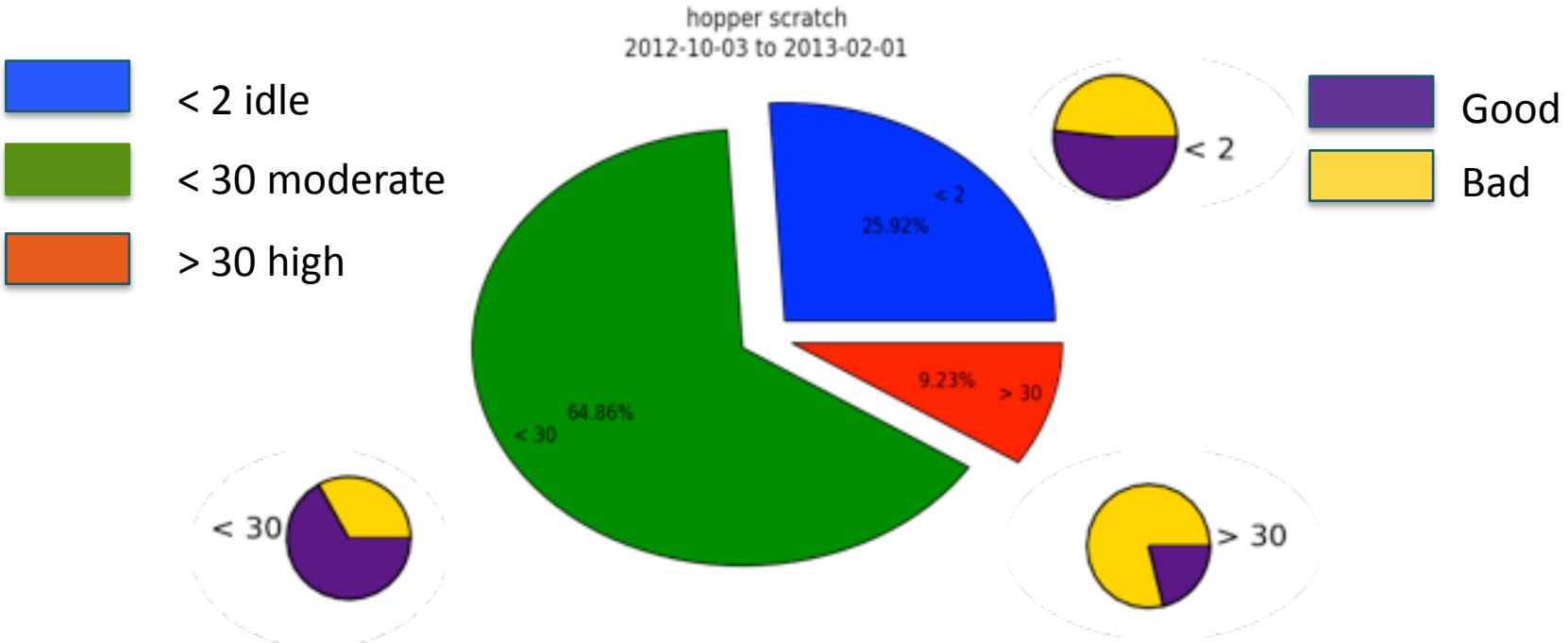
Idle, Moderate, and High Activity

- Looked at this way you can see that the system is actually quite busy a fair amount of the time



Idle, Moderate, and High Activity

- Looked at this way you can see that the system is actually quite busy a fair amount of the time



And at high FSU the small I/Os are especially prevalent!

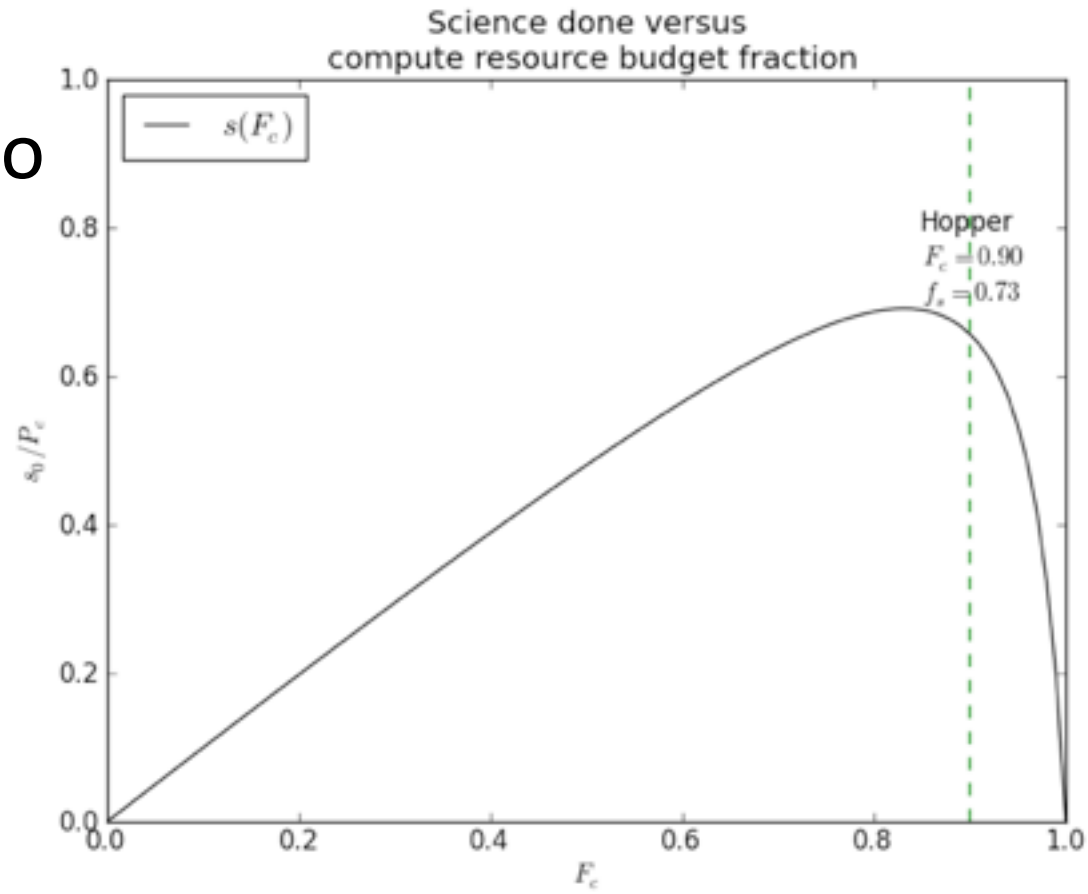
A Cost Model

- The budget for an HPC resource is divided between compute and I/O
- Time spent computing produces a steady amount of data that needs to be read/written
- Time spent reading/writing is time not spent computing
- The amount of science that gets done is proportional to the time spent computing

Together, these specify a system of equations

A Cost Model

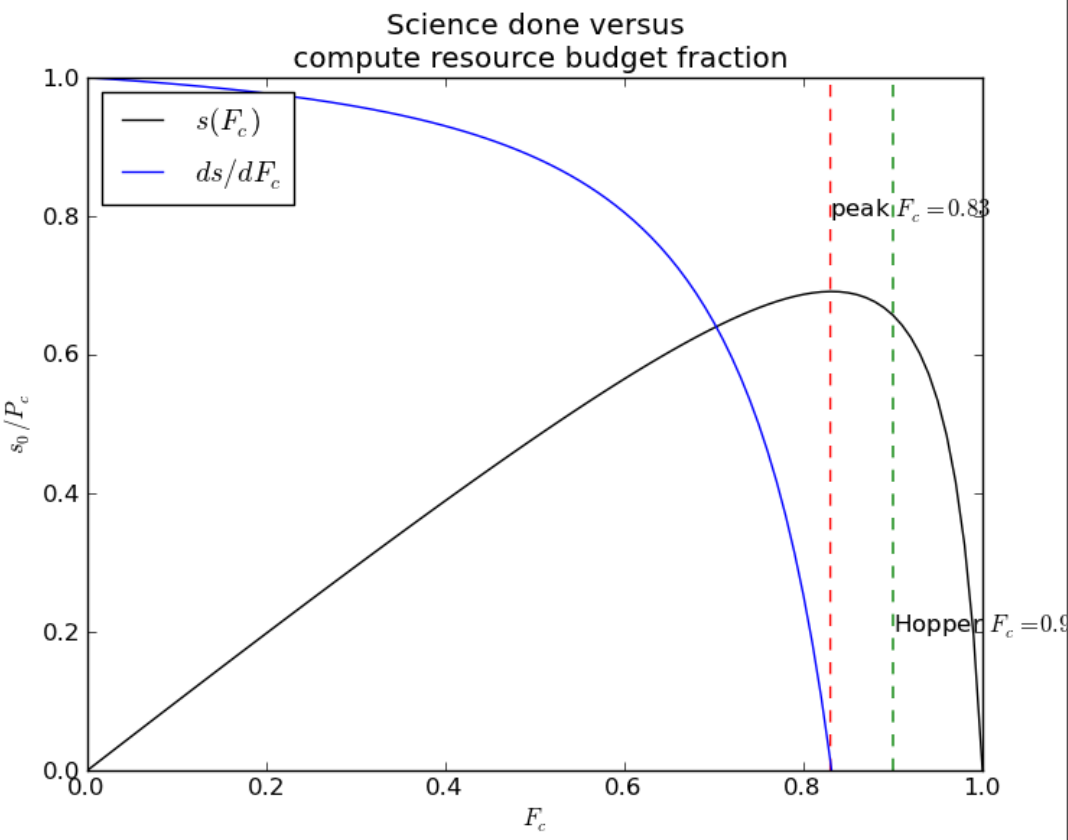
We can solve the system of equations to express the throughput (science per dollar spent) in terms of the fraction of the budget for compute resources.



A Cost Model

It is a straight forward optimization problem to solve for the peak. It turns out that the optimum amount of science occurs when the fraction spent on compute equals the fraction of the time spent doing science:

$$F_c = f_s$$



Planning for a New System

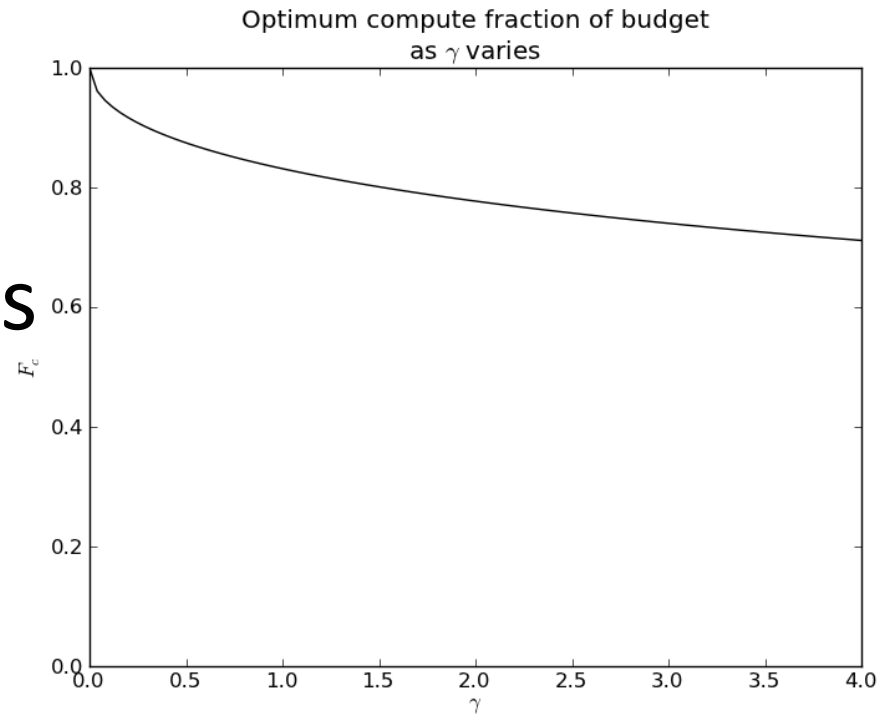
Calibration of the model depended on four descriptive constants:

- The unit cost of nodes, P_n
- The unit cost of I/O bandwidth, P_b
- The amount of science a node does, r_0
- The amount of I/O a unit of bandwidth does, r_1

$$\gamma = \frac{r_0 P_n}{r_1 P_b}$$

Planning for a New System

Given the calibration we've done with Hopper, you can plot the optimum balance as the value of γ varies. Nodes twice as powerful double γ and double the I/O halves it.



Conclusions

- New tools and analysis allowed us to characterize the workload on Hopper
- A simple cost model for HPC resources allows us to find the optimum budget fraction for I/O resources
- The workload result calibrates the cost model
- The optimum budget fraction depends on the system design in a simple way that depends on four characteristics of its construction

Future Work

- Refine the workload characterization
- Incorporate more detail in the underlying model
- Compare with other currently deployed systems

Thank You

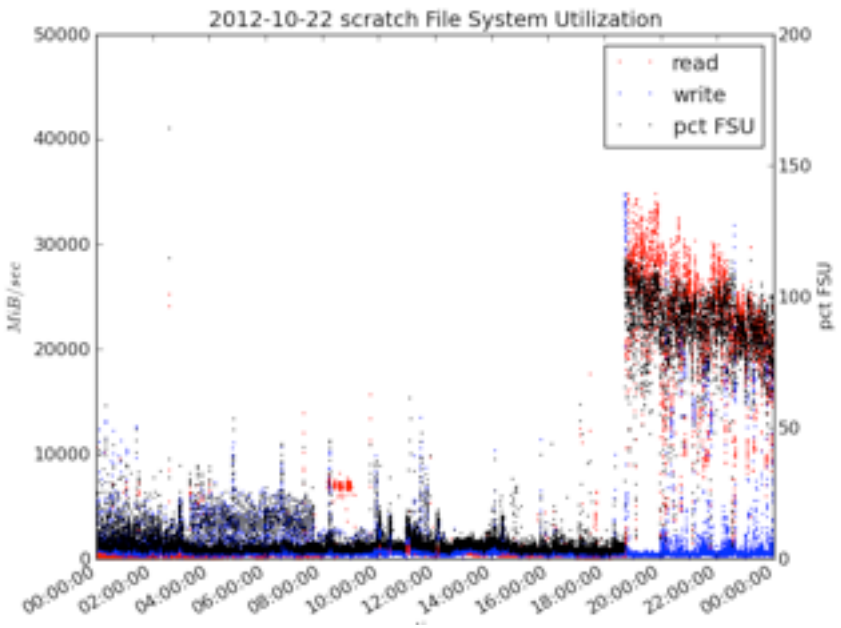
- Questions

National Energy Research Scientific Computing Center

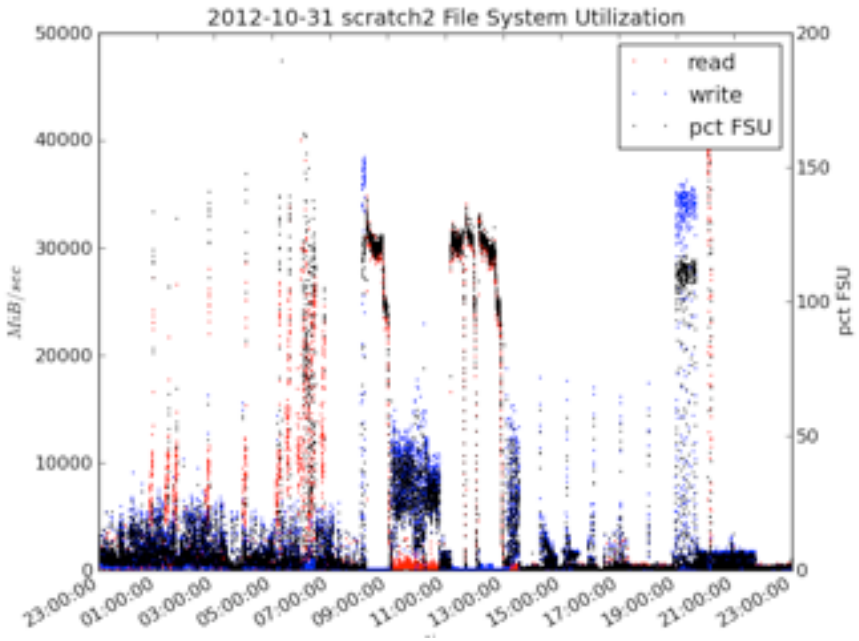
/scratch vs /scratch2

- The two scratch file systems are constructed very nearly identically, but their workloads differ
- They look different, but how different are they?

/scratch

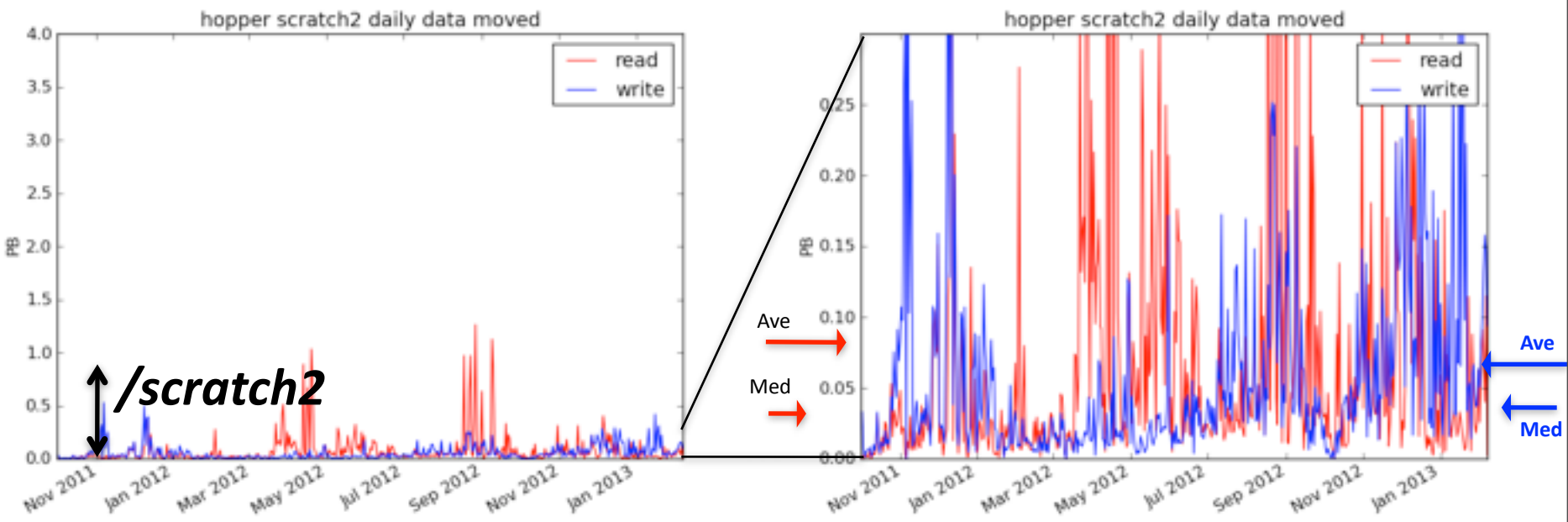


/scratch2



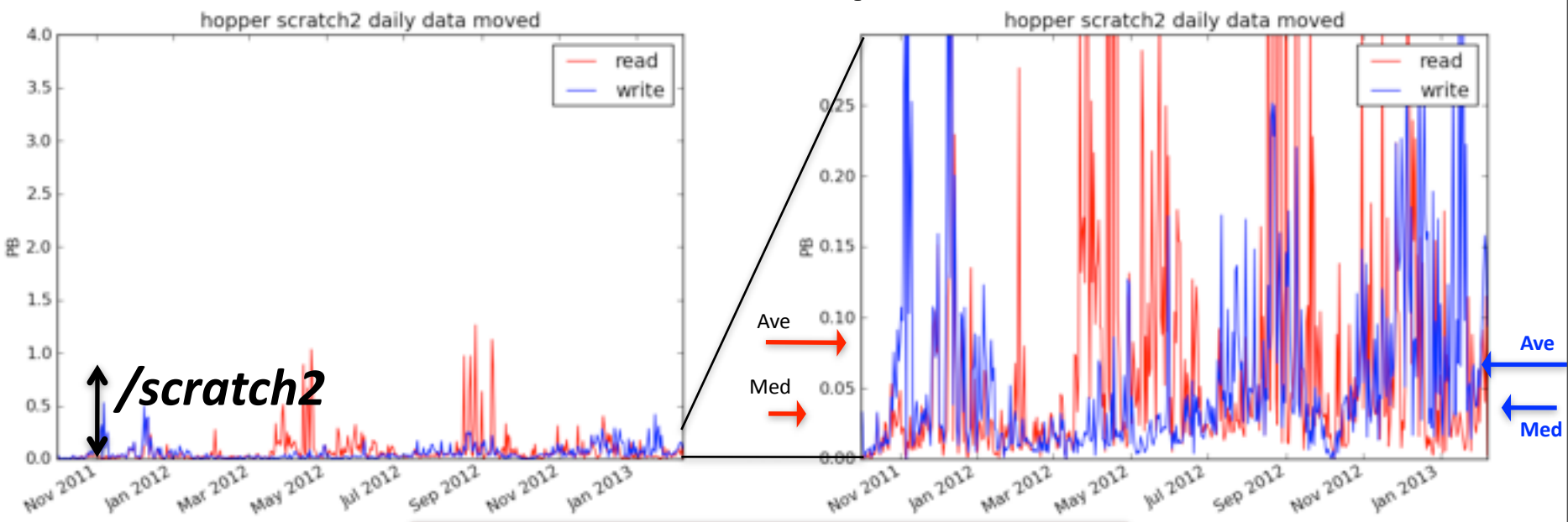
488 days on */scratch2*

- The read I/O traffic to */scratch2* is about 1/2 that to */scratch*
- The write I/O traffic is about 3/4
- */scratch2* is more episodic



488 days on */scratch2*

- The read I/O traffic to */scratch2* is about 1/2 that to */scratch*
- The write I/O traffic is about 3/4
- */scratch2* is more episodic



Is that the only difference?

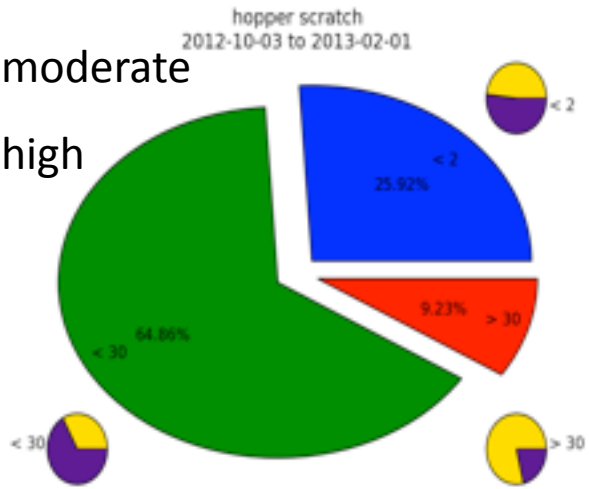
Idle, Moderate, High and Good/Bad

< 2 idle

< 30 moderate

> 30 high

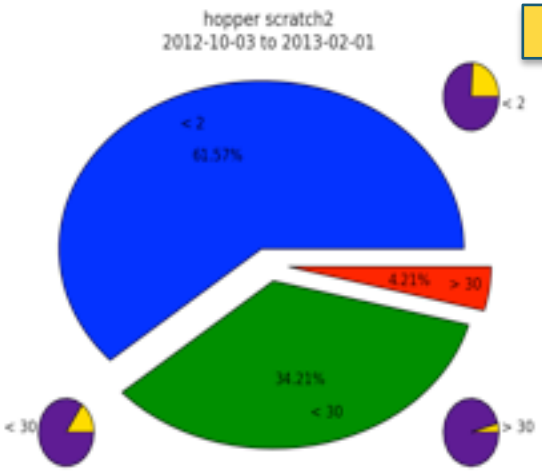
/scratch



/scratch2

Good

Bad

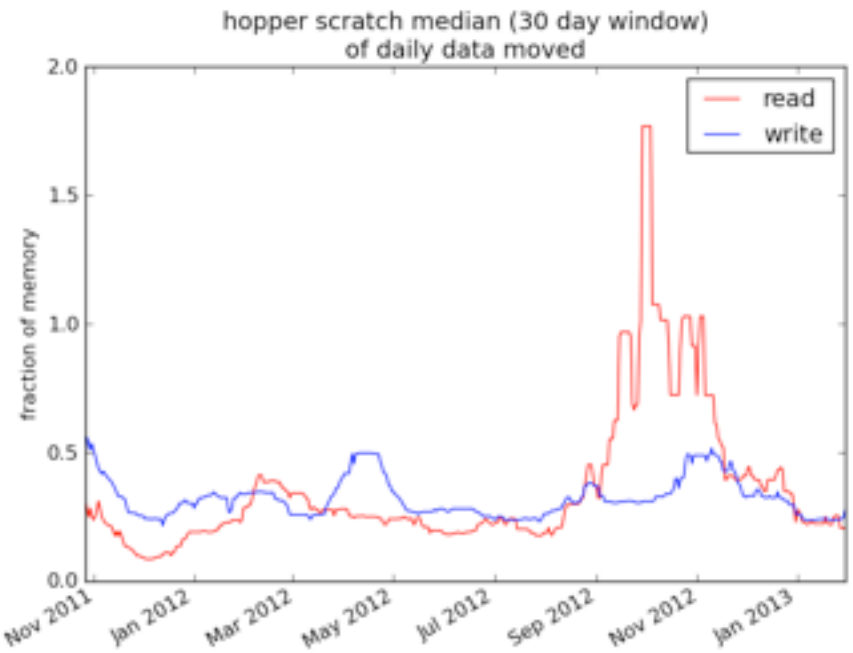


- */scratch* **does a lot of "bad" I/O especially at its peak of activity**
- */scratch2* **does peak activity mainly with "good" I/O**

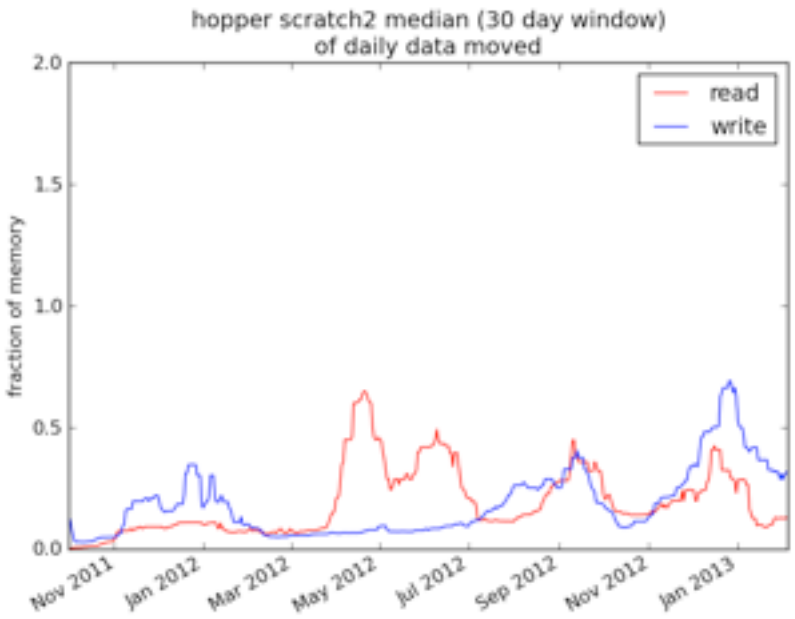
Typical Behavior of */scratch* vs /

- */scratch* does a lot more I/O
- */scratch2* has more episodic I/O

/scratch



/scratch2

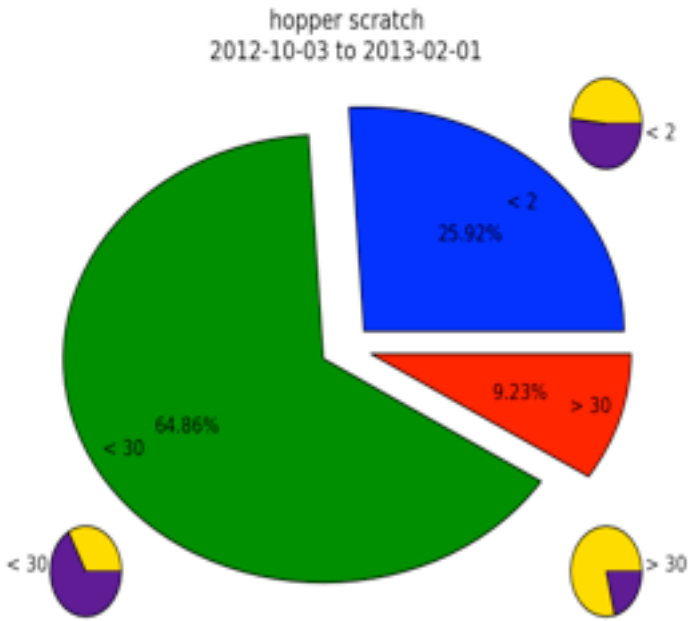


OST Utilization

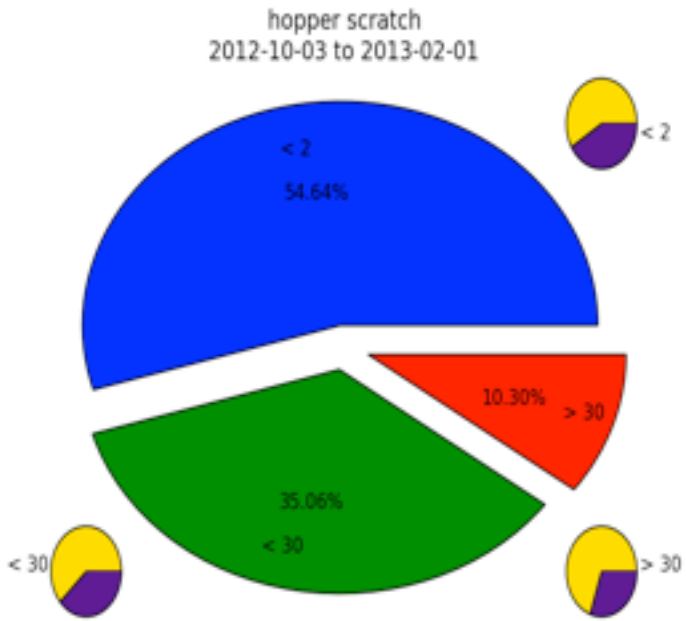
- Recollect that the FSU is the combination of 156 separate OST utilization values
- If you look at each OST in isolation the situation is actually a little worse
- If one OST is running at OST Util=100 the FSU as a whole may still be in the first bin (less than 2), but if that's the OST you want to use it is still going to cause you delays
- Much of the time when an individual OST is busy it's because they all are (thus leading to high FSU)

Idle, Moderate, and High FSU vs OST util

- Now you can see that the OSTs are worse off
- More often idle
- More often busy



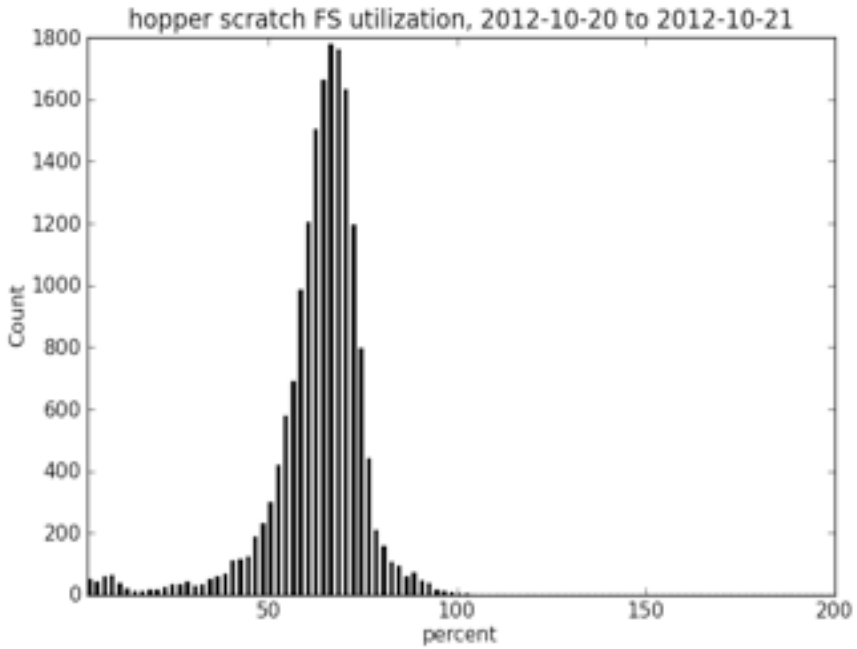
FSU



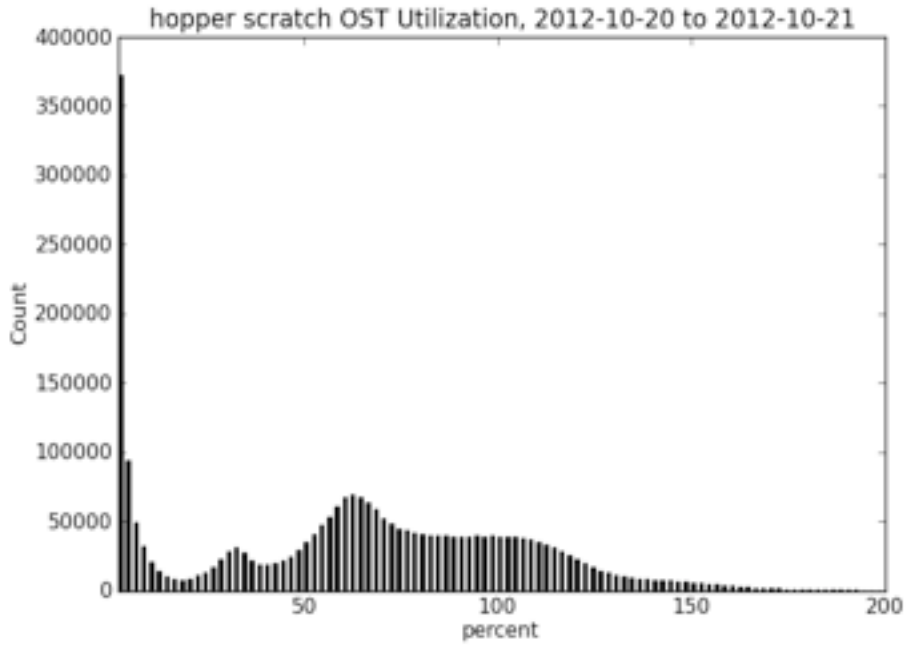
OST util

That Busy Day that was Deceptively Quiet

- Histograms for FSU (left) and OST utilization (right) for that one day
- All 156 OSTs were constantly under heavy load



FSU



OST util

Data Rate is Not a Proxy for Load

- What looks like low activity may actually be high activity that is performing poorly.
- This mainly comes about due to small transaction size between the server and the disk. The I/O becomes latency bound.
- It can also come about in other ways that we won't cover
  - Contention at the server
  - Contention at the disk controller
  - Contention in the HSN
  - Fragmentation of the RPCs
- So what *is* the correct metric?

An Improved Metric

In order to establish an improved metric we need to introduce or improve these things:

- A model for what is happening on the file system
- Experimental results to calibrate the model
- Data to support applying the model
- Observations that show application of the model is producing useful results

A Disk I/O Model

- The time required for a disk I/O is the sum of a constant latency (a_0) and the time proportional to the I/O transaction ($a_1 b$), for an I/O of b bytes
- Use separate coefficients for read and write
- R_b is the number of read I/Os of size b on a given OST (in a given interval), and likewise W_b for write I/Os
- The OST utilization is

$$U = \sum_b R_b(r_0 + r_1 b) + W_b(w_0 + w_1 b)$$

Calibration Experiments

- Drive an OST with a load that has known transaction size
- Ensure that there is a disk seek between each I/O
- Use sufficient resources to reach the OST's peak I/O rate (for that I/O pattern)
- Count the number of I/Os during the interval
- Repeat for a variety of transaction sizes

A File System Utilization Metric

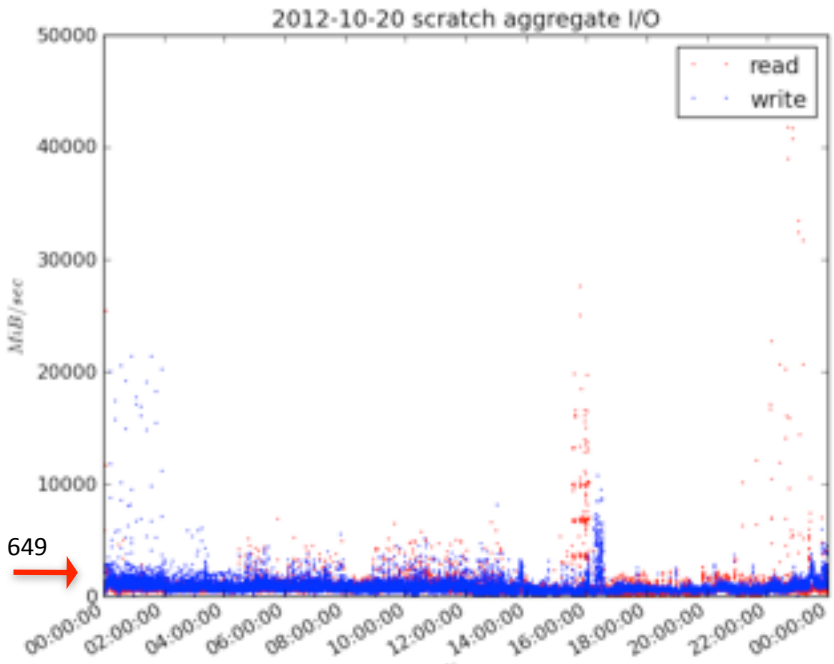
- We now have all the information we need to calculate the OST utilization every five seconds on all 156 OSTs
- The average of those 156 individual values is the File System Utilization (FSU) for that five second interval
- But how much of an effect do these small I/Os actually have on file system performance?

Caveats on Using the Term "Percent FSU"

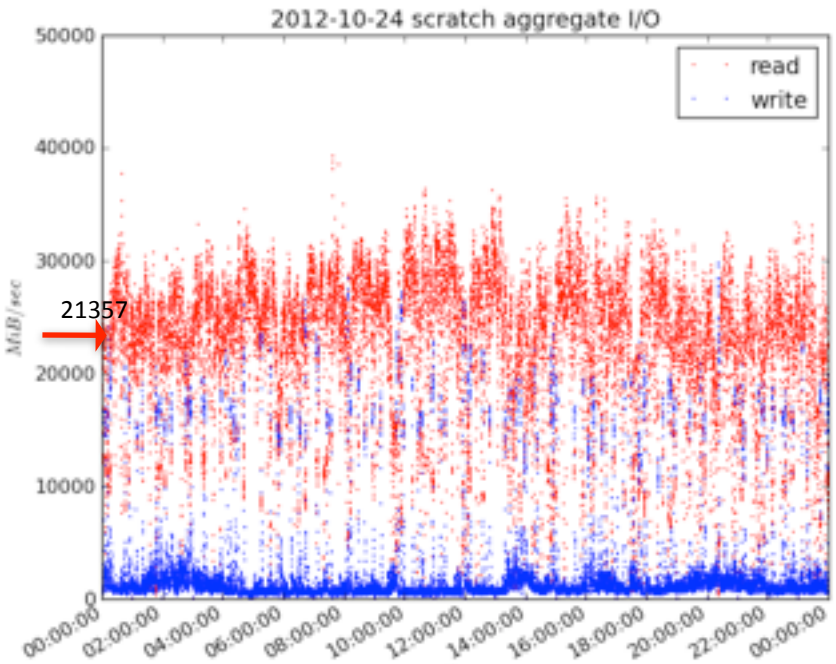
- The coefficients were chosen such that when an OST is handling as many I/Os as the maximum we expect (from the calibration experiments), then the utilization value will be 100
- The calculated value can exceed 100 if, for example, the I/Os are well clustered to avoid seek time
- The file system can be "maxed out" even when the value is below 100, since there are other resource constraints in the I/O path
- The FSU (OST util) percent does not express where the resource was in its actual utilization, but rather where the calculated value is in relation to an ideal maximum.

Metadata Behavior

- A typical day



- **An extreme day**



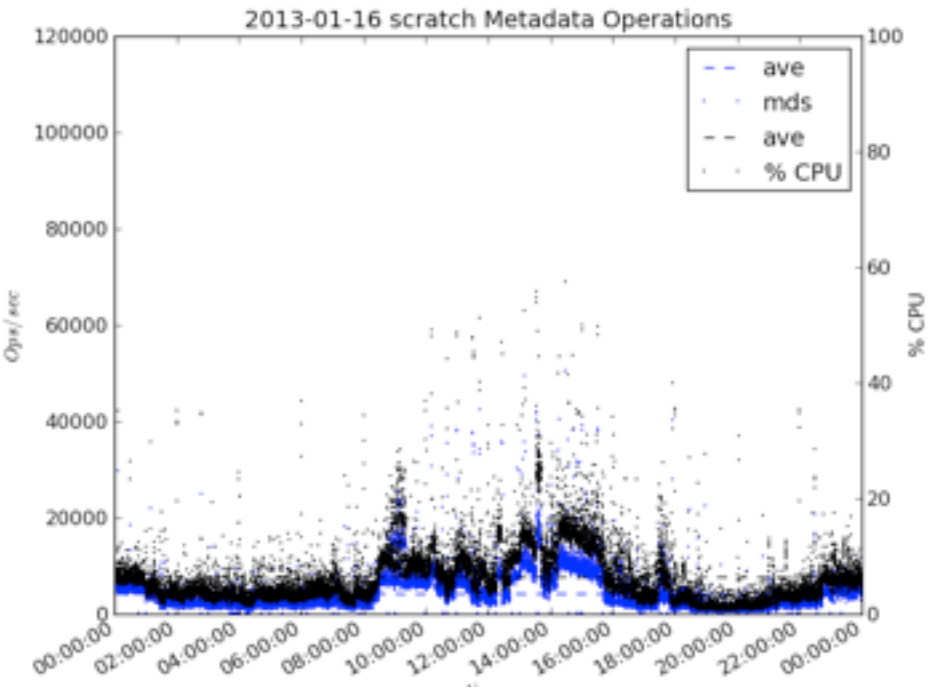
- **This is the raw data showing how fast the file system is processing metadata requests.**

Metadata

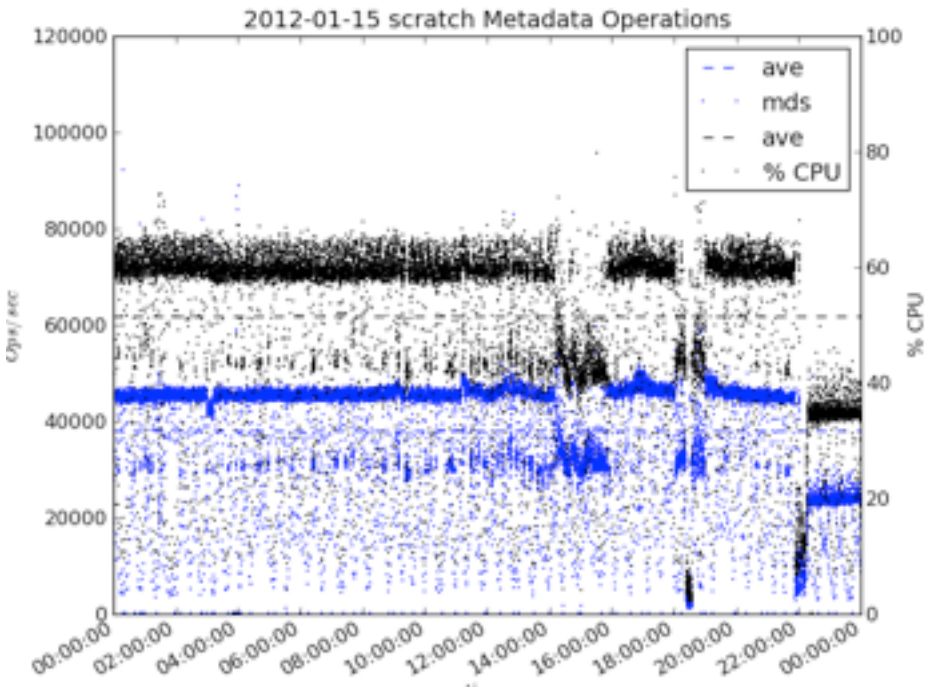
- How busy is the metadata server?

Metadata

- A typical day

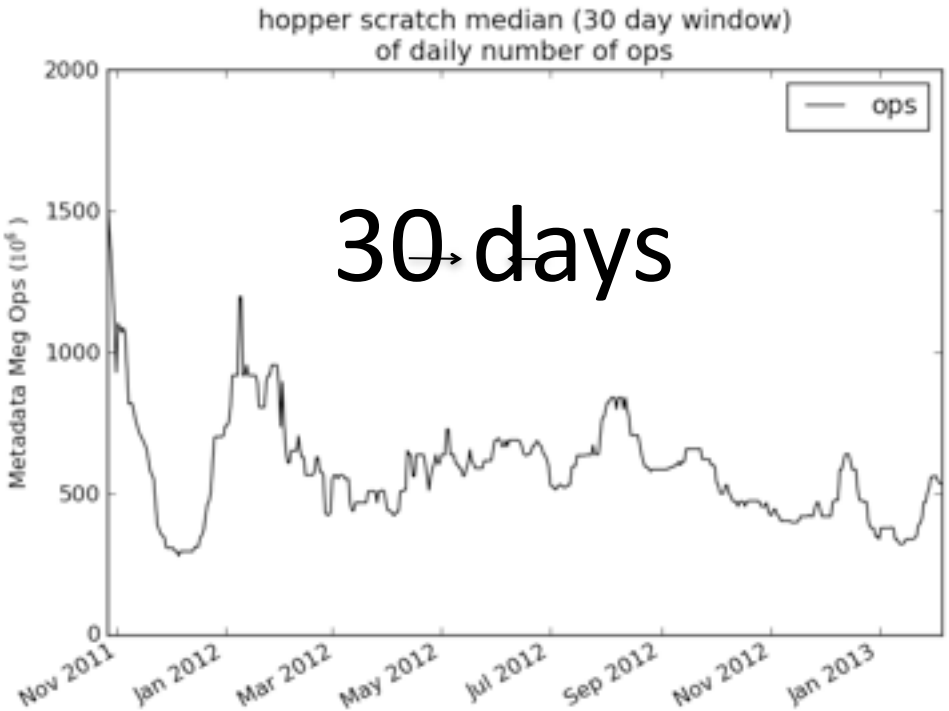


- An extreme day



488 days of Metadata Behavior

- For each day, what was the median activity level over the last thirty days?



MDS Utilization

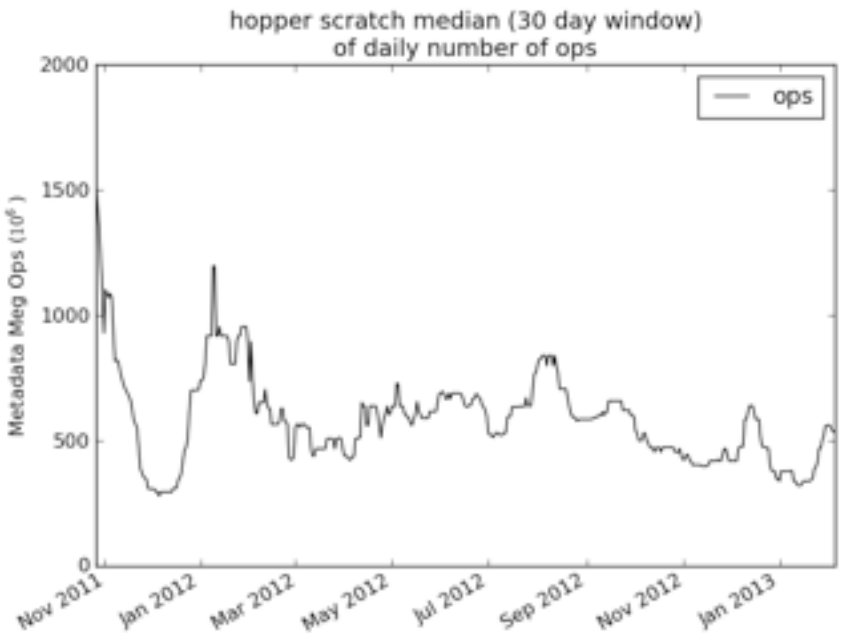
- What looks like low activity may be high activity with poor characteristics
- We have not performed this analysis
- We may have data to generate a more refined MDS utilization metric

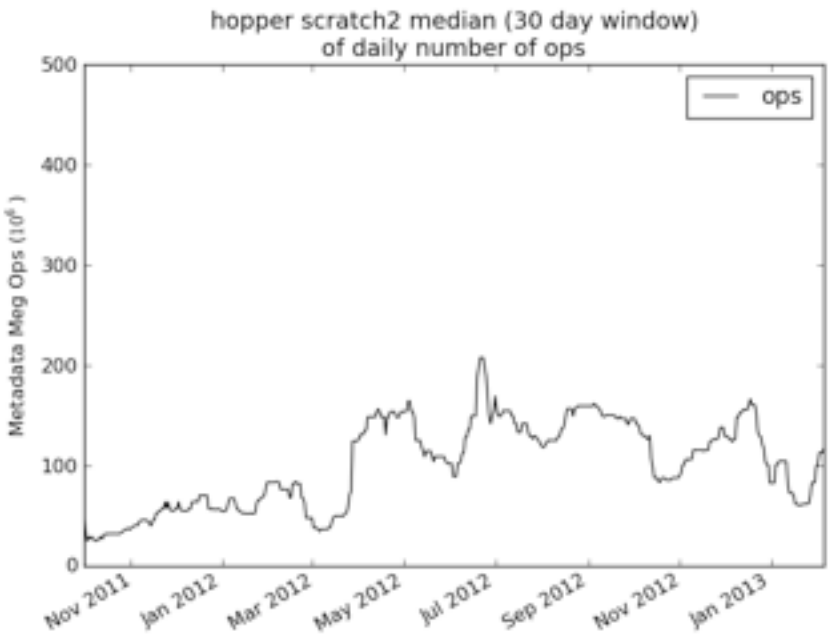
Idle, Moderate, and High Activity

- Presuming operations/second is the metric you can make the same three way split

(I have not done this graphic)

/scratch2 Metadata





/scratch2 Metadata (cont.)

- Side-by-side idle, moderate, high pie charts

(don't have this, yet)