Heavy-tailed Distribution of Parallel I/O System Response Time

Bin Dong, Surendra Byna, and Kesheng Wu
Scientific Data Management group
Lawrence Berkeley National Laboratory, Berkeley, CA

PDSW2015: 10TH Parallel Data Storage Workshop, Austin, TX, November 16, 2015
Outline

• Motivation

• Response time sampling method

• Analysis results of response time
Estimating Response Time of I/O is Essential Element

• Data analysis query plan optimizing
  – Choose index or data organization with minimum read time
  – Scientific Data Services (SDS) framework, PostgresSQL, SciDB

• Data writing performance tuning
  – Select striping size, striping account, and other parameters to reduce write time
  – ExaHDF5, I/O Scheduler

• Simulator, Job Scheduler, Quality of service (QoS), etc.
Modeling Response Time for Parallel I/O

Response time of a single big file request $R$:

$$T = \max (t_1, t_2, \ldots, t_n) + \mu$$

$T$: response time of $n$ small requests

$t_1, t_2, \ldots, t_n$: response times of $n$ small requests

$\mu$: Split overhead, write

$I/O$ Servers in PFS (e.g., OST in Lustre)
Simplifying Response Time Model

\[ T = \max (t_1, t_2, \ldots, t_n) + \mu \]

• Split/merge overhead \( \mu \) is constant

• \( n \) small requests \( \approx n \) sampling (i.i.d.) of \( n \) IO Servers

• \( t_1, \ldots, t_n \approx n \) i.i.d. statistical variables

• Focus study on one (denoted by \( t \)) among \( t_1, \ldots, t_n \)
  – \( t \): continuously distributed variable on \( (0, +\infty) \)
Applying Order Statistics to Estimate $T$

$$T = \text{maximum } (t_1, \ldots, t_n) + \mu$$

$t$ : continuously distributed variable on $(0, +\infty)$

$F_t(x)$ : distribution function of $t$

$f_t(x) = F_t'(x)$ : density function of $t$

- **Step 1**: Compute density function $f_{Y_i}(y)$ with $F_t(x)$ and $f_t(x)$
  - $Y_i$: the $i$-th largest value $(t_1, t_2, \ldots, t_n)$
  - $f_{Y_i}(y) = F(y)^{n-i}(1-F(y))^{n-i} f_t(y) \frac{n!}{[(i-1)!(n-i)!]}$

- **Step 2**: Compute response time $T = Y_n$
Problem Statement

• What is the distribution function $F(t)$ for the response time of each small file request?
  – Existing researches assume
    • Uniform Distribution
    • Normal Distribution
  – Are these assumptions true?
  – If not, are there other distributions fitting better?
Our Method

• Sample the response time of two production storage systems

• Analyze statistical properties of response time
Response Time Sampling Environments

• Hopper and Edison at NERSC\(^1\)
  – 153\(K\) and 130\(K\) CPU cores, 1.28 PF and 2.39PF
  – 5000 registered users
  – 300 online active users on Edison
  – I/O Intensive jobs use Lustre

• Lustre file system
  – Cache on client and I/O server
  – Network latency
  – 1 ~ 143 OSTes

\(^1\)National Energy Research Scientific Computing Center
https://www.nersc.gov/
Sampling Method

• One job sampling one OST
  – A job ≈ A small file request
  – Measure time of reading and writing separately
  – Test different reading/writing sizes
    • 12 different sizes: 512KB, 1MB, 2MB, …, 1024MB
  – Match request size and striping size
Sampling Method

• Measure response time on computing node
  – network, disk, cache

• Cache Consideration
  – No Cache
    • clear cache by accessing memory sized data before sampling
    • call fsync() after write
  – Cache
    • High frequently sampling

Diagram:
- Computing Node /w Lustre Client
- Network Router
- Cache
- Lustre OST
# Sampling Results Statistics Overview

<table>
<thead>
<tr>
<th></th>
<th>Start Time</th>
<th>End Time</th>
<th>Days</th>
<th># of Sampling</th>
<th># of OSTs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edison-No-Cache</td>
<td>08/13/2014</td>
<td>09/17/2014</td>
<td>35</td>
<td>14,977</td>
<td>12</td>
</tr>
<tr>
<td>Edison-Cache</td>
<td>02/20/2015</td>
<td>02/20/2015</td>
<td>1</td>
<td>927,691</td>
<td>12</td>
</tr>
<tr>
<td>Hopper-No-Cache</td>
<td>10/01/2014</td>
<td>01/13/2015</td>
<td>104</td>
<td>13,868</td>
<td>12</td>
</tr>
<tr>
<td>Hopper-Cache</td>
<td>02/20/2015</td>
<td>02/20/2015</td>
<td>1</td>
<td>1,581,364</td>
<td>12</td>
</tr>
<tr>
<td><strong>Summary</strong></td>
<td></td>
<td></td>
<td>141</td>
<td><strong>2,537,900</strong></td>
<td><strong>48</strong></td>
</tr>
</tbody>
</table>
Variability of Raw Response Time for Edison and Hopper, Cache and No-Cache

(a) Edison-NoCache

Request Size 64MB

Response Time (sec)

Time (Aug 13 ~ Sep 17)

(b) Edison-Cache

Request Size 64MB

Response Time (sec)

Time (Feb 20, 2015 ~ Feb 20, 2015)

(c) Hopper-NoCache

Request Size 16MB

Response Time (sec)

Time (Oct 1, 2014 ~ Jan 13, 2015)

(d) Hopper-Cache

Request Size 16MB

Response Time (sec)

Time (Feb 20, 2015 ~ Feb 20, 2015)
Ill-fit of Uniform or Normal Distribution

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Uniform</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kurtosis</td>
<td>- 1.2</td>
<td>3</td>
</tr>
<tr>
<td>Skewness</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Ill-fit of Uniform, Normal, and Other Single Distribution Functions

- A single peak
- Nonsymmetrical
- Tail is real long

Histogram

Characters of Histogram:

- Single distribution functions
- Power Law
- Weibull
- Log Normal
- Exponential
- Gamma
- Normal
- Cauchy
- Uniform

Response Time (sec.)

0 5 10 15 20

0 0.2 0.4 0.6 0.8 1.0 1.2 1.4

Read (Stripe Size: 64MB)

don’t fit very well!
Exploring New Distributions

- Partition response time into Head and Tail
- Find the pivot
  - minimizing KS (Kolmogorov-Smirnov) distances

- Normal
- Cauchy
- Power Law
- Weibull
- Exponential
- Log Normal
- Gamma
Fitting Results

- Edison–NoCache, Read Response Time, 64MB

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head Group</td>
<td>Normal &gt; Cauchy</td>
</tr>
<tr>
<td>Tail Group</td>
<td>Power Law &gt; Log Normal &gt; Exponential &gt; Weibull &gt; Gamma</td>
</tr>
</tbody>
</table>
Fitting Results

- Edison–NoCache, Write Response Time, 64MB

<table>
<thead>
<tr>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head Group</td>
</tr>
<tr>
<td>Normal &gt; Cauchy</td>
</tr>
<tr>
<td>Tail Group</td>
</tr>
<tr>
<td>Power Law &gt; Weibull &gt; Exponential &gt; Log Normal &gt; Gamma</td>
</tr>
</tbody>
</table>
Percentage of Head group and Tail group

- 85% in Head group (i.e., small response time)
- 15% in Tail group (i.e., long response time)
What is Wrong with Using Normal or Uniform?

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Long Response Time (Rare Event)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform Distribution</td>
<td>All equal</td>
</tr>
<tr>
<td>Normal Distribution</td>
<td>2.5%</td>
</tr>
<tr>
<td>Real Storage Systems (Edison and Hopper)</td>
<td>15%</td>
</tr>
</tbody>
</table>

- **Uniform Distribution**:
  - All equal

- **Normal Distribution**:
  - 2.5%

- **Real Storage Systems (Edison and Hopper)**:
  - 15%

![Graph showing probability distribution and response times](image-url)
Summary

• Distribution function of response time of storage system is essential in estimating I/O performance
• We collected 2,537,900 response time sampling from 48 OSTes of 2 petascale storage systems across 141 days
• We found that single Normal or single Power law does not fit the response time
• We found that “Normal + Power law” fits response time better
• Future work
  – sample other storage systems
  – build accurate performance model
  – apply model to applications
Acknowledgments

- Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy (Program manager: Lucy Nowell), support for the SDS project under contract number DE-AC02-05CH11231

- National Energy Research Scientific Computing Center
Heavy-tailed Distribution of Parallel I/O System Response Time

Bin Dong, Surendra Byna, and Kesheng Wu
Scientific Data Management group
Lawrence Berkeley National Laboratory, Berkeley

Thanks, Questions?

➢ other questions, please email to: dbin@lbl.gov