Evaluation of Different Caching Strategies for YouTube – Multimedia Content

Abschlussvortrag zur Bachelor-Thesis
von
Elias Tatros
16.07.2012

Betreuer: Alexander Klein, Heiko Niedermayer
Outline

I. Introduction
   - Motivation
   - Goals & Contribution

II. Caching Framework Design & Architecture
   - Multi-layer Caching Infrastructures
   - Caching Scenarios
   - Caching Strategies
   - Tools
   - Modeled Nodes & Communication
   - Data Set

III. Simulation Results
   - Evaluation Scenario A
   - Evaluation Scenario B
   - Conclusion & Future Work
Motivation

- Internet video traffic growing at high rate*
  - Global Internet video traffic already surpassed global p2p traffic in 2010
  - 2012 Internet video traffic will account for over 50% of consumer internet traffic (86% by 2016)
  - Video on demand traffic will triple by 2015

- YouTube
  - Is currently the most popular video sharing application
  - Represents a significant amount of global internet traffic
  - One of the main reasons for increased HTTP traffic**

* Data taken from Cisco Visual Networking Index, May 2012.
Motivation

- Consequences of video traffic growth
  - ISPs need to keep pace with traffic growth, expand and/or adapt network infrastructure
  - Quality of Experience becomes a decision factor in provider choice
Motivation - Caching as a possible solution
Goals & Contribution

- Analysis, implementation and evaluation of popular caching strategies for multimedia video content

- Development and evaluation of chunk-wise caching strategies for multimedia video content
  - Chunks: 64 KB blocks of video data

- Analysis, implementation and evaluation of hierarchical caching infrastructures
Evaluation of Different Caching Strategies for Multimedia Content
Multi-layer Caching Infrastructures (Scenarios)

- Multi-layer caching infrastructures provide:
  - distribution of traffic and request load
  - savings in backhaul traffic
  - improved response times for locally cached content

- Benefits of multi-layer infrastructure over flat caching scenarios:
  - Reduced processing load on caches and video server
  - Reduced traffic load on popular routes (links)
  - Saved backhaul traffic
  - Faster response times for locally cached content
  - reduced network congestion can result in improved Quality of Experience
Scenario A: Simple two Layer Scenario

Layer 1 Cache:
- Number: 1
- Size: 1.0% - 20% of total unique requested data
- Focus on caching outdated popular content and cross-referenced content (content that is popular in both groups)

Layer 0 Caches:
- Number: 2
- Size: 0.5% - 10% of total unique requested data
- Focus on caching content popular in local network
Scenario B: Advanced two Layer Scenario

- Number of layer 0 local caches doubled
- Enables introduction of correlated groups
- Investigate influence of request correlation between client groups on layer 1 cache
Caching Strategies

- Caching Algorithms – Tasks and Problems:
  - **Storage:**
    Decide if incoming data should be stored
  - **Eviction:**
    Decide which content to evict in order to store new data

- Important Factors for Cache Replacement in YT Scenario:
  - User request behaviour
  - Global video popularity
  - Local video popularity
  - Popularity of individual video chunks

- Influential Factors for Video & Chunk Popularity:
  - Recency, Frequency, Size
Caching Strategies

- Chunk-wise Caching Strategies:
  - LRU Chunk (Recency based):
    - Eviction: remove least recently requested chunk
    - Efficient insertion and removal
  - LRU Request (Recency/Frequency based):
    - Eviction: remove least recently requested chunk
    - Parameter $x$ specifies minimum number of times a chunk needs to be requested before it is stored
    - Requires twice the space of LRU Chunk due to tracking of chunk frequencies

- Full Video Caching Strategies:
  - Video LRU (Recency based):
    - Storage and removal of complete videos
    - Eviction: remove least recently requested video
  - Video Size (Size based):
    - Storage and removal of complete videos
    - Eviction: remove video with largest size
Tools

- **OPNET Modeler**
  - Discrete event simulation
  - Analyze simulated networks
  - Collect statistics
  - Many integrated protocols and devices
  - Hierarchical modeling using Nodes, Modules and Processes

- **MATLAB**
  - Analysis and Evaluation of collected Statistics
Client Nodes:
- Represent group of users / devices
- Each Client is assigned a local cache
- Task: Send video / chunk requests to local cache according to request schedule

Cache Node Modules
Server Node:
- Connected to top cache node in hierarchy
- Task: Respond to content requests
Data set

- Data set collected during a measurement of YouTube traffic within the Munich Scientific Research Network (MWN)
  - Measurement period: 3 months
  - Users in Network: 120,000+
  - Video Requests observed: 7,000,000+

- Question: How to assign observed chunk requests to Client Groups?
Scenario A (simple 2-layer): Request Distribution

(a) Offset Request Distribution

(b) High Correlation within Groups
Scenario A (simple 2-layer): Global Hitrates

<table>
<thead>
<tr>
<th>Statistic</th>
<th>LRU Chunk</th>
<th>LRU Request</th>
<th>Full Vid LRU</th>
<th>Full Vid Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Hit Rate (100GB/200GB)</td>
<td>43.56%</td>
<td>41.65%</td>
<td>37.38%</td>
<td>24.07%</td>
</tr>
<tr>
<td>Avg. Layer 0 Hit Rate (100GB)</td>
<td>41.73%</td>
<td>40.45%</td>
<td>36.28%</td>
<td>22.23%</td>
</tr>
<tr>
<td>Avg. Layer 1 Hit Rate (200GB)</td>
<td>3.36%</td>
<td>2.29%</td>
<td>1.80%</td>
<td>2.50%</td>
</tr>
</tbody>
</table>

(a) Client Group 1

(b) Client Group 2
Scenario A (simple 2-layer): Download from Server

<table>
<thead>
<tr>
<th>Statistic</th>
<th>LRU Chunk</th>
<th>LRU Request</th>
<th>Full Vid LRU</th>
<th>Full Vid Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Downloaded from Server</td>
<td>1.36 TB</td>
<td>1.36 TB</td>
<td>1.62 TB</td>
<td>2.08 TB</td>
</tr>
<tr>
<td>Avg. Data Rate L1 to L0</td>
<td>16.74 Mb/s</td>
<td>16.95 Mb/s</td>
<td>19.39 Mb/s</td>
<td>24.97 Mb/s</td>
</tr>
<tr>
<td>Avg. Data Rate Server to L1</td>
<td>16.01 Mb/s</td>
<td>15.97 Mb/s</td>
<td>18.99 Mb/s</td>
<td>24.49 Mb/s</td>
</tr>
</tbody>
</table>
Scenario B (adv. 2-layer): Request Distribution

Scenario 2 distribution method (variation of alternating distribution)

Request Data File (same as scenario 1)

First Half of Request File
Alternating Distribution between Group 1 and 2

Second Half of Request File
Alternating Distribution between Group 3 and 4

High correlation between G1, G2

Low correlation

High correlation between G3, G4

Client G1
Client G2
Client G1
Client G2

Client G3
Client G4
Client G3
Client G4
### Scenario B (adv. 2-layer): Global Hitrates

<table>
<thead>
<tr>
<th>Statistic</th>
<th>LRU Chunk</th>
<th>LRU Request</th>
<th>Full Vid LRU</th>
<th>Full Vid Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Hit Rate (100GB/200GB)</td>
<td>42.78%</td>
<td>35.04%</td>
<td>37.71%</td>
<td>25.43%</td>
</tr>
</tbody>
</table>

#### Graphs

- **Hit Rate in %**
  - **Cache Size in GB**
  - **(a) Client Group 1**
  - **(c) Client Group 3**

- Graphs show hit rates for different cache sizes and client groups using various caching strategies.
### Scenario B (adv. 2-layer): Average Data Rates

<table>
<thead>
<tr>
<th>Statistic</th>
<th>LRU Chunk</th>
<th>LRU Request</th>
<th>Full Vid LRU</th>
<th>Full Vid Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Data Rate L1 to L0</td>
<td>18.97 Mb/s</td>
<td>19.21 Mb/s</td>
<td>21.44 Mb/s</td>
<td>25.64 Mb/s</td>
</tr>
<tr>
<td>Avg. Data Rate Server to L1</td>
<td>15.99 Mb/s</td>
<td>17.5 Mb/s</td>
<td>18.88 Mb/s</td>
<td>24.16 Mb/s</td>
</tr>
</tbody>
</table>

(a) Layer 1 to Layer 0

(b) Server to Layer 1
Scenario B (adv. 2-layer): Filling of L1 Cache

### Simulation Parameters

<table>
<thead>
<tr>
<th>Property</th>
<th>Client Group 1</th>
<th>Client Group 2</th>
<th>Client Group 3</th>
<th>Client Group 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total num. of requests</td>
<td>417,379</td>
<td>417,379</td>
<td>501,601</td>
<td>501,600</td>
</tr>
<tr>
<td>Total num. of chunks</td>
<td>8,884,083</td>
<td>8,886,416</td>
<td>11,315,185</td>
<td>11,279,376</td>
</tr>
<tr>
<td>Size of req. content</td>
<td>529.53 GB</td>
<td>529.67 GB</td>
<td>674.44 GB</td>
<td>672.30 GB</td>
</tr>
</tbody>
</table>

(a) LRU Request

(b) LRU Chunk
Evaluation of Different Caching Strategies for Multimedia Content

Scenario (A / B / Alt) Comparison: 200 GB L1 Cache

- **Slow Popularity Reaction** → **Slow Growth**
- **Fast Popularity Reaction** → **Quick Stability**
- **More Request Correlation** → **Higher L1 Hit Rates**

![Graph showing hit rate over time for different LRU Chunk scenarios](image)

(a) LRU Chunk
Scenario B (adv. 2-layer): Download from Server

<table>
<thead>
<tr>
<th>Statistic</th>
<th>LRU Chunk</th>
<th>LRU Request</th>
<th>Full Vid LRU</th>
<th>Full Vid Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Downloaded from Server</td>
<td>1.36 TB</td>
<td>1.49 TB</td>
<td>1.61 TB</td>
<td>2.05 TB</td>
</tr>
</tbody>
</table>

![Graph showing the evaluation of different caching strategies for multimedia content.](image-url)
Conclusion & Future Work

- **Chunk-wise** caching strategies for user generated video content offer significant advantages over traditional caching strategies:
  - Higher hit rates at all layers
  - Lower data rates between layers (e.g. Client ↔ Cache, Cache ↔ Server)
  - Less data downloaded from server
  - Less load on caches and server
  - Fewer evictions can result in positive long term effects on health and performance of cache hardware

Future Work

- Run simulation for larger time frames (> 180.6 h) and cache sizes
- Evaluate new cache miss strategy: Request of content from higher layer caches to lower layers in case of cache miss
- Explore more caching strategies and adjust them for chunk-wise caching
- Evaluate use of different caching strategies at each layer
- Investigate the role of correlation between client networks more closely
Thank you for your time and attention.

Questions?
## Simulation Parameters Scenario A / B

<table>
<thead>
<tr>
<th>Property</th>
<th>Client Group 1</th>
<th>Client Group 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total num. of requests</td>
<td>834,758</td>
<td>1,003,201</td>
</tr>
<tr>
<td>Total num. of chunks</td>
<td>17,770,499</td>
<td>22,594,561</td>
</tr>
<tr>
<td>Size of req. content</td>
<td>1059.20 GB</td>
<td>1346.74 GB</td>
</tr>
<tr>
<td>Unique chunks requested</td>
<td>9,042,153</td>
<td>10,994,804</td>
</tr>
<tr>
<td>Size of unique content</td>
<td>538.95 GB</td>
<td>355.34 GB</td>
</tr>
<tr>
<td>Individual videos</td>
<td>358,556</td>
<td>444,627</td>
</tr>
<tr>
<td>Theo. max. hit rates</td>
<td>47.12%</td>
<td>51.34%</td>
</tr>
<tr>
<td>Time of first request</td>
<td>0 s</td>
<td>1 s</td>
</tr>
<tr>
<td>Time of last request</td>
<td>180.6 h</td>
<td>180.6 h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Client Group 1</th>
<th>Client Group 2</th>
<th>Client Group 3</th>
<th>Client Group 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total num. of requests</td>
<td>417,379</td>
<td>417,379</td>
<td>501,601</td>
<td>501,600</td>
</tr>
<tr>
<td>Total num. of chunks</td>
<td>8,884,083</td>
<td>8,886,416</td>
<td>11,315,185</td>
<td>11,279,376</td>
</tr>
<tr>
<td>Size of req. content</td>
<td>529.53 GB</td>
<td>529.67 GB</td>
<td>674.44 GB</td>
<td>672.30 GB</td>
</tr>
<tr>
<td>Unique chunks req.</td>
<td>5,376,913</td>
<td>5,354,232</td>
<td>6,628,817</td>
<td>6,602,644</td>
</tr>
<tr>
<td>Size of unique content</td>
<td>320.49 GB</td>
<td>319.14 GB</td>
<td>395.10 GB</td>
<td>393.55 GB</td>
</tr>
<tr>
<td>Individual videos</td>
<td>225,296</td>
<td>224,876</td>
<td>278,022</td>
<td>277,662</td>
</tr>
<tr>
<td>Theo. max hit rate</td>
<td>39.48%</td>
<td>39.75%</td>
<td>41.42%</td>
<td>41.46%</td>
</tr>
<tr>
<td>Time of first request</td>
<td>0 s</td>
<td>0 s</td>
<td>1 s</td>
<td>3 s</td>
</tr>
<tr>
<td>Time of last request</td>
<td>180.6 h</td>
<td>180.6 h</td>
<td>180.6 h</td>
<td>180.6 h</td>
</tr>
</tbody>
</table>
## Simulation Results Scenario A / B

### Results Scenario A

<table>
<thead>
<tr>
<th>Statistic</th>
<th>LRU Chunk</th>
<th>LRU Request</th>
<th>Full Vid LRU</th>
<th>Full Vid Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Hit Rate (100GB/200GB)</td>
<td>43.56%</td>
<td>41.65%</td>
<td>37.38%</td>
<td>24.07%</td>
</tr>
<tr>
<td>Avg. Layer 0 Hit Rate (100GB)</td>
<td>41.73%</td>
<td>40.45%</td>
<td>36.28%</td>
<td>22.23%</td>
</tr>
<tr>
<td>Avg. Layer 1 Hit Rate (200GB)</td>
<td>3.36%</td>
<td>2.29%</td>
<td>1.80%</td>
<td>2.50%</td>
</tr>
<tr>
<td>Total Data Evicted L1 (200GB)</td>
<td>1.13 TB</td>
<td>0</td>
<td>12.0 TB</td>
<td>16.9 TB</td>
</tr>
<tr>
<td>Data Downloaded from Server</td>
<td>1.36 TB</td>
<td>1.36 TB</td>
<td>1.62 TB</td>
<td>2.08 TB</td>
</tr>
<tr>
<td>Avg. Data Rate L1 to L0</td>
<td>16.74 Mb/s</td>
<td>16.95 Mb/s</td>
<td>19.39 Mb/s</td>
<td>24.97 Mb/s</td>
</tr>
<tr>
<td>Avg. Data Rate Server to L1</td>
<td>16.01 Mb/s</td>
<td>15.97 Mb/s</td>
<td>18.99 Mb/s</td>
<td>24.49 Mb/s</td>
</tr>
</tbody>
</table>

### Results Scenario B

<table>
<thead>
<tr>
<th>Statistic</th>
<th>LRU Chunk</th>
<th>LRU Request</th>
<th>Full Vid LRU</th>
<th>Full Vid Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Hit Rate (100GB/200GB)</td>
<td>42.78%</td>
<td>35.04%</td>
<td>37.71%</td>
<td>25.43%</td>
</tr>
<tr>
<td>Avg. Layer 0 Hit Rate (100GB)</td>
<td>33.30%</td>
<td>32.10%</td>
<td>29.65%</td>
<td>19.97%</td>
</tr>
<tr>
<td>Avg. Layer 1 Hit Rate (200GB)</td>
<td>14.77%</td>
<td>4.56%</td>
<td>11.48%</td>
<td>6.95%</td>
</tr>
<tr>
<td>Total Data Evicted L1 (200GB)</td>
<td>1.12 TB</td>
<td>66.6 GB</td>
<td>11.81 TB</td>
<td>16.63 TB</td>
</tr>
<tr>
<td>Data Downloaded from Server</td>
<td>1.36 TB</td>
<td>1.49 TB</td>
<td>1.61 TB</td>
<td>2.05 TB</td>
</tr>
<tr>
<td>Avg. Data Rate L1 to L0</td>
<td>18.97 Mb/s</td>
<td>19.21 Mb/s</td>
<td>21.44 Mb/s</td>
<td>25.64 Mb/s</td>
</tr>
<tr>
<td>Avg. Data Rate Server to L1</td>
<td>15.99 Mb/s</td>
<td>17.5 Mb/s</td>
<td>18.88 Mb/s</td>
<td>24.16 Mb/s</td>
</tr>
</tbody>
</table>
Scenario A (simple 2-layer): Local Hitrates

Local Hit Rate
Layer 1
Scenario A
Scenario A (simple 2-layer): Filling of L0 Cache

Simulation Parameters

<table>
<thead>
<tr>
<th>Property</th>
<th>Client Group 1</th>
<th>Client Group 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total num. of requests</td>
<td>834,758</td>
<td>1,003,201</td>
</tr>
<tr>
<td>Total num. of chunks</td>
<td>17,770,499</td>
<td>22,594,561</td>
</tr>
<tr>
<td>Size of req. content</td>
<td>1059.20 GB</td>
<td>1346.74 GB</td>
</tr>
</tbody>
</table>

(a) LRU Chunk Strategy
(b) Full Video Caching
## Scenario A (simple 2-layer): Average Data Rates

<table>
<thead>
<tr>
<th>Statistic</th>
<th>LRU Chunk</th>
<th>LRU Request</th>
<th>Full Vid LRU</th>
<th>Full Vid Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Hit Rate (100GB/200GB)</td>
<td>43.56%</td>
<td>41.65%</td>
<td>37.38%</td>
<td>24.07%</td>
</tr>
<tr>
<td>Avg. Layer 0 Hit Rate (100GB)</td>
<td>41.73%</td>
<td>40.45%</td>
<td>36.28%</td>
<td>22.23%</td>
</tr>
<tr>
<td>Avg. Layer 1 Hit Rate (200GB)</td>
<td>3.36%</td>
<td>2.29%</td>
<td>1.80%</td>
<td>2.50%</td>
</tr>
<tr>
<td>Total Data Evicted L1 (200GB)</td>
<td>1.13 TB</td>
<td>0</td>
<td>12.0 TB</td>
<td>16.9 TB</td>
</tr>
<tr>
<td>Data Downloaded from Server</td>
<td>1.36 TB</td>
<td>1.36 TB</td>
<td>1.62 TB</td>
<td>2.08 TB</td>
</tr>
<tr>
<td>Avg. Data Rate L1 to L0</td>
<td>16.74 Mb/s</td>
<td>16.95 Mb/s</td>
<td>19.39 Mb/s</td>
<td>24.97 Mb/s</td>
</tr>
<tr>
<td>Avg. Data Rate Server to L1</td>
<td>16.01 Mb/s</td>
<td>15.97 Mb/s</td>
<td>18.99 Mb/s</td>
<td>24.49 Mb/s</td>
</tr>
</tbody>
</table>

![Graphs showing data rates](image)

(a) Layer 1 to Layer 0

(b) Server to Layer 1
Scenario A (simple 2-layer): Evictions L0 Caches
Scenario A (simple 2-layer): Evictions L1 Cache
Scenario B (adv. 2-layer): Evictions L0 Caches

(a) Client Group 1, first correlation group, Layer 0  
(b) Client Group 3, second correlation group, Layer 0
Scenario A (simple 2-layer): Data Rate over Time

(a) 200 Gigabytes Cache Size

(b) 10 Gigabytes Cache Size
Alternating Request Distribution

Request Data File

- Request / Line 1
- Request / Line 2
- Request / Line 3
- Request / Line 4
  
Client Group 1  Client Group 2

(a) Alternating Request Distribution

Balanced correlation of requests (alternating requests)

Request Data File

- Request / Line 1
- Request / Line 2
- Request / Line 3
- Request / Line 4
- Request / Line 5
- Request / Line 6
  
Balanced correlation

Client G 1  Client G 2  Client G 1  Client G 2

(b) Balanced Correlation in and between Groups

Evaluation of Different Caching Strategies for Multimedia Content
Alternating Blocks Distribution

High inter-group correlation (alternating blocks)

Request Data File

- Request Block 1
  Lines 0 to blocklength

- Request Block 2
  Lines blocklength to 2*blocklength

- Request Block 2
  Lines 2*blocklength to 3*blocklength

- Request Block 2
  Lines 3*blocklength to 4*blocklength

Client G 1

Client G 2

High inter-group correlation

gap

High inter-group correlation
Related Work


- A Chunk-based Caching Algorithm for Streaming Video, Dohy Hong, Danny De Vleeschauwer, Francois Baccelli, Alcatel-Lucent

- Watch Global, Cache Local: YouTube Network Traffic at a Campus Network – Measurements and Implications, Michael Zink, Kyoungwon Suh, Yu Gu, Jim Kurose, University of Massachusetts

- Application Flow Control in Youtube Video Streams, Shane Alcock, Richard Nelson, University of Waikato