On Content Delivery to Heterogeneous Devices

Sharayu Moharir
School of Technology and Computer Science, TIFR

Joint work with Rahul Vaze
Motivation
Content Delivery Networks

1. Large amount of content
2. Device heterogeneity
Motivation

Device Heterogeneity

End-users
Different operating systems, screen sizes, bit-rate requirements, codec support etc.

New Challenge: Delivering content in multiple formats
New Resource: Computational power - transcoders
Content Delivery Network

Tasks

✦ What to store on the front-end servers?
✦ How to use transcoding resources?
Setting
Front-end Servers

✦ Contents, large
✦ Storage - Vanishing fraction of all contents \((o(n), \text{e.g., } \sqrt{n})\)
✦ Service - Limited requests served concurrently
✦ Non-uniform storage and service capabilities

Front-end server
Limited storage and service capability, transcoding resources
## Setting

### Cost of serving requests

<table>
<thead>
<tr>
<th>Step Description</th>
<th>Cost Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Serve using front-end server</td>
<td>( C_{\text{min}} )</td>
</tr>
<tr>
<td>2. Fetch and serve</td>
<td>( C_{\text{min}} + C_{\text{Fetch}} )</td>
</tr>
<tr>
<td>3. Transcode and serve</td>
<td>( C_{\text{min}} + C_{\text{Transcode}} )</td>
</tr>
<tr>
<td>4. Serve using back-end server</td>
<td>( C_{\text{max}} )</td>
</tr>
</tbody>
</table>

No queues

\[
\begin{align*}
C_{\text{min}} & < C_{\text{max}} \\
C_{\text{Fetch}}, C_{\text{Transcode}} & > 0
\end{align*}
\]

**Goal:** Optimize content replication on front-end servers to minimize the cost of serving requests.
Setting
Content & Format Popularity

- Heavy tailed content popularity*
  Zipf distribution
  - Requests for $C_i \sim \text{Poisson}(\lambda_i)$
  - $\lambda_i \propto i^\beta$, $\beta > 0$

- Format popularity
  Non-uniform & content dependent

- Supportable load

$Liu$ et al., *Measurement and analysis of an internet streaming service to mobile devices*, *IEEE Transactions on Parallel and Distributed Systems*. 
Goal: Optimize content replication on front-end servers to minimize the cost of serving request.
Candidate Strategies

I. Transcode on the fly* (ToF):

Store master format, transcode on demand to serve requests
e.g., VUCLIP - mobile VoD service, dynamic adaptive transcoding

II. Lazy Transcoding and Re-transcoding** (LTR):

Store transcoded versions, delete obsolete formats periodically

*U.S. Patent No. 8,869,218
**U.S. Patent No. 8,782,285
**Definition:** \( \Gamma_{\text{ALG}} = \text{Cost per request} \)

**Theorem:** \( \lim_{n \to \infty} E[\Gamma_{\text{DIST-LTR}}] = C_{\text{min}} \)

<table>
<thead>
<tr>
<th>Routing</th>
<th>Random routing - Probability request routed to server ( j ) ( \propto ) service capacity of server ( j )</th>
</tr>
</thead>
</table>
| Content Replication | On a request arrival for \( C_{i,f} \):
  
  Case 1 - Server busy: serve using back-end server
  Case 2 - \( C_{i,f} \) available: serve request
  Case 3 - \( C_{i,f} \) not available: fetch or transcode, replace content(s) not being used with \( C_{i,f} \) |
Assume that the front-end server can serve $M(n)$ parallel requests.

Recall: Content popularity $\sim$ Zipf($\beta$), $\beta > 1$

Example:

$M(n) = \sqrt{n}$, $\beta = 1$

$1 - n^{-1/2}$

$(M(n))^{1/\beta} = n^{1/4}$

$\text{Threshold}(M(n), \beta)$

At least 1 active request at all times

Always stored locally, requests served at low cost

Possibly served at higher cost

Decreasing popularity
Transcode on the Fly

Definitions

\( \Gamma_{ALG} = \text{Cost per request} \)

\( q = \text{Expected fraction of requests for the master format} \)

\[
\lim_{n \to \infty} E[\Gamma_{ToF}] \geq c_{\text{min}} + \min\{C_{\text{Transcode}}, C_{\text{max}} - c_{\text{min}}\} (1-q)
\]

✦ Routing using global information
✦ Co-ordination across front-end servers
✦ Use knowledge of content popularity
✦ Static/adaptive content replication

Request for other formats - transcode/serve using back-end server
**DIST-LTR**

| Routing | Random routing - Probability request routed to server $j$  
|         | $\propto$ service capacity of server $j$ |
| Content Replication | On a request arrival for $C_{i,f}$:  
|         | Case 1 - Server busy: serve using back-end server  
|         | Case 2 - $C_{i,f}$ available: serve request  
|         | Case 3 - $C_{i,f}$ not available: fetch or transcode, replace content(s) not being used with $C_{i,f}$  

- Randomly chosen content (LTR-RANDOM)  
- Least recently used content (LTR-LRU)  
- Least frequently used content (LTR-LFU)
Simulations

Cost vs Zipf Parameter

$c_{\text{min}} = 1$
$c_{\text{max}} = 3$
$c_{\text{Fetch}} = 1$
$c_{\text{Transcode}} = 1$
$\beta = 1.2$
Simulations

Cost vs Front-end Storage

- LTR-RANDOM
- ToF (Lower Bound)

Cost per Request

Cost vs Front-end Storage

- $c_{min} = 1$
- $c_{max} = 3$
- $C_{Fetch} = 1$
- $C_{Transcode} = 1$
- $\beta = 1.2$
- $F = 4$
Simulations

Cost vs Number of Formats

- $c_{min} = 1$
- $c_{max} = 3$
- $C_{Fetch} = 1$
- $C_{Transcode} = 1$
- $\beta = 1.2$
Simulations

Cost vs Zipf Parameter

![Graph showing cost per request vs Zipf parameter]

- LTR-RANDOM
- ToF (Lower Bound)

- $c_{\text{min}} = 1$
- $c_{\text{max}} = 3$
- $c_{\text{Fetch}} = 1$
- $c_{\text{Transcode}} = 1$
- $F = 4$
Netflix Data

Content Popularity

Relative Content Popularity

Content

Slope = -0.1
Slope = -2
**Simulations**

**Netflix Content Popularity**

![Graph showing Cost per Request vs Front-end Storage](image)

- Cost per Request along the y-axis.
- Front-end Storage along the x-axis.

- Line with points indicating LTR-RANDOM simulations.
- Line with circles indicating ToF (Lower Bound) simulations.

### Parameters:

- $c_{min} = 1$
- $c_{max} = 3$
- $c_{Fetch} = 1$
- $c_{Transcode} = 1$

### Values:

- $F = 4$
## Related Work

### Device Heterogeneity
- Measurement and analysis of an internet streaming service to mobile devices
  Liu, Li, Guo, Shen, Chen & Lan, *IEEE Transactions on Parallel and Distributed Systems*

- Joint online transcoding & geo-distributed delivery for dynamic adaptive streaming

### Large content catalogs
- Serving content with unknown demand: the high-dimensional regime
  S.M., Ghaderi, Sanghavi & Shakkottai, *ACM Sigmetrics* 2014

- Adaptive replication in distributed content delivery networks
  Leconte, Lelarge & Massoulie, *ITC* 2015

- Bipartite graph structures for efficient balancing of heterogeneous loads
  Leconte, Lelarge & Massoulie, *Sigmetrics* 2012

- Queueing system topologies with limited flexibility
  Tsitsiklis & Xu, *Sigmetrics* 2013
Conclusions

Task - Content replication for content delivery in multiple formats

Candidate Approaches -
- Transcode on the fly: Store content in one high-quality master format
- DIST-LTR: Stores multiple formats of the same content

Results -
- The transcode on the fly approach is strictly suboptimal
- DIST-LTR is asymptotically optimal, even without coordination
Thanks