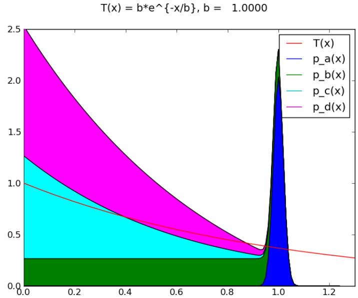
The Statistical Properties of Lustre Server-side I/O

A work in progress

Lustre User Group April 12, 2011

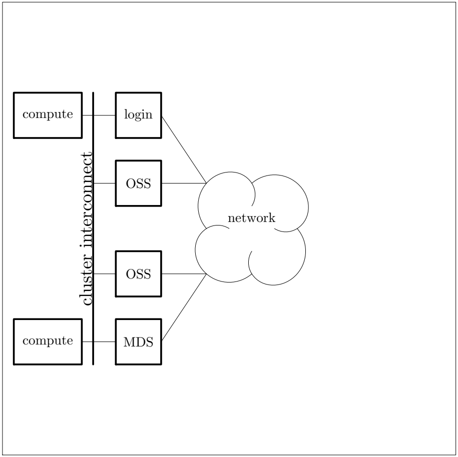


Andrew Uselton
National Energy Research Scientific Computing Center
Lawrence Berkeley National Lab

Contents

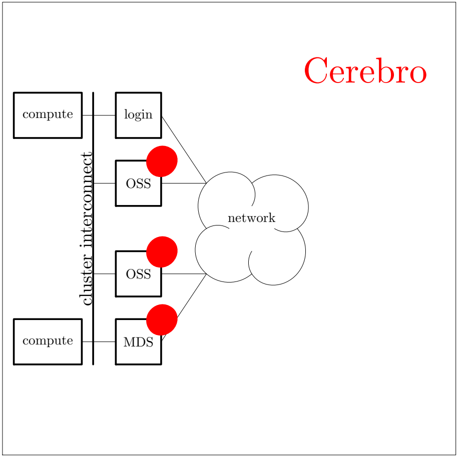
Outline

System layout



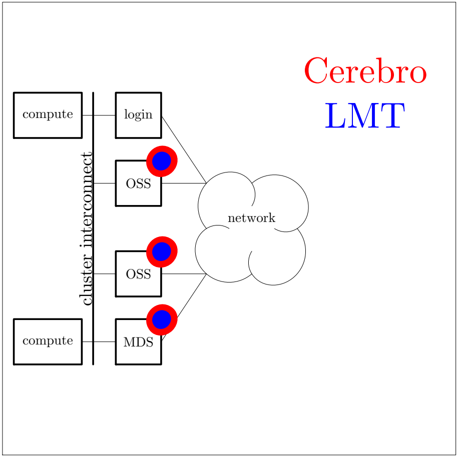
- milage may vary

Cerebro



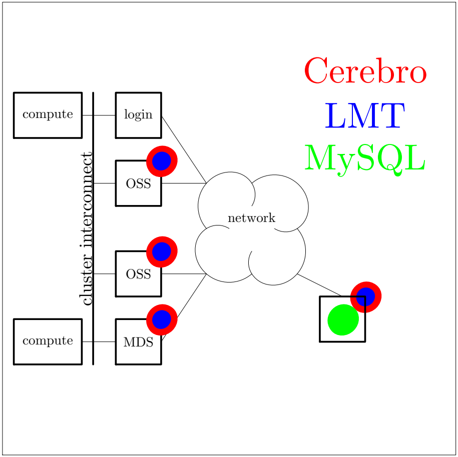
- lightweight
- extensible
- handles data transfer

LMT



- compiled libraries
- one per sever
- harvests `/proc` values

MySQL



- daemon receives packets (UDP)
- library processes contents
- db stores values
- cron job summarizes (optionally ages)
- misc. tools for querying db

Data

- Bytes read
- Bytes written
- Inodes available
- Queue depths
- Operations (eg. `open()`) per second
- *many more*

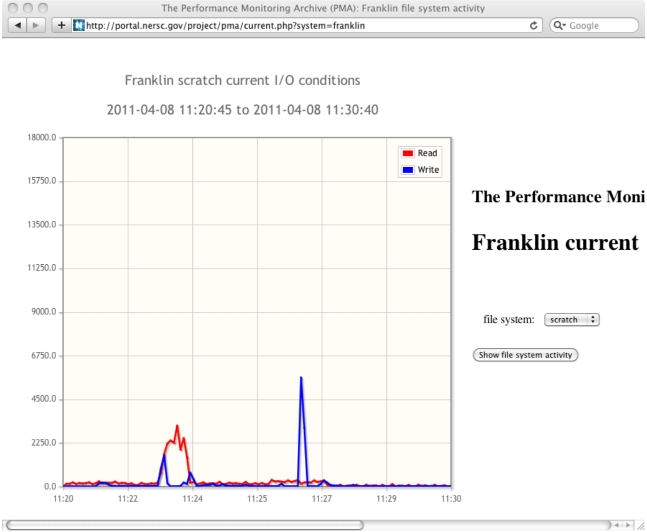
Data

- Bytes read
- Bytes written
- Inodes available
- Queue depths
- Operations (eg. `open()`) per second
- *many more*

Outline

1. LMT: The Lustre Monitoring Tool
2. **LMT Use Cases**
3. I/O System Balance
4. Occurence Histograms
5. A Simple Model
6. Conclusions

Monitoring Activity in Real Time



The Performance Monitoring Archive (PMA): Franklin file system activity

http://portal.nersc.gov/project/pma/current.php?system=franklin

Franklin scratch current I/O conditions

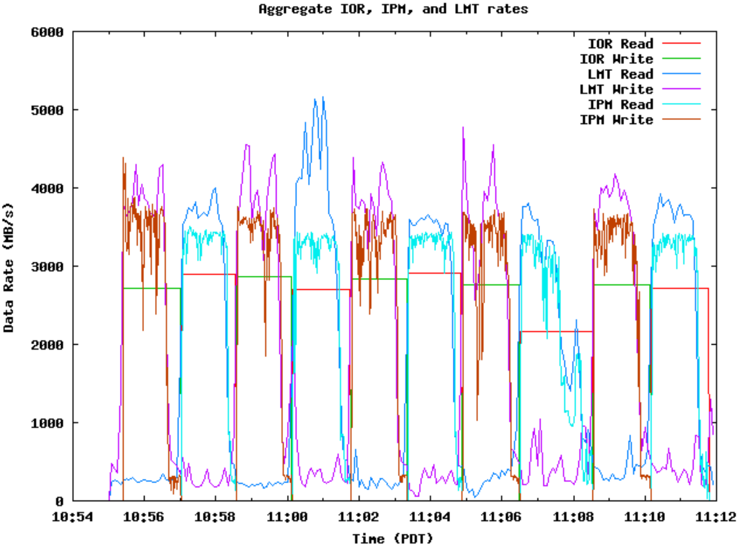
2011-04-08 11:20:45 to 2011-04-08 11:30:40

The Performance Moni

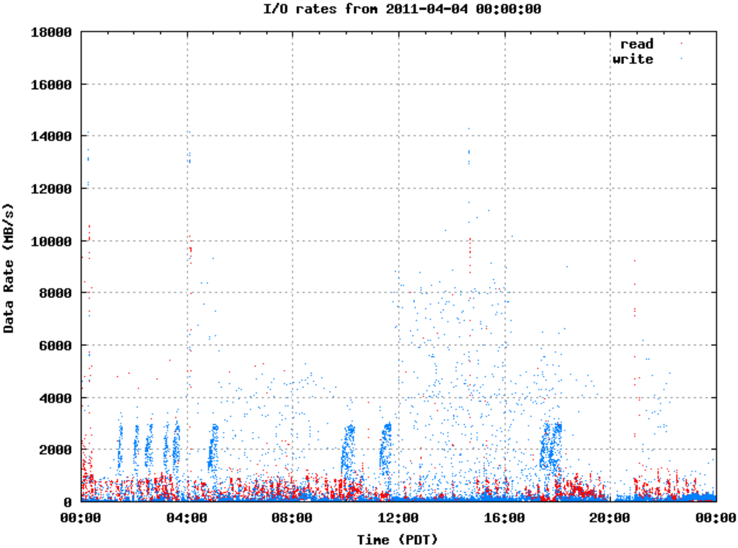
Franklin current

file system: scratch

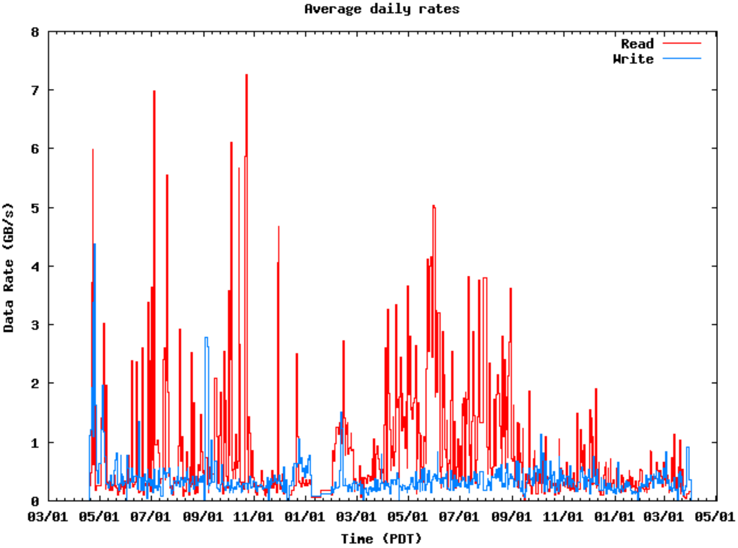
Detailed Performance Analysis



Gestalt of a Full Day of Activity

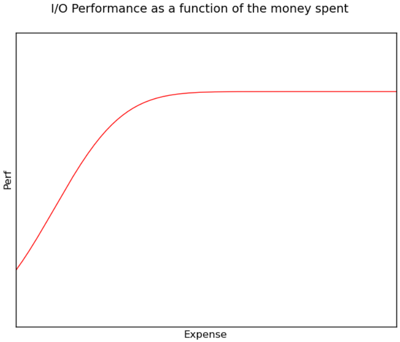


Monitoring Long Term Trends



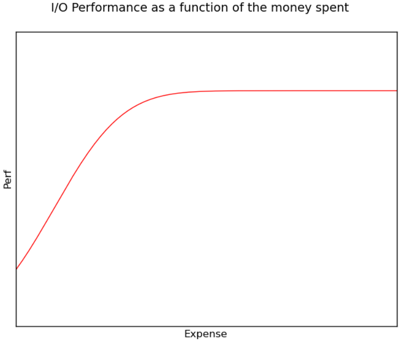
Outline

I/O System Balance - between cost and performance



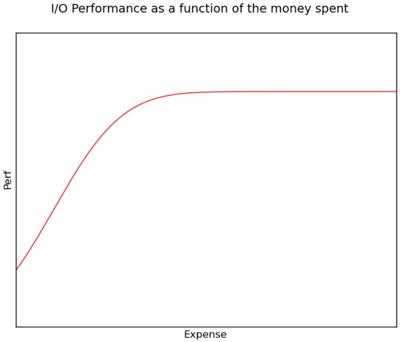
- More money spent means (we hope) better performance

I/O System Balance - between cost and performance



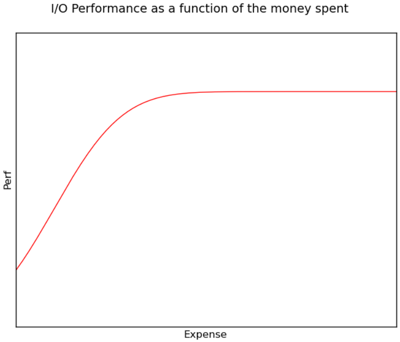
- More money spent means (we hope) better performance
- Upto a point

I/O System Balance - between cost and performance



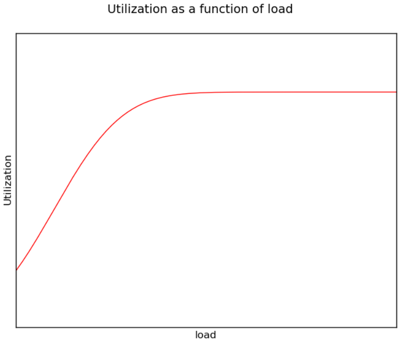
- More money spent means (we hope) better performance
- Upto a point
- How can you tell where that point is?

I/O System Balance - between cost and performance



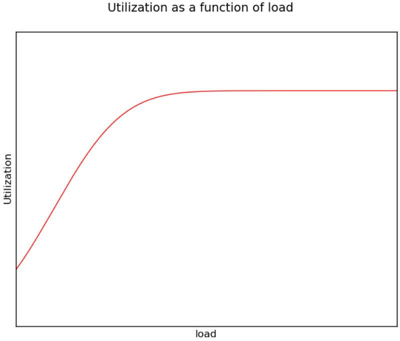
- More money spent means (we hope) better performance
- Upto a point
- How can you tell where that point is?
- The answer depends on both the I/O system and the workload

I/O System Balance - between I/O and compute capacity



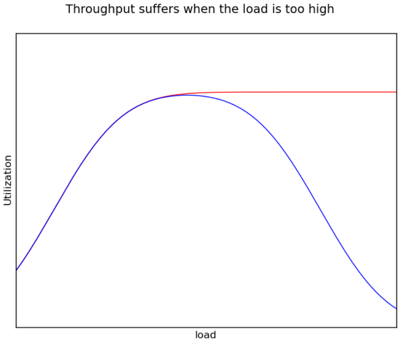
- We want to keep the compute resource near 100% utilized

I/O System Balance - between I/O and compute capacity



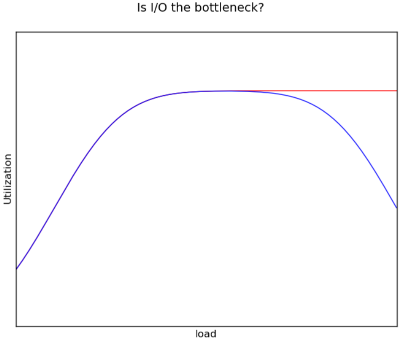
- We want to keep the compute resource near 100% utilized
- Job schedulers are designed to make this happen

I/O System Balance - between I/O and compute capacity



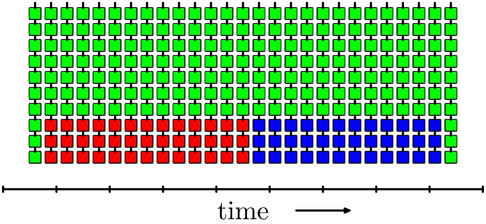
- We want to keep the compute resource near 100% utilized
- Job schedulers are designed to make this happen
- A clogged I/O system creates a hidden penalty

I/O System Balance - between I/O and compute capacity

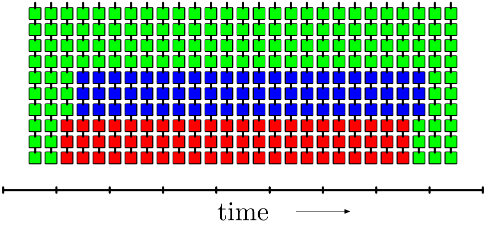


- We want to keep the compute resource near 100% utilized
- Job schedulers are designed to make this happen
- A clogged I/O system creates a hidden penalty
- Can we “buy” compute resource (cheaper) by buying I/O?

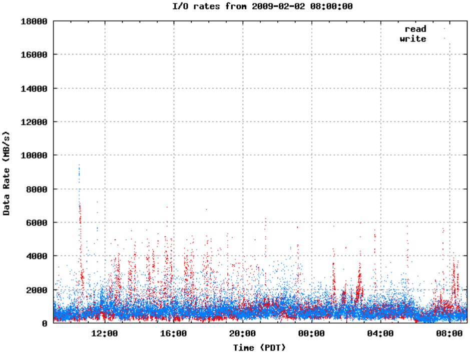
I/O Contention



I/O Contention

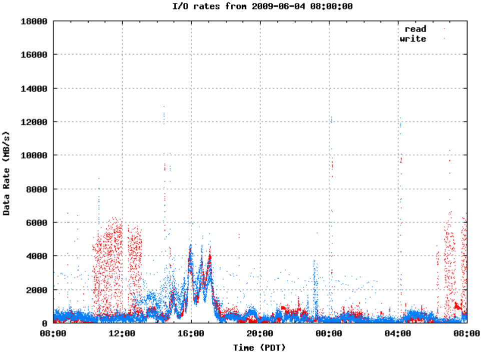


Case Study: April 2009 I/O Upgrade



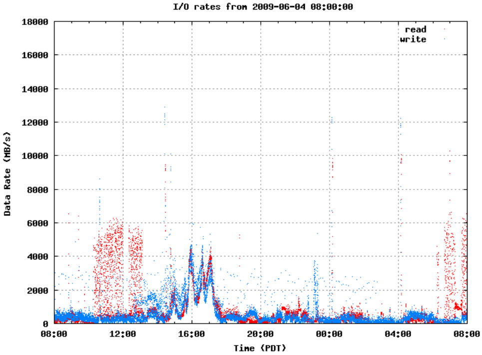
- I/O upgrade in April 2009 significantly improved performance

Case Study: April 2009 I/O Upgrade



- I/O upgrade in April 2009 significantly improved performance
- It is hard to see that fact in the before and after rate graphs

Case Study: April 2009 I/O Upgrade

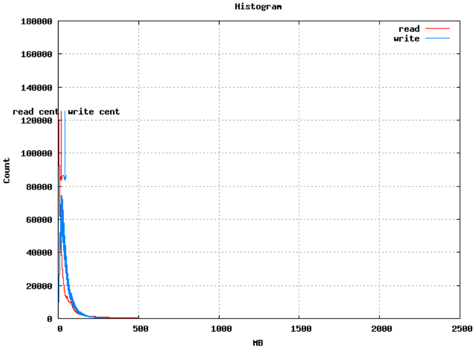


- I/O upgrade in April 2009 significantly improved performance
- It is hard to see that fact in the before and after rate graphs
- Were the workloads on the two days even comparable?

Outline

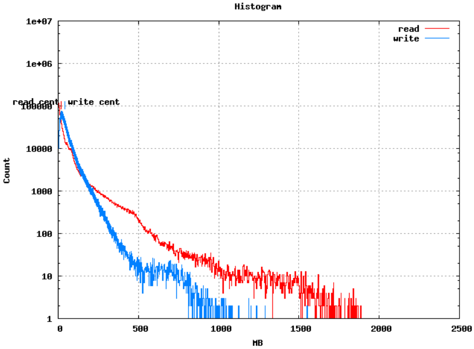
1. LMT: The Lustre Monitoring Tool
2. LMT Use Cases
3. I/O System Balance
4. **Occurence Histograms**
5. A Simple Model
6. Conclusions

Hisogram: Before April 2009



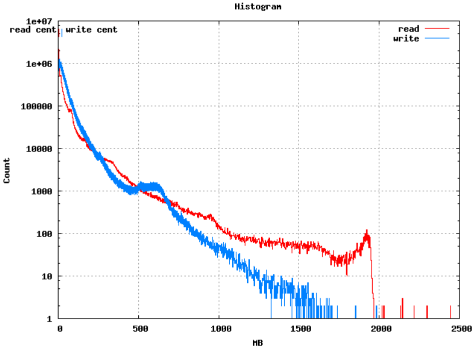
- A histogram shows the frequency that I/O of a particular size occurred

Hisogram: Before April 2009



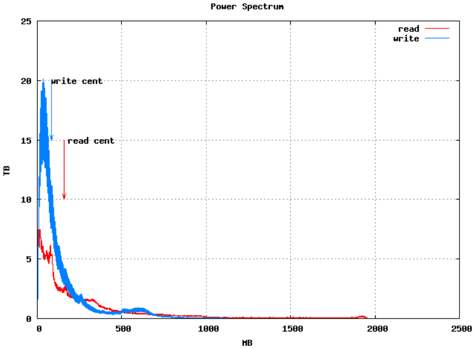
- A histogram shows the frequency that I/O of a particular size occurred
- A log scale makes it easier to see the shape of the distribution

Hisogram: Before April 2009



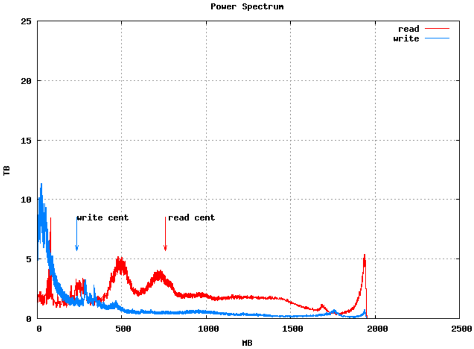
- A histogram shows the frequency that I/O of a particular size occurred
- A log scale makes it easier to see the shape of the distribution
- A histogram can compile data over an arbitrary time scale

Power Spectrum: Before and After



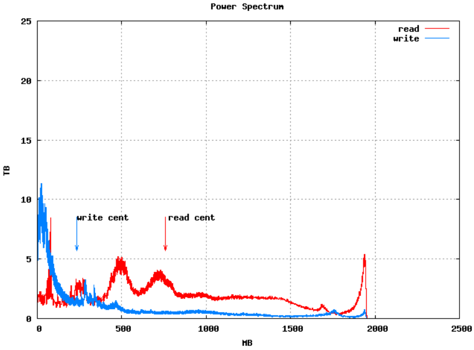
- A power spectrum mutiplies the histogram by the size of the observations

Power Spectrum: Before and After



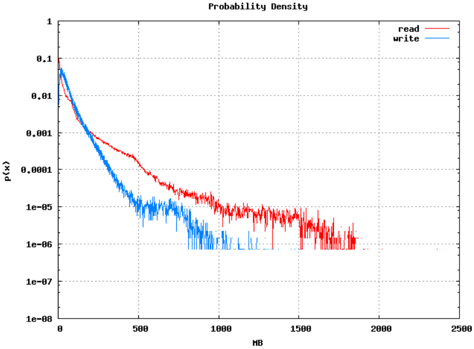
- A power spectrum mutiplies the histogram by the size of the observations
- before and after data (without log scale)

Power Spectrum: Before and After



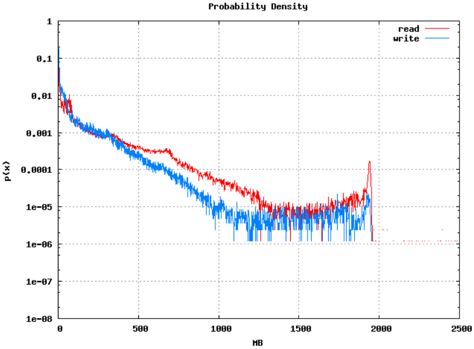
- A power spectrum mutiplies the histogram by the size of the observations
- before and after data (without log scale)
- This emphasizes the significance of the larger transactions

Probability Density: Before and After Comparrison



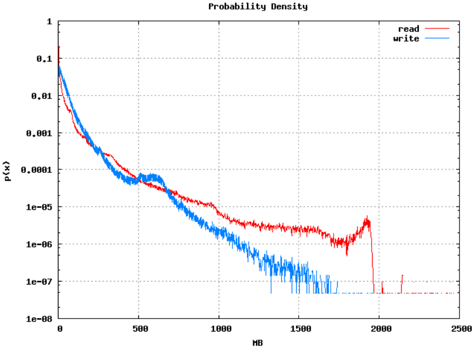
- one day of data before a major upgrade

Probability Density: Before and After Comparrison



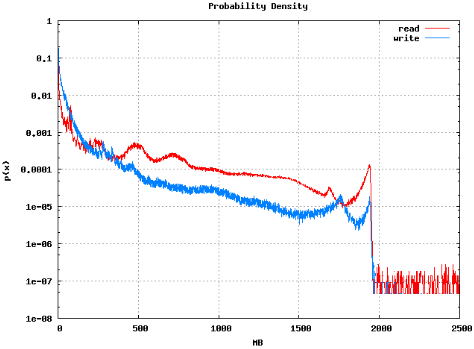
- one day of data before a major upgrade and after
- but are the two days really comparable?

Probability Density: Before and After Comparrison



- one day of data before a major upgrade and after
- but are the two days really comparable?
- We can expect (hope) that the workload over long timescales is constant

Probability Density: Before and After Comparrison

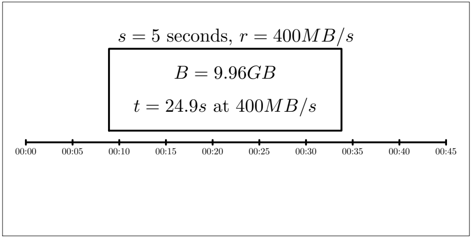


- one day of data before a major upgrade and after
- but are the two days really comparable?
- We can expect (hope) that the workload over long timescales is constant

Outline

1. LMT: The Lustre Monitoring Tool
2. LMT Use Cases
3. I/O System Balance
4. Occurence Histograms
5. **A Simple Model**
6. Conclusions

A Simple Model



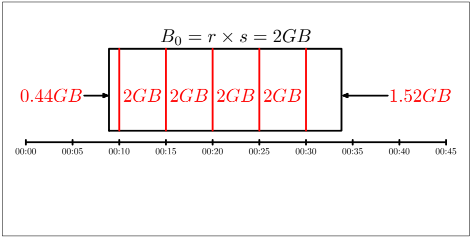
$s = 5$ seconds, $r = 400MB/s$

$B = 9.96GB$

$t = 24.9s$ at $400MB/s$

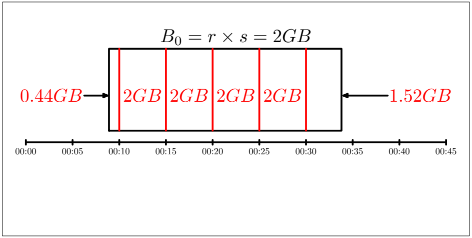
- A transaction arrives at some arbitrary point

A Simple Model



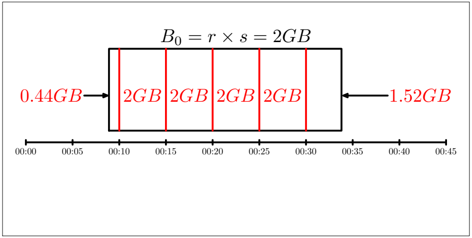
- A transaction arrives at some arbitrary point
- It is split accross multiple obseration intervals

A Simple Model



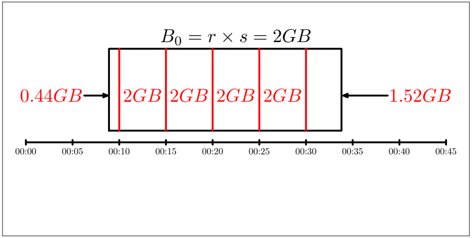
- A transaction arrives at some arbitrary point
- It is split accross multiple obseration intervals
- Assumptions:
    - All the I/O in the transaction comes in together

A Simple Model



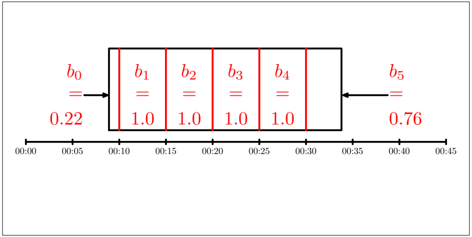
- A transaction arrives at some arbitrary point
- It is split accross multiple obseration intervals
- Assumptions:
  - All the I/O in the transaction comes in together
  - The I/O proceeds at its maximum rate until complete

A Simple Model



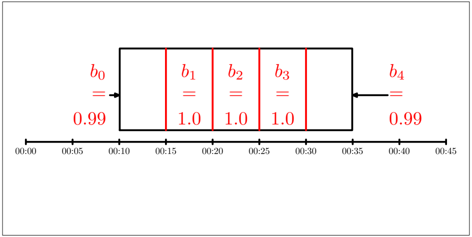
- A transaction arrives at some arbitrary point
- It is split accross multiple obseration intervals
- Assumptions:
  - All the I/O in the transaction comes in together
  - The I/O proceeds at its maximum rate until complete
  - One transaction at a time

Big Transactions $T > 1$, $(B > B_0)$



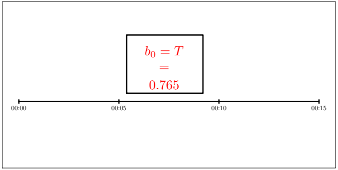
- It simplifies things to express the transaction size as a multiple of the maximum observation size: $T = B/B_0$
- $b_0 + b_5 < 1$ and $n = \lfloor T \rfloor + 2$

Big Transactions $T > 1$, $(B > B_0)$



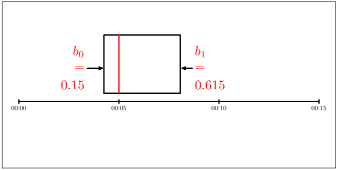
- It simplifies things to express the transaction size as a multiple of the maximum observation size: $T = B/B_0$
- $b_0 + b_5 < 1$ and $n = \lfloor T \rfloor + 2$
- $b_0 + b_4 > 1$ and $n = \lfloor T \rfloor + 1$

Small Transactions $T < 1$, ($B < B_0$)



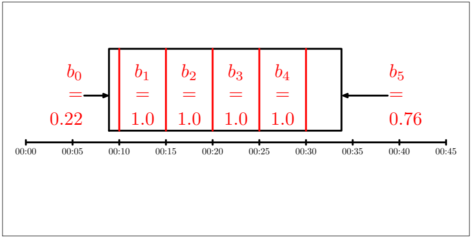
- Fits within one observation interval

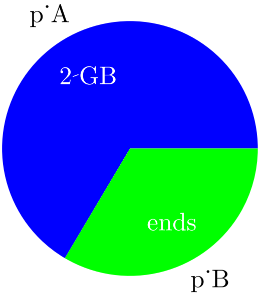
Small Transactions $T < 1$, $(B < B_0)$



- Fits within one observation interval
- Split accross two

Distribution of Observations for Large Transactions, $T > 1$

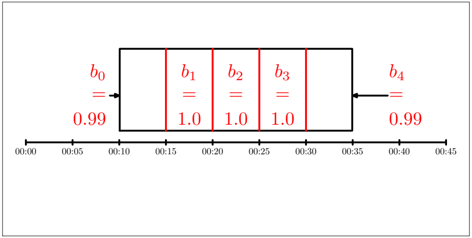


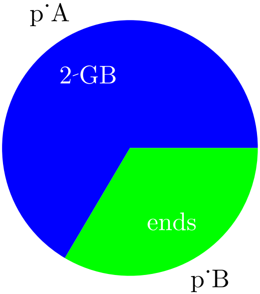


- $p_A = \frac{T-1}{T+1}$ chance that an observation is at $x = 1$

LMT: The Lustre Monitoring Tool
LMT Use Cases
I/O System Balance
Occurence Histograms
A Simple Model
Conclusions

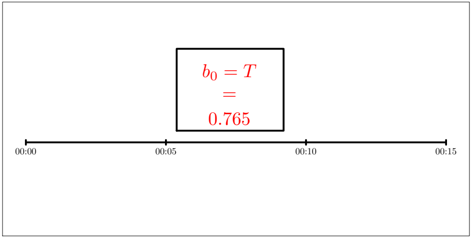
Distribution of Observations for Large Transactions, $T > 1$

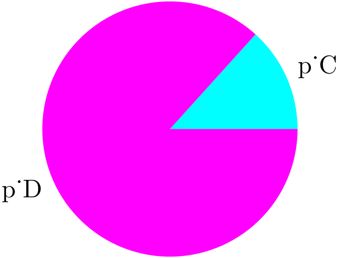




- $p_A = \frac{T-1}{T+1}$ chance that an observation is at $x = 1$
- $p_B = \frac{2}{T+1}$ chance that an observation is an *end*

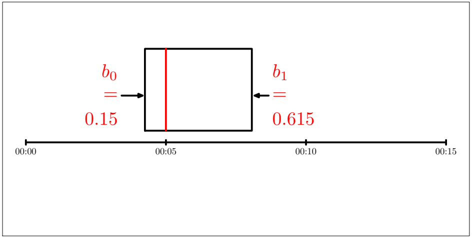
Distribution of Observations for Small Transactions, $T < 1$

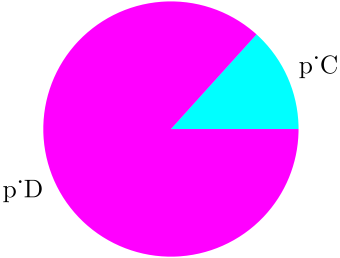




- $p_C = \frac{1-T}{1+T}$ chance that an observation is at $x = T$

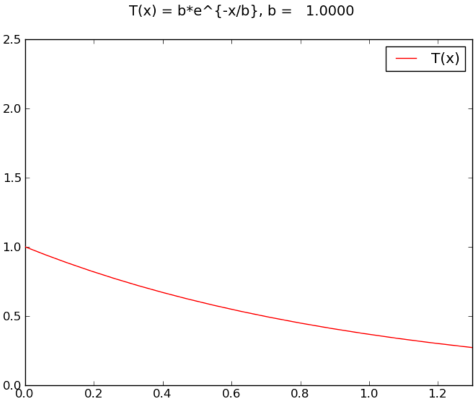
Distribution of Observations for Small Transactions, $T < 1$





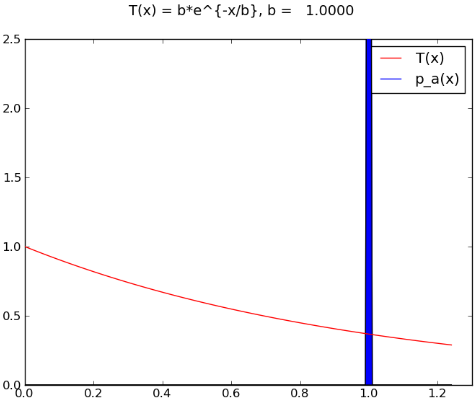
- $p_C = \frac{1-T}{1+T}$ chance that an observation is at $x = T$
- $p_D = \frac{2T}{1+T}$ chance that it is as piece of a *split* transaction

A Distribution $T(x)$ Over Transaction Sizes ($x > 1$)



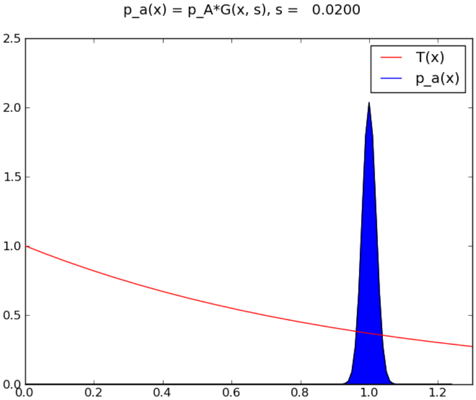
- Suppose $T(x) = \beta \exp(-x/\beta)$

A Distribution $T(x)$ Over Transaction Sizes ($x > 1$)



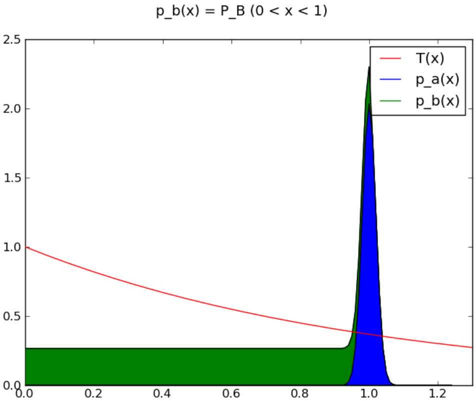
- Suppose $T(x) = \beta \exp(-x/\beta)$
- $p_a(x)$ is a delta function with $p_A = \int_1^\infty \frac{x-1}{x+1} T(x) dx$

A Distribution $T(x)$ Over Transaction Sizes ($x > 1$)



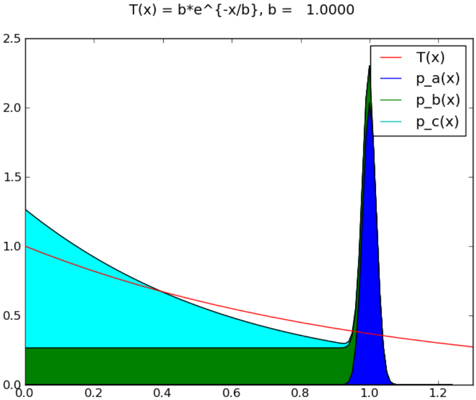
- Suppose $T(x) = \beta \exp(-x/\beta)$
- $p_a(x)$ is a delta function with $p_A = \int_1^\infty \frac{x-1}{x+1} T(x) dx$
- But suppose there is a variability $p_a(x) = p_A G(x, \sigma)$ (Gaussian)

A Distribution $T(x)$ Over Transaction Sizes ($x > 1$)



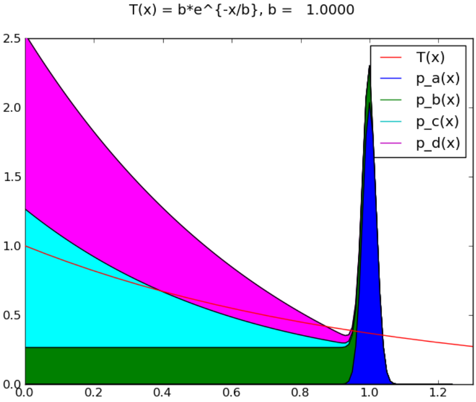
- Suppose $T(x) = \beta \exp(-x/\beta)$
- $p_a(x)$ is a delta function with $p_A = \int_1^\infty \frac{x-1}{x+1} T(x) dx$
- But suppose there is a variability $p_a(x) = p_A G(x, \sigma)$ (Gaussian)
- $p_b(x) = \int_1^\infty \frac{2}{x+1} T(x) dx$

A Distribution $T(x)$ Over Transaction Sizes ($x < 1$)



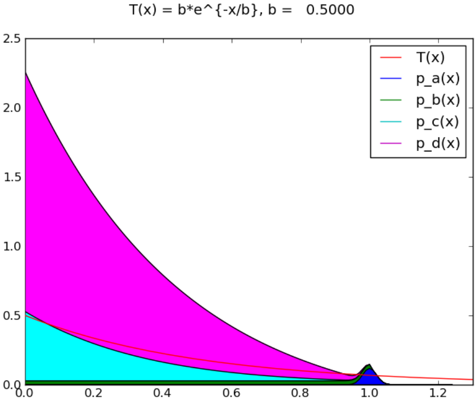
- $p_c(x) = (1 - x)T(x)$

A Distribution $T(x)$ Over Transaction Sizes ($x < 1$)



- $p_c(x) = (1 - x)T(x)$
- $p_d(x) = 2 * \int_x^1 T(x')dx'$

A Distribution $T(x)$ Over Transaction Sizes ($x < 1$)

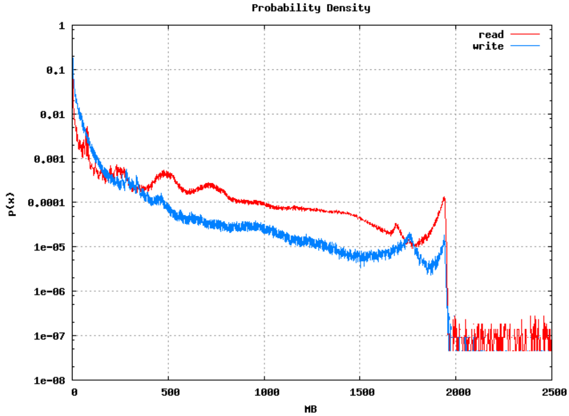


- $p_c(x) = (1 - x)T(x)$
- $p_d(x) = 2 * \int_x^1 T(x')dx'$

Outline

1. LMT: The Lustre Monitoring Tool
2. LMT Use Cases
3. I/O System Balance
4. Occurence Histograms
5. A Simple Model
6. **Conclusions**

Challenges



- curve shape
- curve fit estimates
- variability
- modes

Conclusions

- LMT is a source of data on file system performance
- I/O contention can result from a system imbalance
- System balance depends on the workload
- A statistical view can illuminate the workload pattern
- A very simple model helps relate the workload to the observations
- Using the observations to infer the workload is hard

Conclusions

- LMT is a source of data on file system performance
- I/O contention can result from a system imbalance
- System balance depends on the workload
- A statistical view can illuminate the workload pattern
- A very simple model helps relate the workload to the observations
- Using the observations to infer the workload is hard
- But not impossible

Questions?

- `https://computing.llnl.gov/linux/cerebro.html`
  Al Chu
- `http://code.google.com/p/lmt/`
  Herb Wartens, Jim Garlick

Additional Slides

(counts in millions)

	read	*write*	*both*	*read* × *rwrite*
count	450			
zero	205	124	70	
bin_0	267	178	141	
$zero/count$	0.46	0.28	0.15	0.07
$bin_0/count$	0.60	0.40	0.31	0.23

A large fraction of all observations are 0.0 and even more are in the first bin close to 0.0. If the *read* and *write* I/O streams were truely independent the occurence of both *read* and *write* observations simultaneously would be about the product of their separate probabilities.

Zeros, Month by Month