Coded Caching for Files with Distinct File Sizes

Jinbei Zhang†, Xiaojun Lin‡, Chih-Chun Wang‡,
Xinbing Wang†
†Shanghai Jiao Tong University
‡Purdue University
The Importance of Caching

Data traffic continues to grow at significant rates
A major fraction (60-80%) of traffic will be generated by multimedia content, such as video
Caching is important for reducing backhaul requirement in serving large volumes of content that multiple users are interested in
An Example of Coded Caching [Maddah-Ali and Niesen ’14]

K=3 users
Cache size M=1

An Example of Coded Caching [Maddah-Ali and Niesen ’14]

N=3 files: \( A = (A_1, A_2, A_3) \)
\( B = (B_1, B_2, B_3) \)
\( C = (C_1, C_2, C_3) \)

K=3 users
Cache size M=1

An Example of Coded Caching [Maddah-Ali and Niesen ’14]

N=3 files: $A = (A_1, A_2, A_3)$
$B = (B_1, B_2, B_3)$
$C = (C_1, C_2, C_3)$

Back-haul Requirement: K=3 users
Cache size M=1

• Uncoded Caching

User 1 wants $A$
User 2 wants $B$
User 3 wants $C$

An Example of Coded Caching [Maddah-Ali and Niesen ’14]

N=3 files: \( A = (A_1, A_2, A_3) \)
\( B = (B_1, B_2, B_3) \)
\( C = (C_1, C_2, C_3) \)

Back-hual Requirement:
• Uncoded Caching

K=3 users
Cache size M=1

User 1 wants \( A \)
User 2 wants \( B \)
User 3 wants \( C \)

An Example of Coded Caching [Maddah-Ali and Niesen ’14]

N=3 files: 

- $A = (A_1, A_2, A_3)$
- $B = (B_1, B_2, B_3)$
- $C = (C_1, C_2, C_3)$

Back-hual Requirement:

- Uncoded Caching

\[ K \cdot \left(1 - \frac{M}{N}\right) = 2 \]

K=3 users
Cache size M=1

An Example of Coded Caching [Maddah-Ali and Niesen ’14]

N=3 files:  
\[ A = (A_1, A_2, A_3) \]
\[ B = (B_1, B_2, B_3) \]
\[ C = (C_1, C_2, C_3) \]

Backhaul Requirement:
- Uncoded Caching  
  \[ K \cdot \left( 1 - \frac{M}{N} \right) = 2 \]
- Coded Caching [1]

An Example of Coded Caching [Maddah-Ali and Niesen ’14]

N=3 files: \( A = (A_1, A_2, A_3) \)
\[ B = (B_1, B_2, B_3) \]
\[ C = (C_1, C_2, C_3) \]

Backhaul Requirement:
- Uncoded Caching
\[ K \cdot \left(1 - \frac{M}{N}\right) = 2 \]
- Coded Caching [1]

User 1 wants \( A \)
User 2 wants \( B \)
User 3 wants \( C \)

An Example of Coded Caching [Maddah-Ali and Niesen ’14]

N=3 files: \( A = (A_1, A_2, A_3) \)
\( B = (B_1, B_2, B_3) \)
\( C = (C_1, C_2, C_3) \)

Back-haul Requirement:

- Uncoded Caching
  \( K \cdot \left(1 - \frac{M}{N}\right) = 2 \)
- Coded Caching [1]

User 1 wants \( A \)
User 2 wants \( B \)
User 3 wants \( C \)

An Example of Coded Caching [Maddah-Ali and Niesen ’14]

N=3 files: \( A = (A_1, A_2, A_3) \)
\( B = (B_1, B_2, B_3) \)
\( C = (C_1, C_2, C_3) \)

Back-haul Requirement:
- Uncoded Caching
  \[ K \cdot \left( 1 - \frac{M}{N} \right) = 2 \]
- Coded Caching [1]

An Example of Coded Caching [Maddah-Ali and Niesen ’14]

N=3 files: $A = (A_1, A_2, A_3)$
$B = (B_1, B_2, B_3)$
$C = (C_1, C_2, C_3)$

Back-haul Requirement:
- Uncoded Caching
  $K \cdot \left(1 - \frac{M}{N}\right) = 2$
- Coded Caching [1]

An Example of Coded Caching [Maddah-Ali and Niesen ’14]

N=3 files: 
\[ A = (A_1, A_2, A_3) \]
\[ B = (B_1, B_2, B_3) \]
\[ C = (C_1, C_2, C_3) \]

Back-haul Requirement:
- Uncoded Caching
  \[ K \cdot \left(1 - \frac{M}{N}\right) = 2 \]
- Coded Caching [1]
  \[ K \cdot \left(1 - \frac{M}{N}\right) \cdot \frac{1}{1 + \frac{KM}{N}} = 1 \]

K=3 users
Cache size M=1

User 1 wants A
User 2 wants B
User 3 wants C

An Example of Coded Caching [Maddah-Ali and Niesen ’14]

N=3 files: \( A = (A_1, A_2, A_3) \)
\( B = (B_1, B_2, B_3) \)
\( C = (C_1, C_2, C_3) \)

Back-haul Requirement:

- Uncoded Caching
  \[ K \cdot \left( 1 - \frac{M}{N} \right) = 2 \]
- Coded Caching [1]
  \[ K \cdot \left( 1 - \frac{M}{N} \right) \cdot \frac{1}{1 + \frac{K M}{N}} = 1 \]

Recent Efforts

- [Maddah-Ali and Niesen ’14] shows that the worst-case transmission rate $K \cdot \left(1 - \frac{M}{N}\right) \cdot \frac{1}{1 + \frac{KM}{N}}$ is at most a constant factor (12x) away from the (information-theoretic) minimum possible

- Generalized to
  - Decentralized/probabilistic caching schemes [Ali and Niesen `14]
  - Hierarchical caching [Karamchandani et al `14]
  - Online caching [Pedarsani et al `13]
  - Expected Rate [Niesen and Ali `14, Ji et al `14, Hachem et al `14]

- These studies all assume a homogeneous setting where all files are of a same size
The Question

- In practice, file sizes are often quite different

- In *heterogeneous* settings:
  - Should *larger* files be cached more aggressively?
  - What is the fundamental *limit* of the performance of coded caching with distinct file sizes?
Our Contribution

- Heterogeneous *file-sizes*
  - (Roughly) *quadratically* more content is cached for larger files
  - We show *logarithmic-factor* bounds

- While the new achievable schemes are quite intuitive, the corresponding *lower bounds* are more involved and reveal useful insights
Network Model: Distinct File Sizes

- Server with a broadcast channel
- $K$ users: cache size $M$
- $N$ files: $\mathcal{F} = \{F_1, \ldots, F_N\}$
- Request pattern: $W_i = \{f_{i1}, \ldots, f_{iK}\}$, $f_{ik} \in \mathcal{F}$
  Rate for $W_i$ is $r(W_i)$
- **Worst-case rate:**
  $R = \max_i r(W_i)$
Network Model: Distinct File Sizes

- Server with a broadcast channel
- $K$ users: cache size $M$
- $N$ files: $\mathcal{F} = \{\mathcal{F}_1, \ldots, \mathcal{F}_N\}$
- Request pattern: $W_i = \{f_{i1}, \ldots, f_{iK}\}, f_{ik} \in \mathcal{F}$
  Rate for $W_i$ is $r(W_i)$
- **Worst-case rate:**
  $$R = \max_i r(W_i)$$
Network Model: Distinct File Sizes

- Server with a broadcast channel
- $K$ users: cache size $M$
- $N$ files: $\mathbb{F} = \{F_1, \ldots, F_N\}$
- Request pattern:
  $W_i = \{f_{i1}, \ldots, f_{iK}\}, f_{ik} \in \mathbb{F}$
  Rate for $W_i$ is $r(W_i)$
- **Worst-case rate:**
  \[ R = \max_i r(W_i) \]
Network Model: Distinct File Sizes

- Server with a broadcast channel
- $K$ users: cache size $M$
- $N$ files: $\mathbb{F} = \{F_1, \ldots, F_N\}$
- Request pattern:
  $W_i = \{f_{i1}, \ldots, f_{iK}\}, f_{ik} \in \mathbb{F}$
  Rate for $W_i$ is $r(W_i)$
- **Worst-case rate:**
  $R = \max_i r(W_i)$
Power-of-2 Simplification

All files
Power-of-2 Simplification

All files

Re-arrangement

$F_1$
Power-of-2 Simplification

\[ F_1, N'_1 \]
\[ F_2 = \frac{F_1}{2}, N'_2 \]
\[ F_T = \frac{F_1}{2^{T-1}}, N'_T \]

\[ F_1, N_1 \]
\[ F_2 = \frac{F_1}{2}, N_2 \]
\[ F_T = \frac{F_1}{2^{T-1}}, N_T \]

All files

Re-arrangement
Power-of-2 Simplification

\[ F_1, N_1' \]

\[ F_2 = \frac{F_1}{2}, N_2' \]

\[ F_T = \frac{F_1}{2^{T-1}}, N_T' \]

\[ F_1, N_1 \]

\[ F_2 = \frac{F_1}{2}, N_2 \]

\[ F_T = \frac{F_1}{2^{T-1}}, N_T \]
Power-of-2 Simplification

\[ F_1, N_1 \]

\[ F_2 = \frac{F_1}{2}, N_2 \]

\[ F_{\overline{T}} = \frac{F_1}{2^{T-1}}, N_2 \]

\[ F_T = \frac{F_1}{2^{T-1}}, N_T \]
Power-of-2 Simplification

\[ F_1, N_1 \]

\[ F_2 = \frac{F_1}{2}, N_2 \]

\[ F_{\overline{T}} = \frac{F_1}{2^{T-1}}, N_2 \]

\[ F_T = \frac{F_1}{2^{T-1}}, N_T \]

- \( \overline{T} = \min\{T, \log_2 K\} \)
  - Files of type \( l > \overline{T} \) can be virtually neglected
  - Their sizes \( \leq F_1/K \)
Power-of-2 Simplification

\[
F_1, N_1
\]

\[
F_2 = \frac{F_1}{2}, N_2
\]

\[
F_T = \frac{F_1}{2^{T-1}}, N_T
\]

\[
F_{\bar{T}} = \frac{F_1}{2^{\bar{T}-1}}, N_2
\]

- \( \bar{T} = \min\{T, \log_2 K\} \)
  - Files of type \( l > \bar{T} \) can be virtually neglected
  - Their sizes \( \leq F_1/K \)
Intuition Towards Quadratic Caching

• The worst-case rate with uniform file size of 1 [Maddah-Ali & Niesen `14] is given by

\[ K \cdot \left(1 - \frac{M}{N}\right) \cdot \frac{1}{1 + \frac{KM}{N}} \approx \frac{N}{M} - 1 \approx \frac{N}{M}, \text{ when } K \gg \frac{N}{M} \text{ and } M \ll N \]

• Each user caches every “bit” of each file with probability \( q = \frac{M}{N} \)

\[ \text{Worst-case rate} \approx \frac{N}{M} = \frac{1}{q} \]

• When the uniform file-size is \( F \), these numbers become:

\[ \text{Caching probability } q = \frac{M}{NF}, \]

\[ \text{Worst-case rate: } \]

\[ KF \cdot \left(1 - \frac{M}{NF}\right) \cdot \frac{1}{1 + \frac{KM}{NF}} \approx \frac{NF^2}{M} = \frac{F}{q}, \text{ when } q \ll 1 \text{ and } K \gg 1/q \]
Two Achievable Schemes (UB1 vs UB2)

Achievable Scheme 1 (UB1)

- All files are cached with an equal probability $q$
- *Linearly* more content is cached for larger files

- Cache constraint:
  $$q \sum_{l=1}^{\bar{T}} N_l F_l = M$$
  with
  $$\bar{T} = \min(T, \log_2 K)$$

- $R_{UB1} \geq O(\max_l \frac{F_l}{q})$
Two Achievable Schemes (UB1 vs UB2)

Achievable Scheme 1 (UB1)

- All files are cached with an equal probability $q$
- **Linearly** more content is cached for larger files
- Cache constraint:
  $$q \sum_{l=1}^{T} N_l F_l = M$$
  with
  $$\bar{T} = \min(T, \log_2 K)$$

$$R_{UB1} \geq O(\max_l \frac{F_l}{q})$$

$F_1$ suffers
## Two Achievable Schemes (UB1 vs UB2)

### Achievable Scheme 1 (UB1)

- All files are cached with an equal probability $q$
- **Linearly** more content is cached for larger files
- Cache constraint:
  \[ q \sum_{l=1}^{\bar{T}} N_l F_l = M \text{ with } \bar{T} = \min(T, \log_2 K) \]
  \[ R_{UB1} \geq O(\max_l \frac{F_l}{q}) \]
  
  \( F_1 \) suffers

### Achievable Scheme 2 (UB2)

- Let the caching probability be
  \[ q_l = F_l / c \]
- **Quadratically** more content is cached for larger files
- Cache constraint:
  \[ \sum_{l=1}^{\bar{T}} q_l N_l F_l = M \Rightarrow \sum_{l=1}^{\bar{T}} N_l F_l^2 = cM \]
  \[ R_{UB2} \leq \sum_{l=1}^{\bar{T}} \frac{F_l}{q_l} + K \cdot F_{\bar{T}+1} \]
Two Achievable Schemes (UB1 vs UB2)

Achievable Scheme 1 (UB1)

- All files are cached with an equal probability $q$
- **Linearly** more content is cached for larger files
- Cache constraint: $q \sum_{l=1}^{\bar{T}} N_l F_l = M$ with $\bar{T} = \min(T, \log_2 K)$

$$R_{UB1} \geq O\left(\max_l \frac{F_l}{q}\right)$$

$F_1$ suffers

Achievable Scheme 2 (UB2)

- Let the caching probability be $q_l = F_l/c$
- **Quadratically** more content is cached for larger files
- Cache constraint:

$$\sum_{l=1}^{\bar{T}} q_l N_l F_l = M$$

$$\implies \sum_{l=1}^{\bar{T}} N_l F_l^2 = cM$$

$$R_{UB2} \leq \sum_{l=1}^{\bar{T}} \frac{F_l}{q_l} + K \cdot F_{\bar{T}+1}$$

# of types * rate for each type

$F_1$ suffers
Two Achievable Schemes (UB1 vs UB2)

Achievable Scheme 1 (UB1)

\[ R_{UB1} \geq O\left(\frac{F_1 \sum_{l=1}^{\bar{T}} N_l F_l}{4M}\right) \]

Consider \( T = \log_2 K, N_{l+1} = 4N_l \)

Achievable Scheme 2 (UB2)

\[ R_{UB2} \leq (\bar{T} + 1) \frac{\sum_{l=1}^{\bar{T}} N_l F_l^2}{M} \]
Two Achievable Schemes (UB1 vs UB2)

Achievable Scheme 1 (UB1)

\[ R_{UB1} \geq O\left(\frac{F_1 \sum_{l=1}^{T} N_l F_l}{4M}\right) \]

Consider \( T = \log_2 K \), \( N_{l+1} = 4N_l \)

\[ R_{UB1} \geq K \frac{N_1 F_1^2}{8M} \]

Achievable Scheme 2 (UB2)

\[ R_{UB2} \leq (\bar{T} + 1) \frac{\sum_{l=1}^{\bar{T}} N_l F_l^2}{M} \]

\[ R_{UB2} \leq (\bar{T} + 1) \bar{T} \frac{N_1 F_1^2}{M} \]
Two Achievable Schemes (UB1 vs UB2)

Achievable Scheme 1 (UB1)

\[ R_{UB1} \geq O\left(\frac{F_1 \sum_{l=1}^{T} N_l F_l}{4M}\right) \]

- Consider \( T = \log_2 K, \) \( N_{l+1} = 4N_l \)

\[ R_{UB1} \geq K \frac{N_1 F_1^2}{8M} \]

Achievable Scheme 2 (UB2)

\[ R_{UB2} \leq (\bar{T} + 1) \frac{\sum_{l=1}^{T} N_l F_l^2}{M} \]

\[ R_{UB2} \leq (\bar{T} + 1) \bar{T} \frac{N_1 F_1^2}{M} \]

Critical to cache \textit{quadratically} more content for larger files!
In order to serve all request patterns, we must have

\[ \frac{2M}{F} R^*(F) + \frac{NF}{2M} M \geq NF \quad \Rightarrow \quad R^*(F) \geq \frac{NF^2}{4M} = \frac{F}{4q} \]
In order to serve all request patterns, we must have

\[
\left[ \frac{\sum_{l \in \Phi} N_l}{L} \right] \cdot R^* (\mathbb{F}) + L \cdot M \geq \sum_{l \in \Phi} N_l F_l
\]

Maximizing over \( L \) and \( \Phi \)

\[
R^* (\mathbb{F}) \geq \max_{\Phi} \frac{(\sum_{l \in \Phi} N_l F_l)^2}{4M \sum_{l \in \Phi} N_l}
\]

(LB1)
LB1 vs UB2

Lower Bound 1 (LB1)

- Consider $N_{l+1} = 4N_l$

$$R_{LB1} \leq \frac{N_1 F_1^2}{M}$$

Achievable Bound 2 (UB2)

$$R_{UB2} = (\bar{T} + 1) \frac{\sum_{l=1}^{\bar{T}} N_l F_l^2}{M}$$

(Number of types vs log(K))

(Number of types vs log(rate))
LB1 vs UB2

Lower Bound 1 (LB1)

- Consider \( N_{l+1} = 4N_l \)

\[ R_{LB1} \leq \frac{N_1F_1^2}{M} \]

Achievable Bound 2 (UB2)

\[ R_{UB2} = (\bar{T} + 1) \frac{\sum_{l=1}^{T} N_l F_l^2}{M} \]

LB1 fails to account for heterogeneous caching probabilities!
An Improved Lower Bound (LB2)

Proposition 1: Under

- Assumption 1: $\frac{2M}{F_1}$ and $\frac{N_l F_l}{2M}$ are integers for all $1 \leq l \leq \bar{T}$
- Assumption 2: $\sum_{l=1}^{\bar{T}} \frac{N_l F_l}{2M} \leq K$

We must have

$$R^* (F) \geq \sum_{l=1}^{\bar{T}} \frac{N_l F_l^2}{4M}$$

(LB2)

Compared with UB2:

$$R_{UB2} = (\bar{T} + 1) \frac{\sum_{l=1}^{\bar{T}} N_l F_l^2}{M}$$
Intuition for LB2 (Two types: $F_2 = F_1/2$)

$$R^*(\mathcal{F}) \geq \sum_{l=1}^{\bar{T}} \frac{N_l F_l^2}{4M} = \frac{N_1 F_1^2}{4M} + \frac{N_2 F_2^2}{4M} \quad \text{(LB2)}$$

$s_1 = \frac{N_1 F_1}{2M}$ users ($U_1$)

$s_2 = \frac{N_2 F_2}{2M}$ users ($U_2$)
Intuition for LB2 (Two types: $F_2 = F_1/2$)

$$R^* (\mathbb{F}) \geq \sum_{l=1}^{\tilde{T}} \frac{N_l F_l^2}{4M} = \frac{N_1 F_1^2}{4M} + \frac{N_2 F_2^2}{4M}$$

(LB2)

$$s_1 = \frac{N_1 F_1}{2M} \text{ users } (U_1)$$

$$s_2 = \frac{N_2 F_2}{2M} \text{ users } (U_2)$$

$$\frac{N_1}{s_1} = \frac{2M}{F_1} \text{ patterns}$$

$\cdots$  \hspace{1cm} $\cdots$

$N_1$ files of type 1 ($\mathbb{F}_1$)  \hspace{1cm} $\cdots$
Intuition for LB2 (Two types: $F_2 = F_1/2$)

\[
R^*(\mathbb{F}) \geq \sum_{l=1}^{\bar{T}} \frac{N_l F_l^2}{4M} = \frac{N_1 F_1^2}{4M} + \frac{N_2 F_2^2}{4M}
\]  

(LB2)

\[
s_1 = \frac{N_1 F_1}{2M} \quad \text{users (U_1)}
\]

\[
s_2 = \frac{N_2 F_2}{2M} \quad \text{users (U_2)}
\]

\[
\frac{N_1}{s_1} = \frac{2M}{F_1}
\]

patterns

\[
N_1 \quad \text{files of type 1 (\mathbb{F}_1)}
\]

\[
\cdots
\]

\[
R^*(\mathbb{F}) \geq \frac{N_1 F_1^2}{4M}
\]
Intuition for LB2 (Two types: $F_2 = F_1/2$)

$$R^*(\mathbb{F}) \geq \sum_{l=1}^{\bar{T}} \frac{N_l F_l^2}{4M} = \frac{N_1 F_1^2}{4M} + \frac{N_2 F_2^2}{4M}$$

(LB2)

- $s_1 = \frac{N_1 F_1}{2M}$ users ($U_1$)
- $\frac{N_1}{s_1} = \frac{2M}{F_1}$
- $N_1$ patterns
- $N_1$ files of type 1 ($\mathbb{F}_1$)

- $s_2 = \frac{N_2 F_2}{2M}$ users ($U_2$)
- $N_2$ files of type 2 ($\mathbb{F}_2$)
Intuition for LB2 (Two types: $F_2 = F_1/2$)

$$R^* (\mathbb{F}) \geq \sum_{l=1}^{\bar{T}} \frac{N_l F_l^2}{4M} = \frac{N_1 F_1^2}{4M} + \frac{N_2 F_2^2}{4M}$$

(LB2)

\[s_1 = \frac{N_1 F_1}{2M}\] users ($U_1$)

\[s_2 = \frac{N_2 F_2}{2M}\] users ($U_2$)

\[\frac{N_1}{s_1} = \frac{2M}{F_1}\] patterns

\[\frac{N_1}{s_1} = \frac{2M}{F_1}\] patterns

$N_1$ files of type 1 ($\mathbb{F}_1$)

$F_{21}$
Intuition for LB2 (Two types: $F_2 = F_1/2$)

$$R^* (\mathbb{F}) \geq \sum_{l=1}^{\tilde{T}} \frac{N_l F_l^2}{4M} = \frac{N_1 F_1^2}{4M} + \frac{N_2 F_2^2}{4M} \quad \text{(LB2)}$$

$$s_1 = \frac{N_1 F_1}{2M} \quad \text{users (U_1)}$$

$$s_2 = \frac{N_2 F_2}{2M} \quad \text{users (U_2)}$$

$$\frac{N_1}{s_1} = \frac{2M}{F_1} \quad \text{patterns}$$

$$\frac{N_1}{s_1} = \frac{2M}{F_1} \quad \text{patterns}$$
Intuition for LB2 (Two types: $F_2 = F_1/2$)

$$R^*(F) \geq \sum_{l=1}^{T} \frac{N_l F_l^2}{4M} = \frac{N_1 F_1^2}{4M} + \frac{N_2 F_2^2}{4M}$$

(LB2)

$$s_1 = \frac{N_1 F_1}{2M} \text{ users (U_1)}$$

$$s_2 = \frac{N_2 F_2}{2M} \text{ users (U_2)}$$
Proof: (Two types: $F_2 = F_1/2$)

- $s_1 = \frac{N_1 F_1}{2M}$, $s_2 = \frac{N_2 F_2}{2M}$
- $H(\mathcal{M}_1) = 0.5H(F_1) = \frac{N_1 F_1}{2}$
- $H(\mathcal{M}_2) = 0.5H(F_2) = \frac{N_2 F_2}{2}$

$$H(\mathcal{R}_{D_1}, \mathcal{M}_1) = H(\mathcal{R}_{D_1}, \mathcal{M}_1|F_1) + I(\mathcal{R}_{D_1}, \mathcal{M}_1; F_1)$$
Proof: (Two types: $F_2 = F_1/2$)

- $s_1 = \frac{N_1 F_1}{2M}$, $s_2 = \frac{N_2 F_2}{2M}$
- $H(M_1) = 0.5H(F_1) = \frac{N_1 F_1}{2}$
- $H(M_2) = 0.5H(F_2) = \frac{N_2 F_2}{2}$

\[
H(RD_1, M_1) = H(RD_1, M_1 | F_1) + I(RD_1, M_1; F_1) \\
0.5H(F_1) + \frac{1}{H(F_1)}
\]
Proof: (Two types: $F_2 = F_1/2$)

- $s_1 = \frac{N_1 F_1}{2M}$, $s_2 = \frac{N_2 F_2}{2M}$
- $H(M_1) = 0.5 H(F_1) = \frac{N_1 F_1}{2}$
- $H(M_2) = 0.5 H(F_2) = \frac{N_2 F_2}{2}$

\[
H(R_{D_1}, M_1) = H(R_{D_1}, M_1 | F_1) + I(R_{D_1}, M_1; F_1) \]
\[
0.5 H(F_1) \]
\[
\text{H} \]
\[
H(R_{D_1}) \geq H(R_{D_1}, M_1 | F_1) + 0.5 H(F_1)
\]
Proof: (Two types: $F_2 = F_1/2$)

- $s_1 = \frac{N_1 F_1}{2M}$, $s_2 = \frac{N_2 F_2}{2M}$
- $H(M_1) = 0.5H(F_1) = \frac{N_1 F_1}{2}$
- $H(M_2) = 0.5H(F_2) = \frac{N_2 F_2}{2}$

\[
H(R_{D_1}, M_1) = H(R_{D_1}, M_1 | F_1) + I(R_{D_1}, M_1; F_1) \\
0.5H(F_1) \\
H(F_1) \\
H(R_{D_1}) \geq H(R_{D_1}, M_1 | F_1) + 0.5H(F_1) \\
H(R_{D_2}) \geq H(R_{D_2}, M_1 | F_1) + 0.5H(F_1)
\]
Proof: (Two types: $F_2 = F_1/2$)

- $s_1 = \frac{N_1 F_1}{2M}$, $s_2 = \frac{N_2 F_2}{2M}$
- $H(M_1) = 0.5H(F_1) = \frac{N_1 F_1}{2}$
- $H(M_2) = 0.5H(F_2) = \frac{N_2 F_2}{2}$

\[
H(R_{D_1}, M_1) = H(R_{D_1}, M_1 | F_1) + I(R_{D_1}, M_1; F_1)
\]

\[
0.5H(F_1)
\]

\[
H(R_{D_1}) \geq H(R_{D_1}, M_1 | F_1) + 0.5H(F_1)
\]

\[
H(R_{D_2}) \geq H(R_{D_2}, M_1 | F_1) + 0.5H(F_1)
\]

$H(M_2)$

\[
H(R_{D_1}) + H(R_{D_2}) \geq H(F_2) - 0.5H(F_2) + H(F_1)
\]
Proof: \((T>=3)\)

\[ \frac{4N_1}{s_1} R^*(F) \geq 4I(\mathcal{R}_{D_1}; F_1) \geq 4 \cdot \frac{N_1 F_1}{2} \]

\[ + 2I(\mathcal{R}_{D_1} \cup \mathcal{R}_{D_2}; F_2 | F_1) \geq 2 \cdot \frac{N_2 F_2}{2} \]

\[ + I(\mathcal{R}_{D_1} \cup \mathcal{R}_{D_2} \cup \mathcal{R}_{D_3} \cup \mathcal{R}_{D_4}; F_3 | F_1, F_2) \geq \frac{N_3 F_3}{2} \]
Without Assumptions 1 & 2

- Intuition
  - Memory sharing argument
- Assume there exists a lower bound $R$

\[
M' = M - \sum_{l=1}^{T_1} N_l(F_l - R)
\]

- Group 1
  - $F_l \geq 2R$
  - Cache $F_l - R$ portion
- Group 2 (can be empty)
  - $F_l < 2R$, $F_l > 2M'$
  - Each user requests 1 file
- Group 3
  - $F_l \leq 2M'$
  - Using previous results
Comparisons

• With Assumptions 1 & 2

<table>
<thead>
<tr>
<th>Rate</th>
<th>LB2/LB1</th>
<th>Gain</th>
<th>UB1/UB2</th>
<th>Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 types</td>
<td>4/3.6</td>
<td>11%</td>
<td>15.3757/13.2523</td>
<td>16%</td>
</tr>
<tr>
<td>3 types</td>
<td>5/3.8462</td>
<td>30%</td>
<td>21.2593/16.2196</td>
<td>31%</td>
</tr>
</tbody>
</table>
Comparisons

• With Assumptions 1 & 2

<table>
<thead>
<tr>
<th>Rate</th>
<th>LB2/LB1</th>
<th>Gain</th>
<th>UB1/UB2</th>
<th>Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 types</td>
<td>4/3.6</td>
<td>11%</td>
<td>15.3757/13.2523</td>
<td>16%</td>
</tr>
<tr>
<td>3 types</td>
<td>5/3.8462</td>
<td>30%</td>
<td>21.2593/16.2196</td>
<td>31%</td>
</tr>
</tbody>
</table>

General Result (without Assumptions 1 and 2):

\[ R_{UB2} \leq (32\log_2 K + 22)R_{LB2} \]
Conclusion and Discussions

• Heterogeneous *file-sizes*
  – *Quadratically* more content is cached for larger files
  – We show *logarithmic-factor* gap

• While the new achievable schemes are quite intuitive, the corresponding *lower bounds* more involved and reveal useful insights

• Future Work:
  – A tighter lower-bound?
  – New techniques in the achievable scheme?
  – Remove the logarithmic factor
  – Sharper insights into the general lower bound
Thank you!