Lustre* Client I/O Parallelization

Dmitry Eremin
LUG17

* Some names and brands may be claimed as the property of others.
Agenda

• What issues are we solving?

• Current I/O performance

• I/O data flows and Flame Graphs

• Ideas of Improvements

• Results
What issues are we solving?

• Single-threaded applications cannot utilize performance benefits of multiple I/O operations in Lustre* if a single core is slow

• The Network bandwidth cannot be saturated because of
  • Intel® OPA fabric uses a host CPU for packets processing
  • High overhead of single thread execution
  • Slow memory transfer operations

• Blocking the userspace I/O operation
  • To do Read Ahead

Number of cores in client machines is continually increasing, yet single-threaded I/O is still common.

A solution other than adding more compute nodes is needed.

* Some names and brands may be claimed as the property of others.
Hardware

Intel® Xeon® CPU E5-2697 v4 @ 2.30GHz
- L2 cache: 256K, L3 cache: 46080K
- CPUs: 72, Cores: 36, Threads per core: 2
- Intel® OPA HFI Silicon 100 Series [discrete]

Intel® Xeon Phi™ CPU 7250 @ 1.40GHz
- L2 cache: 1024K
- CPUs: 272, Cores: 68, Threads per core: 4
- Intel® OPA HFI Silicon 100 Series [discrete]
Intel® Xeon® E5-2697 IOR results
current version of Lustre*

![Graph showing MB/s performance with data points for BDW write and BDW read]

<table>
<thead>
<tr>
<th>Data Type</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BDW write</strong></td>
<td>1407.32</td>
<td>2524.64</td>
<td>4519.91</td>
<td>6518.61</td>
<td>6324.85</td>
<td>6112.59</td>
<td>5457.38</td>
<td>5845.68</td>
<td>5780.83</td>
</tr>
<tr>
<td><strong>BDW read</strong></td>
<td>811.99</td>
<td>1552.78</td>
<td>2800.55</td>
<td>3795.03</td>
<td>4643.87</td>
<td>4614.23</td>
<td>4668.15</td>
<td>5250.87</td>
<td>5300.38</td>
</tr>
</tbody>
</table>

* Some names and brands may be claimed as the property of others.

Source: Intel measured or estimated as of May 2017. Please see configuration details on slide 4 for configuration details.
Intel® Xeon Phi™ 7250 IOR results
current version of Lustre*

Source: Intel measured or estimated as of May 2017. Please see configuration details on slide 4 for configuration details.

* Some names and brands may be claimed as the property of others.
Write operation data flow

- For each PAGE_SIZE of user data:
  - Allocate new page in page cache (cl_page_alloc(), grab_cache_page())
  - Copy data from user space to page cache (copy_user_enhanced_fast_string())
  - Submit pages to RPC queue (vvp_io_write_commit())
  - Wake up ptlrpcd thread and wait for completion
Flame Graph of Write operation

- Page cache allocation is \( \frac{1}{2} \) of all write time
- Submit pages to RPC queue
- Copy user data

Source: Intel measured or estimated as of May 2017. Please see configuration details on slide 4 for configuration details.
Read operation data flow

- Read Ahead (`ll_readahead()`) for each page:
  - Allocate new page in page cache (`cl_page_alloc()`, `grab_cache_page()`)
  - Submit page to RPC queue for read (`cl_io_submit_rw()`)
  - Wake up `ptlrpcd` thread and wait for completion

- Copy data from page cache to user space (`copy_user_enhanced_fast_string()`)

![Diagram of data flow]
Flame Graph of Read operation

- Read Ahead is \(\approx \frac{1}{2}\) of all read time
- Page cache allocation is \(\approx \frac{3}{4}\) of Read Ahead
- Submit pages to RPC queue for read
- Copy user data

Source: Intel measured or estimated as of May 2017. Please see configuration details on slide 4 for configuration details.
Ideas of Improvements

• Split time consuming operations into several threads
  • Page cache allocation in parallel
  • Submit pages into several RPC queues in parallel
  • Copy user data in parallel**

• Move Read Ahead into asynchronous threads
  • Resume main read thread as soon as possible
  • Avoid cost of unused reads (false read ahead)

• Utilize more ptrlrpcd threads
  • Process network packets in parallel**

• Multiple network connection
  • Use multiple IB/OPA endpoints between nodes

** Very important for less performant cores.
Introduce new Parallel Tasks API

- Farm tasks out to be done in parallel on multiple CPUs
- Use system thread pool (don’t create new thread pool)
- Provide ordered or disordered task completion
- Submit for execution in Round Robin fashion

https://jira.hpdd.intel.com/browse/LU-8964
Multiple IB/OPA Endpoints

- Lustre* use a single network channel between nodes
  - Intel® OPA fabric use a host CPU for packets processing, so on machine with slow cores it cannot utilize full network bandwidth

- Optimization of IB/OPA LND driver
  - create multiple endpoints and balance the traffic over them

- https://jira.hpdd.intel.com/browse/LU-8943

* Some names and brands may be claimed as the property of others.
Flame Graph of Read with async RA

- Copy user data is \(\sim \frac{1}{2}\) of all read time
- Initial Read Ahead

Source: Intel measured or estimated as of May 2017. Please see configuration details on slide 4 for configuration details.

https://jira.hpdd.intel.com/browse/LU-8964
Flame Graph of Read Ahead thread

- Load the data in parallel (don’t serialize an user read)
- Can be stopped asynchronously at any time

Source: Intel measured or estimated as of May 2017. Please see configuration details on slide 4 for configuration details

https://jira.hpdd.intel.com/browse/LU-8964
Intel® Xeon® E5-2697 IOR results
Parallel I/O off, async Read Ahead

Source: Intel measured or estimated as of May 2017. Please see configuration details on slide 4 for configuration details.
Intel® Xeon® E5-2697 IOR results
Parallel I/O on, async Read Ahead

Source: Intel measured or estimated as of May 2017. Please see configuration details on slide 4 for configuration details.
Intel® Xeon® E5-2697 IOR results
Parallel I/O on, async Read Ahead, MQP

- BDW write
- BDW read
- BDW-mp1 write
- BDW-mp1 read

Source: Intel measured or estimated as of May 2017. Please see configuration details on slide 4 for configuration details.
## Intel® Xeon Phi™ 7250 IOR results

Parallel I/O off, async Read Ahead

![Chart showing IOR results](image)

<table>
<thead>
<tr>
<th>Parallel I/O</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>KNL write</strong></td>
<td>420.17</td>
<td>829.18</td>
<td>1619.05</td>
<td>2588.43</td>
<td>3478.02</td>
<td>3202.02</td>
<td>2663.39</td>
<td>2625.08</td>
<td>2131.31</td>
</tr>
<tr>
<td><strong>KNL read</strong></td>
<td>351.04</td>
<td>663.89</td>
<td>1313.74</td>
<td>2419.18</td>
<td>3278.37</td>
<td>3201.24</td>
<td>2984.83</td>
<td>2488.51</td>
<td>2271.53</td>
</tr>
<tr>
<td><strong>KNL-p0 write</strong></td>
<td>405.06</td>
<td>814.89</td>
<td>1638.32</td>
<td>2656.96</td>
<td>3758.21</td>
<td>3241.44</td>
<td>2586.71</td>
<td>2609.21</td>
<td>1859.98</td>
</tr>
<tr>
<td><strong>KNL-p0 read</strong></td>
<td>947.38</td>
<td>1816.44</td>
<td>3536.47</td>
<td>3200.26</td>
<td>3549</td>
<td>3391.45</td>
<td>3815.71</td>
<td>2679.95</td>
<td>2761.81</td>
</tr>
</tbody>
</table>

Source: Intel measured or estimated as of May 2017. Please see configuration details on slide 4 for configuration details.
Intel® Xeon Phi™ 7250 IOR results
Parallel I/O on, async Read Ahead

Source: Intel measured or estimated as of May 2017. Please see configuration details on slide 4 for configuration details.
## Intel® Xeon Phi™ 7250 IOR results

Parallel I/O on, async Read Ahead, MQP

![Graph showing parallel I/O results](image)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNL write</td>
<td>420.17</td>
<td>829.18</td>
<td>1619.05</td>
<td>2588.43</td>
<td>3478.02</td>
<td>3202.02</td>
<td>2663.39</td>
<td>2625.08</td>
<td>2131.31</td>
</tr>
<tr>
<td>KNL read</td>
<td>351.04</td>
<td>663.89</td>
<td>1313.74</td>
<td>2419.18</td>
<td>3278.37</td>
<td>3201.24</td>
<td>2984.83</td>
<td>2488.51</td>
<td>2271.53</td>
</tr>
<tr>
<td>KNL-mp1 write</td>
<td>1188.43</td>
<td>2085.11</td>
<td>3128.45</td>
<td>3794.76</td>
<td>4170.03</td>
<td>2932.48</td>
<td>2605.06</td>
<td>2776.42</td>
<td>2264.78</td>
</tr>
<tr>
<td>KNL-mp1 read</td>
<td>1111.12</td>
<td>2031.99</td>
<td>3114.42</td>
<td>3496.88</td>
<td>3195.69</td>
<td>3831.24</td>
<td>3511.97</td>
<td>3110.61</td>
<td>2782.79</td>
</tr>
</tbody>
</table>

Source: Intel measured or estimated as of May 2017. Please see configuration details on slide 4 for configuration details.
Hybrid Coordinate Ocean Model (HYCOM) application

Testing a real world application - see [https://hycom.org](https://hycom.org)

Current version of Lustre*

<table>
<thead>
<tr>
<th>Library</th>
<th>Calls</th>
<th>Time</th>
<th>Time/call</th>
</tr>
</thead>
<tbody>
<tr>
<td>zaio**</td>
<td>6</td>
<td>0.39432</td>
<td>0.06572032</td>
</tr>
<tr>
<td>zaiord</td>
<td>99</td>
<td>2.24527</td>
<td>0.02267950</td>
</tr>
<tr>
<td>zaiowr</td>
<td>726</td>
<td>81.19858</td>
<td>0.11184378</td>
</tr>
<tr>
<td>total</td>
<td>1</td>
<td>5203.09187</td>
<td>5203.09187198</td>
</tr>
</tbody>
</table>

Parallel I/O on, async Read Ahead, MQP

<table>
<thead>
<tr>
<th>Library</th>
<th>Calls</th>
<th>Time</th>
<th>Time/call</th>
</tr>
</thead>
<tbody>
<tr>
<td>zaio**</td>
<td>6</td>
<td>0.31822</td>
<td>0.05303649</td>
</tr>
<tr>
<td>zaiord</td>
<td>99</td>
<td>3.48087</td>
<td>0.03516028</td>
</tr>
<tr>
<td>zaiowr</td>
<td>726</td>
<td>65.78485</td>
<td>0.09061274</td>
</tr>
<tr>
<td>total</td>
<td>1</td>
<td>5149.86002</td>
<td>5149.86001992</td>
</tr>
</tbody>
</table>

* Some names and brands may be claimed as the property of others.
Summary

• Using parallel I/O and multiple IB/OPA endpoints Lustre* now can utilize performance benefits of
  • Multi-cores systems even for single-threaded applications
  • Multiple I/O operations in Lustre even if a single core is less performant
  • Intel® OPA fabric which uses a host CPU for packets processing
• Using asynchronous Read Ahead Lustre now
  • Doesn’t block the userspace I/O operation to do Read Ahead
  • For some workloads up to 2x speed-up in reads/writes has been observed
• Write work expected to be in 2.10.0; read work in 2.10.x maintenance release

* Some names and brands may be claimed as the property of others.
Notices and Disclaimers

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks

© Intel Corporation. Intel, Intel Inside, the Intel logo, Xeon, Intel Xeon Phi, Intel Xeon Phi logos and Xeon logos are trademarks of Intel Corporation or its subsidiaries in the United States and/or other countries.
Questions?