Erasure Code in Ceph
Loic Dachary @ Red Hat
Save Space

The OSD died! Did I lose data?

No! Let me recover it for you.
5 minutes role playing game
XOR

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
3 peta => 1.3 peta
Harder object mutations / recovery
Simple operations and tiering

Replicated

Erasure Coded
Promoted to replica on read

Replicated

Erasure Coded
Sam & David: internals
Janne & Andreas & Loic & Takeshi
erasure code
Released May 2014: Firefly
Why save space before shortage?
Reliability Model

- After an OSD is lost
- Recovery
- Backfilling a new OSD
April 2015 : Hammer
Repair $K=10$, $M=4$

Diagram:

- **Datacenter 1**: chunks 1 to 7
- **Datacenter 2**: chunks 8 to 14
Locally Recoverable Codes
LRC @ Red Hat

Datacenter 1
chunks 1 to 7

Datacenter 2
chunks 8 to 14

local chunk

local chunk
SHEC Takeshi @ Fujitsu

No need to be read

D1 D2 D3 D4 D5 D6 D7 D8 D9 D10

P1 P2 P3 P4 P5 P6

SHEC(10,6,3) a minimum union of calculation ranges including D6/D9
ISA plugin Yuan @ Intel

SIMD aka SSE2, SSE3, SSE4

Only for Intel processors

~50% Faster
Infernalis
Hitchhiker Rashmi @ U.C. Berkeley

\begin{itemize}
  \item Block 1: \( a_1 \) \quad \text{and} \quad \text{green rectangle:} \quad \text{a}_2
  \item Block 2: \( b_1 \) \quad \text{and} \quad b_2
  \item Block 3: \( a_1 + b_1 \) \quad \text{and} \quad \text{green rectangle:} \quad a_2 + b_2
  \item Block 4: \( 2b_1 - a_2 - 2b_2 \) \quad \text{and} \quad a_2 + 2b_2 + a_1
\end{itemize}

1 byte \quad 1 byte
ldachary@redhat.com

Artwork GPLv3+ Tartaruga Feliz

Thanks!