Efficiency at Scale

Sanjeev Kumar
Director of Engineering, Facebook

International Workshop on Rack-scale Computing, April 2014
## Agenda

1. Overview
2. Datacenter Architecture
3. Case Study: Optimizing BLOB Storage system
4. Questions
Facebook Stats

- 1.15 billion users [6/2013]
- ~700 million people use Facebook daily
- 350+ million photos added per day [1/2013]
- 240+ billion photos
- 4.5 billion likes, posts and comments per day [5/2013]
A large and growing server footprint
Services Provided by Facebook
Salient Points

- Efficiency matters
- Complex Software Stack
  - 1000+ specialized services to run
    - A few large services + Long tail
- Custom hardware: cost of designing, validating, fixing
- Number of machines a service needs can change quickly

Many sources of complexity
Simplify as much as possible
<table>
<thead>
<tr>
<th></th>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Overview</td>
</tr>
<tr>
<td>2</td>
<td>Datacenter Architecture</td>
</tr>
<tr>
<td>3</td>
<td>Case Study: Optimizing BLOB Storage system</td>
</tr>
<tr>
<td>4</td>
<td>Questions</td>
</tr>
</tbody>
</table>
Service Cluster

Back-End Cluster

Front-End Cluster

Web
250 racks

Cache (~144TB)

Ads
30 racks

Multifeed
9 racks

Other small services

Service Cluster

Search Photos Msg Others

Back-End Cluster

UDB ADS-DB Tao Leader
Infrastructure Redundancy

Regional Datacenter 1

Regional Datacenter 2

Regional Datacenter 3

Regional Datacenter 4
<table>
<thead>
<tr>
<th>Standard Systems</th>
<th>Type I</th>
<th>Type II</th>
<th>Type III</th>
<th>Type IV</th>
<th>Type V</th>
<th>Type VI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
<td>High 2 x EN2670</td>
<td>Low 1 x 6128HE (AMD)</td>
<td>Medium 2 x X5650</td>
<td>Medium 2 x X5650</td>
<td>Low 1 x L5630</td>
<td>High 2 x EN2660</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>Low 16GB</td>
<td>High 144GB</td>
<td>High 144GB</td>
<td>Medium 48GB</td>
<td>Low 18GB</td>
<td>High 144GB</td>
</tr>
<tr>
<td><strong>Disk</strong></td>
<td>Low 250GB</td>
<td>Low 250GB</td>
<td>High IOP 6 x 600GB SAS +2x1.3TB Flash</td>
<td>High 12 x 3TB SATA</td>
<td>High 12 x 3TB SATA</td>
<td>Medium 1TB SATA</td>
</tr>
<tr>
<td><strong>Services</strong></td>
<td>Web, Chat, Ads</td>
<td>Memcache, Ads</td>
<td>Database</td>
<td>Hadoop</td>
<td>Photos, Video</td>
<td>Multifeed, Search</td>
</tr>
<tr>
<td>Standard Systems</td>
<td>Type I</td>
<td>Type II</td>
<td>Type III</td>
<td>Type IV</td>
<td>Type V</td>
<td>Type VI</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>---------</td>
<td>----------</td>
<td>---------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>CPU</td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Memory</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Disk</td>
<td>Low</td>
<td>Low</td>
<td>High IOPs</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>Services</td>
<td>Web, Chat, Ads</td>
<td>Memcache, Ads</td>
<td>Database</td>
<td>Hadoop</td>
<td>Photos, Video</td>
<td>Multifeed, Search</td>
</tr>
</tbody>
</table>
## Server Generations

<table>
<thead>
<tr>
<th>Web Servers</th>
<th>2008</th>
<th>2009</th>
<th>2010</th>
<th>2011</th>
<th>2012</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rack Composition</td>
<td>L5420 (SC)</td>
<td>L5520 (NHM)</td>
<td>L5639 (WSM)</td>
<td>X5650 (XWSM)</td>
<td>EN2670 (SND)</td>
</tr>
<tr>
<td>Cores / Speed</td>
<td>8 real cores 2.50 GHz</td>
<td>16 logical CPUs (HT) 2.27 GHz</td>
<td>24 logical CPUs (HT) 2.13 GHz</td>
<td>24 logical CPUs (HT) 2.67 GHz</td>
<td>32 Logical CPUs (HT) 2.33GHz</td>
</tr>
<tr>
<td>RCUs</td>
<td>0.6</td>
<td>1</td>
<td>1.4</td>
<td>1.75</td>
<td>2.41</td>
</tr>
</tbody>
</table>
Web v1

Density: 2 Motherboards

Web v2
Web v1

CPU: Increased Performance
## Agenda

1. Overview
2. Datacenter Architecture
3. Case Study: Optimizing BLOB Storage system
4. Questions
### Storage Systems

<table>
<thead>
<tr>
<th></th>
<th>Total Size</th>
<th>Storage Technology</th>
<th>Bottlenecks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social Graph</td>
<td>Single-digit petabytes</td>
<td>MySQL &amp; Alternatives</td>
<td>Random read IOPS</td>
</tr>
<tr>
<td>Messages &amp; Time Series Data</td>
<td>10s of petabytes</td>
<td>HBase and HDFS</td>
<td>Write IOPS &amp; Storage capacity</td>
</tr>
<tr>
<td>Photos/Videos/BLOBs</td>
<td>100s of petabytes</td>
<td>Haystack</td>
<td>Storage capacity</td>
</tr>
<tr>
<td>Data Warehouse</td>
<td>100s of petabytes</td>
<td>Hive, HDFS, and Hadoop</td>
<td>Storage capacity</td>
</tr>
<tr>
<td>Cold Storage</td>
<td>Exabytes**</td>
<td>Custom</td>
<td>Storage Capacity</td>
</tr>
</tbody>
</table>
BLOB Storage

- Storage for Photos, Videos, Attachments, etc.

- Evolved over many generations
  - Constrained resource shifts and needs to be optimized for
    - Generation 1: Time to Market
    - Generation 2 & 3: Optimize the I/O request rate (Cost)
    - Generation 4: Optimize for Storage Efficiency (Cost)
Generation 1: Commercial Filers

- New Photos Product
- First build it the easy way
  - Commercial Storage Tier + HTTP server
  - Each Photo is stored as a separate file
- Quickly up and running
  - Reliably Store and Serve Photos
- **But:** Inefficient
  - Limited by IO rate and not storage density
  - Average 10 IOs to serve each photo
  - Wasted IO to traverse the directory structure
Effective but inefficient

- Disks are slow: 100 reads per disk per second
  - 1 photo read $\Rightarrow$ 10 disk reads
- Each disk can serve 10 photos per second

A copy of each photo

A copy of each photo

A copy of each photo

A copy of each photo
Generation 2: Gen 1 Optimized

- Optimization Example:
  - Cache NFS handles to reduce wasted IO operations
- Reduce the number of IO operations per photo by 3X
- **But:**
  - **Still expensive:** High end storage boxes
  - **Still inefficient:** Still IO bound and wasting IOs

NFS Storage Optimized
Generation 3: Haystack [OSDI’10]

- Custom Solution
  - Commodity Storage Hardware
  - Optimized for 1 IO operation per request
    - File system on top of a file system
    - Compact Index in memory
    - Metadata and data laid out contiguously
- Efficient from IO perspective
- **But:**
  - Problem has changed now

Single Disk IO to read/write a photo
Photo lifecycle

Relevance vs. Time

Warm
20% of total traffic is to the photos with ages less than 10 hours (these photos account for 0.06% of total photos)

34% of total traffic is to the photos with ages less than 24 hours (these photos account for 0.14% of total photos)

58% of total traffic is to the photos with ages less than 10 days (these photos account for 1.4% of total photos)

68% of total traffic is to the photos with ages less than 30 days (these photos account for 3.9% of total photos)

82% of total traffic is to the photos with ages less than 60 days (these photos account for 7.7% of total photos)

95% of total traffic is to the photos with ages less than 1 year (these photos account for 46% of total photos)

97% of total traffic is to the photos with ages less than 2 years (these photos account for 74% of total photos)
Generation 4: Tiered Storage

- Different storage solutions for hot, warm, cold photos

- **Hot** ➔ Haystack
- **Warm** ➔ F4 Storage System (+ Cold Storage)
Agenda

1. Overview
2. Datacenter Architecture
3. Case Study: Optimizing BLOB Storage system
4. Questions