

Fundamental Limits of Caching

Urs Niesen
Jointly with Mohammad Maddah-Ali

Bell Labs, Alcatel-Lucent

Video on Demand

Video on demand is getting increasingly popular:

- Netflix streaming service
- Amazon Instant Video
- Hulu
- Verizon / Comcast on Demand
- ...

Video on Demand

Video on demand is getting increasingly popular:

- Netflix streaming service
- Amazon Instant Video
- Hulu
- Verizon / Comcast on Demand
- ...

⇒ Places significant stress on service provider's networks

Video on Demand

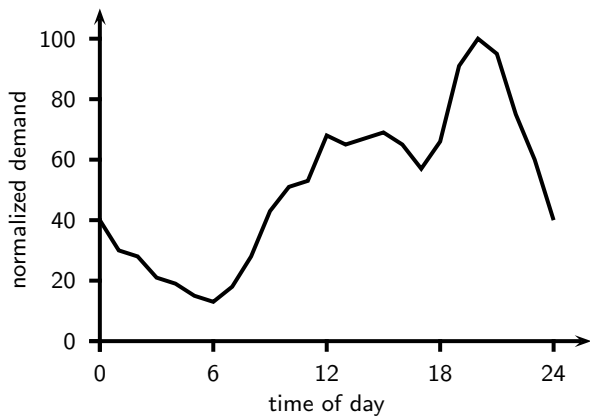
Video on demand is getting increasingly popular:

- Netflix streaming service
- Amazon Instant Video
- Hulu
- Verizon / Comcast on Demand
- ...

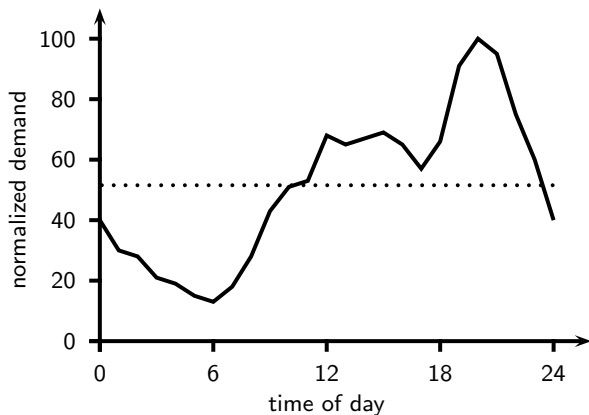
⇒ Places significant stress on service provider's networks

⇒ Caching (prefetching) can be used to mitigate this stress

Caching (Prefetching)

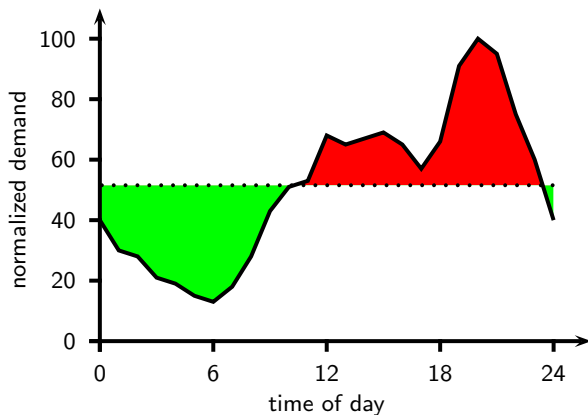


Caching (Prefetching)



- High temporal traffic variability

Caching (Prefetching)



- High temporal traffic variability
- Caching can help smooth traffic

The Role of Caching

Conventional beliefs about caching:

The Role of Caching

Conventional beliefs about caching:

- Caches useful to deliver content **locally**

The Role of Caching

Conventional beliefs about caching:

- Caches useful to deliver content **locally**
- **Local** cache size matters

The Role of Caching

Conventional beliefs about caching:

- Caches useful to deliver content **locally**
- **Local** cache size matters
- Statistically identical users \Rightarrow **identical** cache content

The Role of Caching

Conventional beliefs about caching:

- Caches useful to deliver content **locally**
- **Local** cache size matters
- Statistically identical users \Rightarrow **identical** cache content

Insights from this work:

The Role of Caching

Conventional beliefs about caching:

- ~~Caches useful to deliver content~~ locally
- Local cache size matters
- Statistically identical users \Rightarrow identical cache content

Insights from this work:

- The main gain in caching is global

The Role of Caching

Conventional beliefs about caching:

- ~~Caches useful to deliver content~~ **locally**
- ~~Local~~ cache size matters
- Statistically identical users \Rightarrow **identical** cache content

Insights from this work:

- The main gain in caching is **global**
- **Global** cache size matters

The Role of Caching

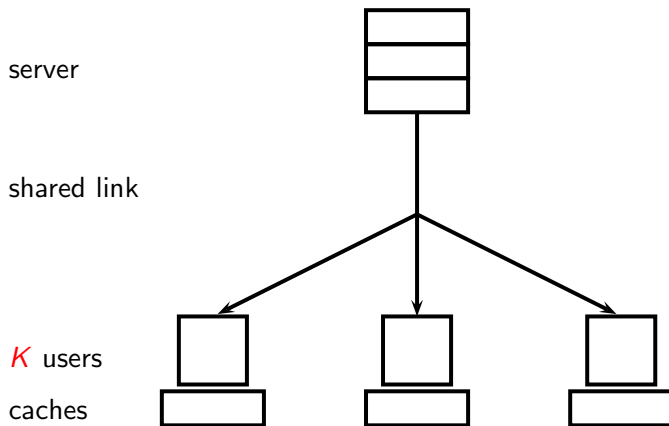
Conventional beliefs about caching:

- ~~Caches useful to deliver content~~ **locally**
- ~~Local~~ cache size matters
- ~~Statistically identical users~~ \Rightarrow **identical** cache content

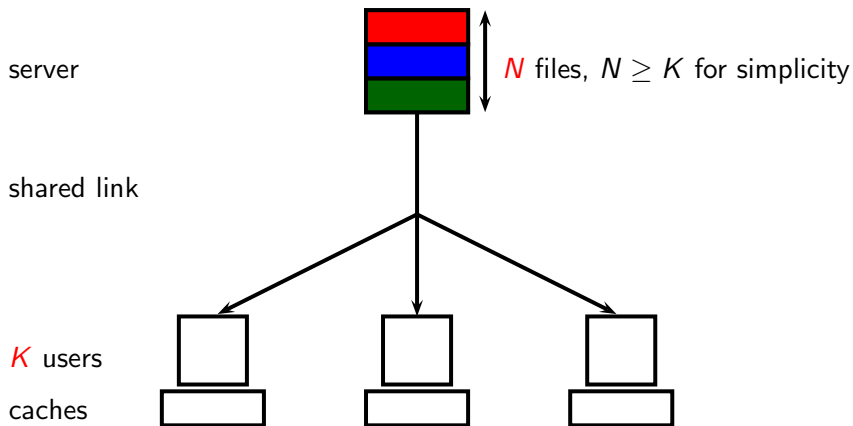
Insights from this work:

- The main gain in caching is **global**
- **Global** cache size matters
- ~~Statistically identical users~~ \Rightarrow **different** cache content

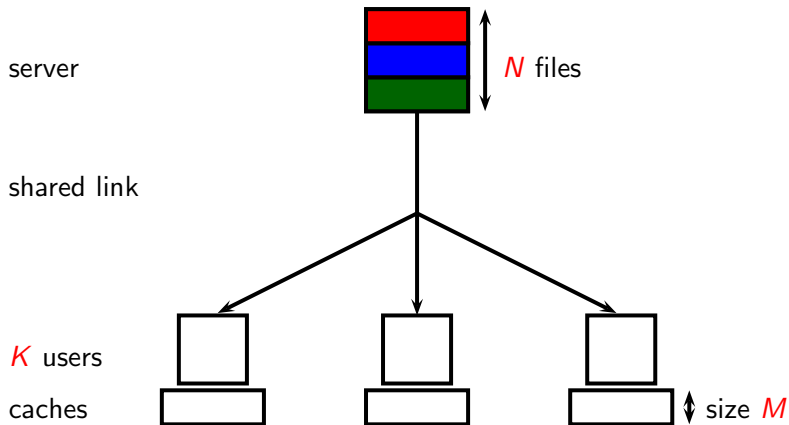
Problem Setting



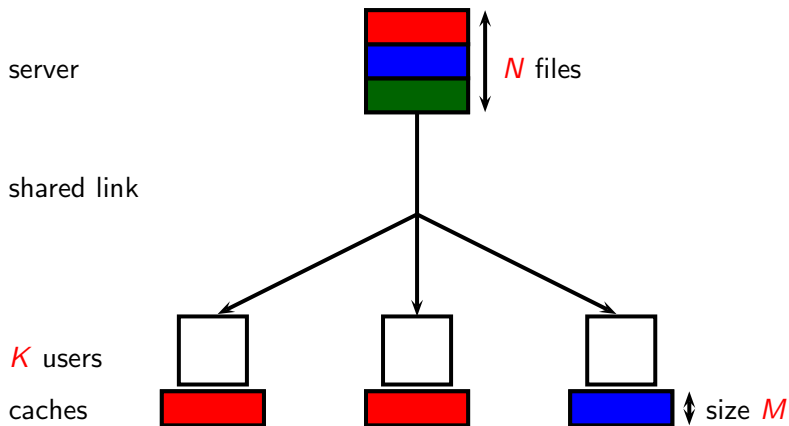
Problem Setting



Problem Setting

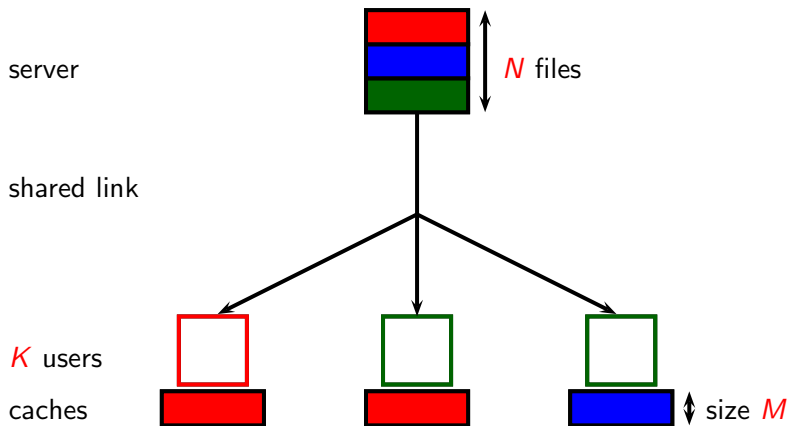


Problem Setting



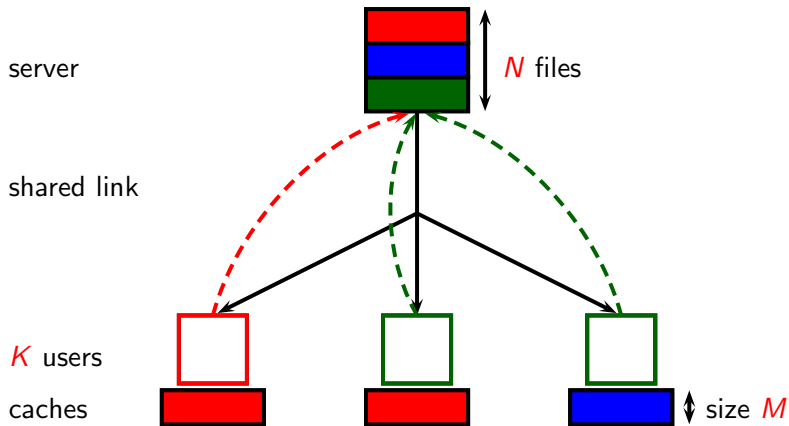
Placement: cache arbitrary function of files (linear, nonlinear, ...)

Problem Setting



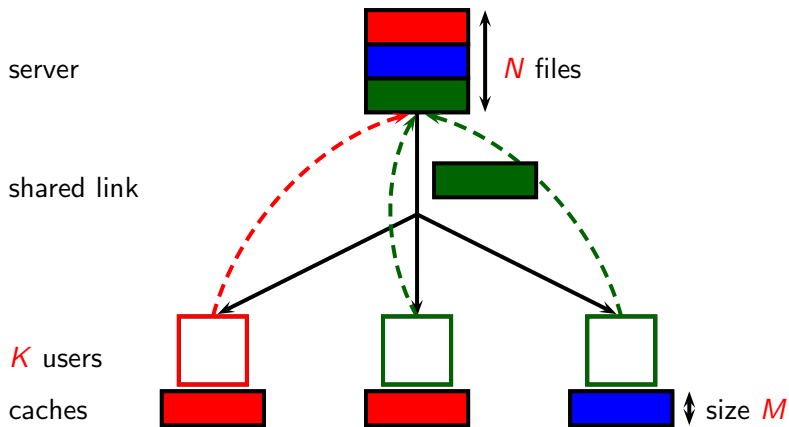
Delivery:

Problem Setting



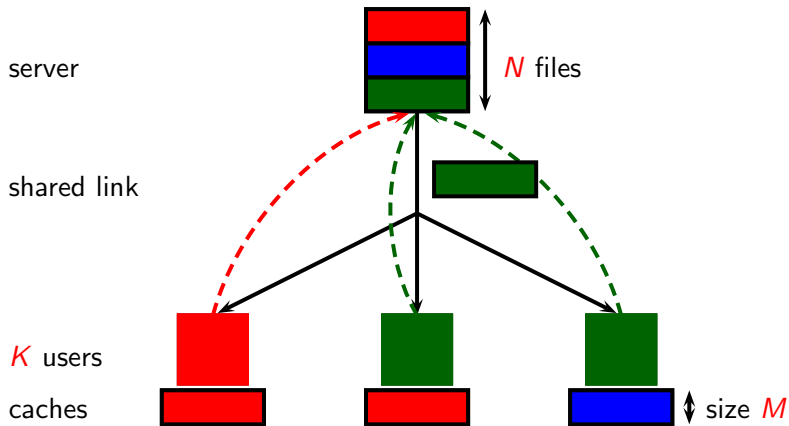
Delivery: - requests are revealed to server

Problem Setting



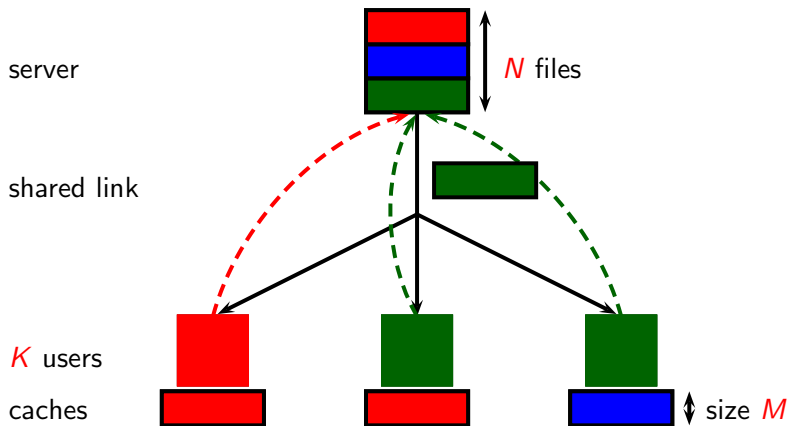
- Delivery:
- requests are revealed to server
 - server sends arbitrary function of files

Problem Setting



- Delivery:
- requests are revealed to server
 - server sends arbitrary function of files

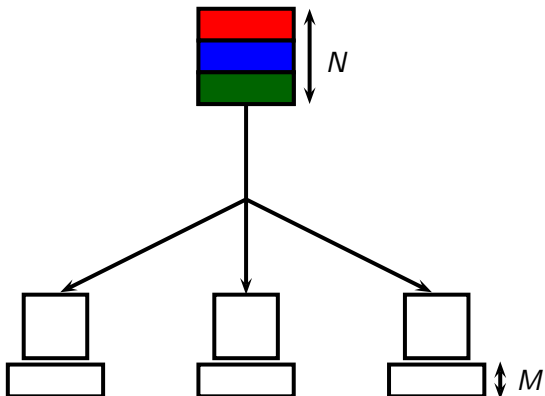
Problem Setting



Question: smallest worst-case rate $R(M)$ needed in delivery phase?

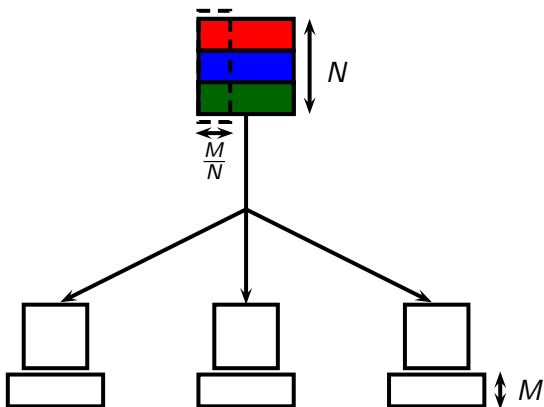
Conventional Caching Scheme

N files, K users, cache size M



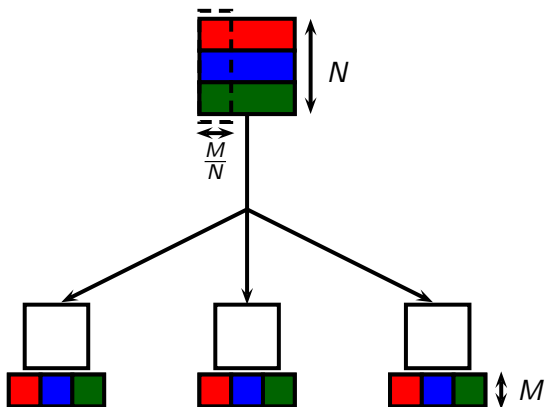
Conventional Caching Scheme

N files, K users, cache size M



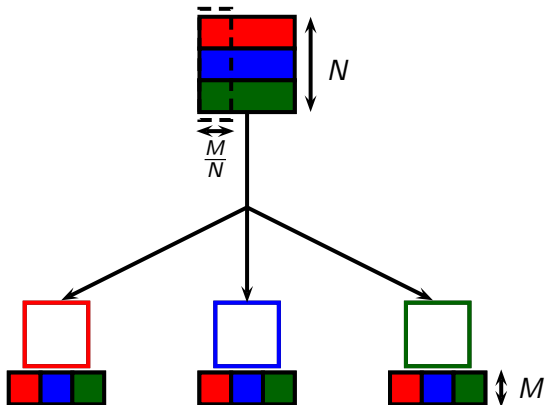
Conventional Caching Scheme

N files, K users, cache size M



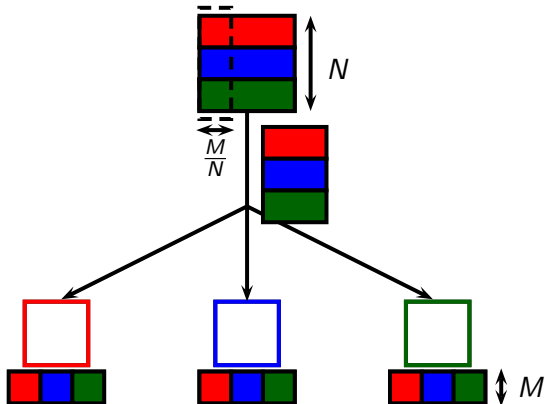
Conventional Caching Scheme

N files, K users, cache size M



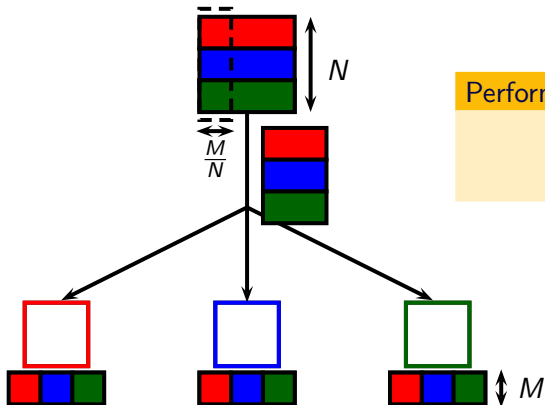
Conventional Caching Scheme

N files, K users, cache size M



Conventional Caching Scheme

N files, K users, cache size M

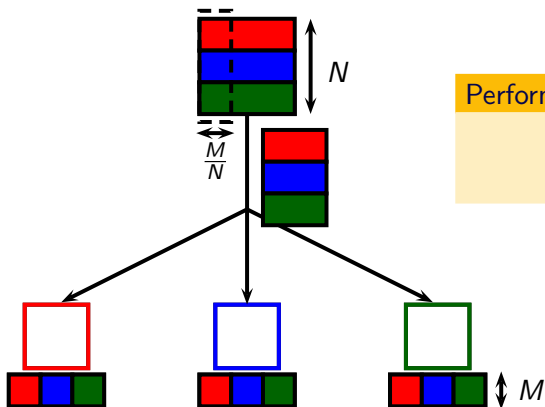


Performance of conventional scheme:

$$R(M) = K \cdot (1 - M/N)$$

Conventional Caching Scheme

N files, K users, cache size M

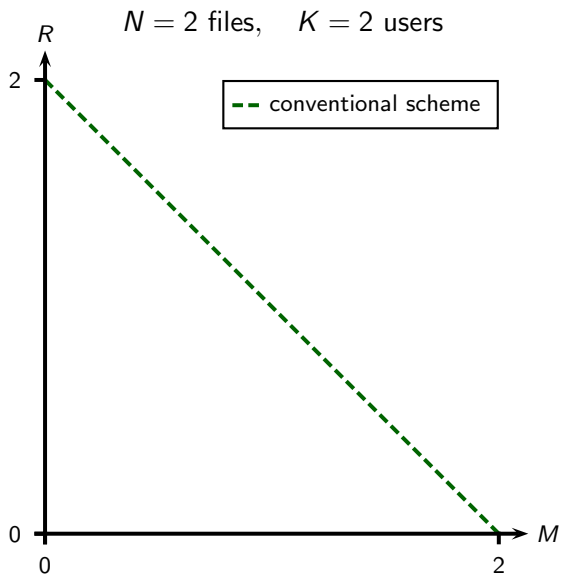


Performance of conventional scheme:

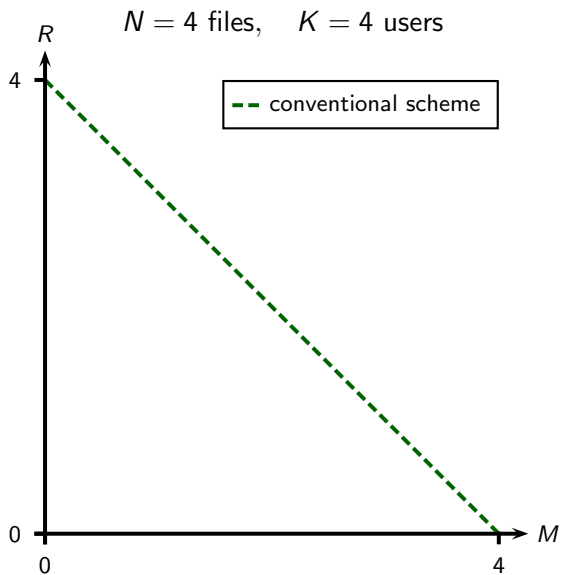
$$R(M) = K \cdot (1 - M/N)$$

- Caches provide content locally \Rightarrow local cache size matters
- Identical cache content at users

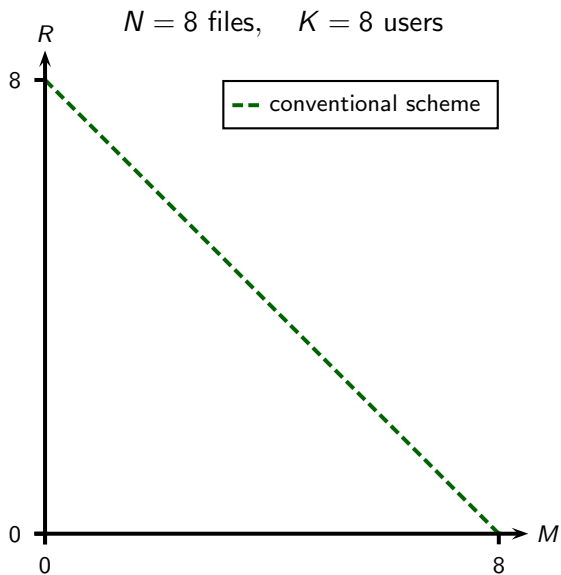
Conventional Caching Scheme



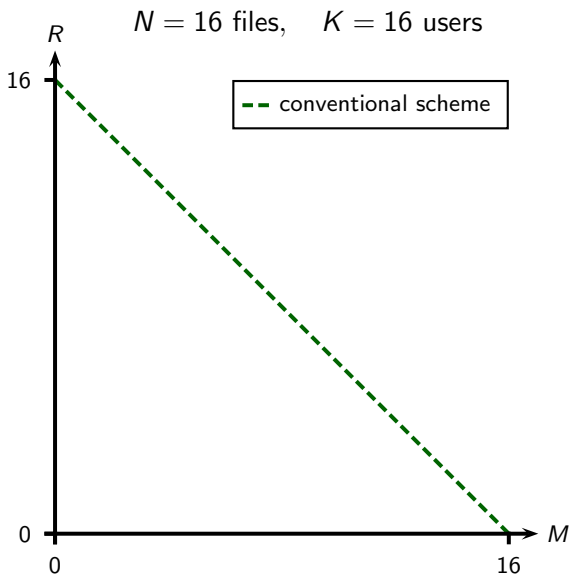
Conventional Caching Scheme



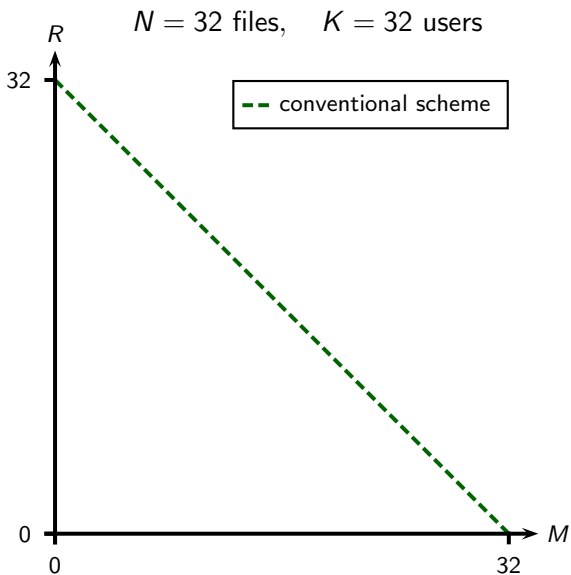
Conventional Caching Scheme



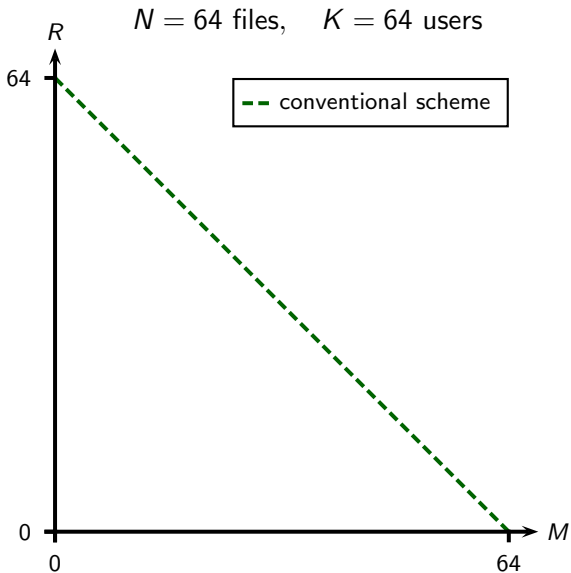
Conventional Caching Scheme



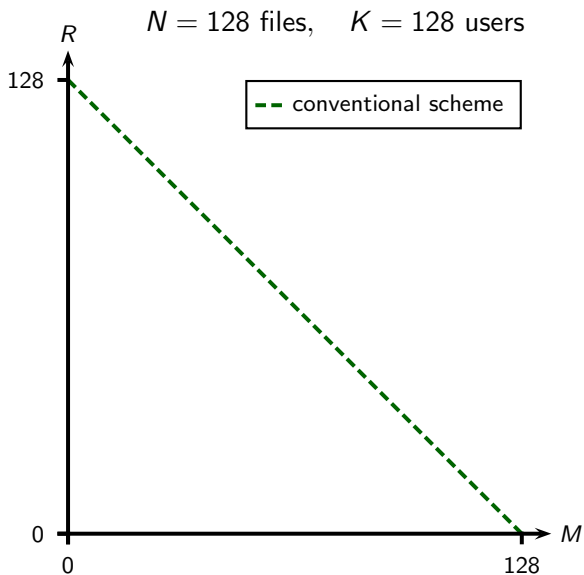
Conventional Caching Scheme



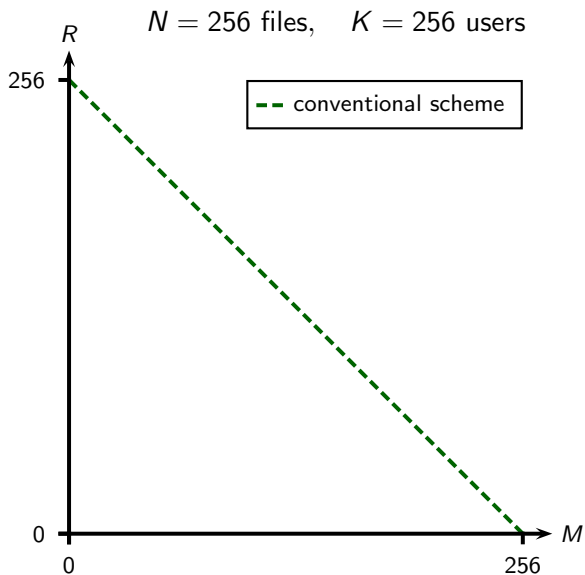
Conventional Caching Scheme



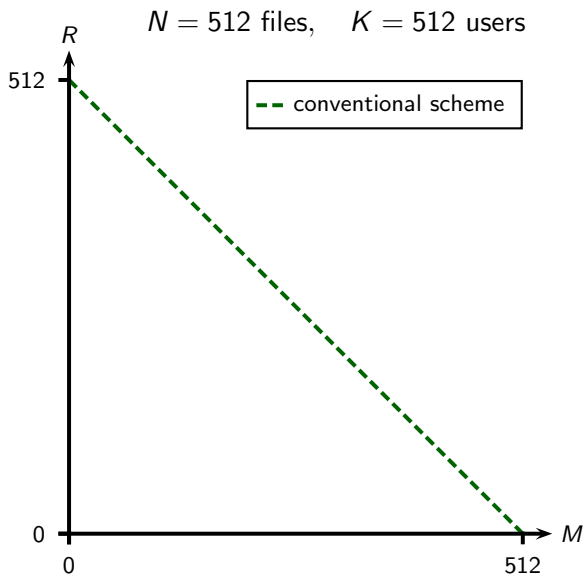
Conventional Caching Scheme



Conventional Caching Scheme



Conventional Caching Scheme



Proposed Caching Scheme

N files, K users, cache size M

Design guidelines advocated in this work:

- The main gain in caching is global
- Global cache size matters
- Different cache content at users

Proposed Caching Scheme

N files, K users, cache size M

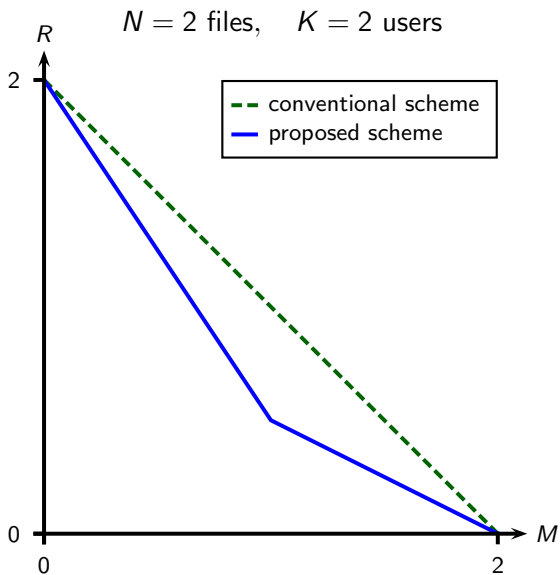
Design guidelines advocated in this work:

- The main gain in caching is global
- Global cache size matters
- Different cache content at users

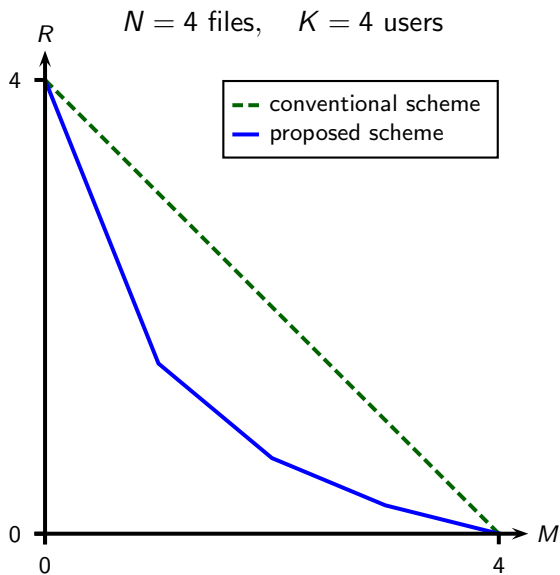
Performance of proposed scheme:

$$R(M) = K \cdot (1 - M/N) \cdot \frac{1}{1 + KM/N}$$

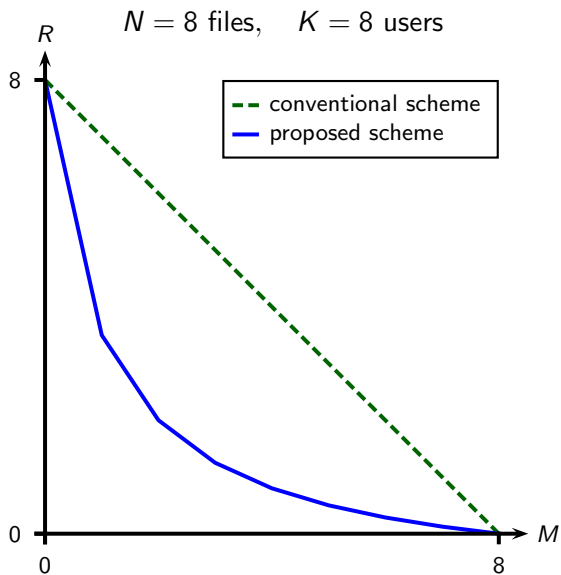
Proposed Caching Scheme



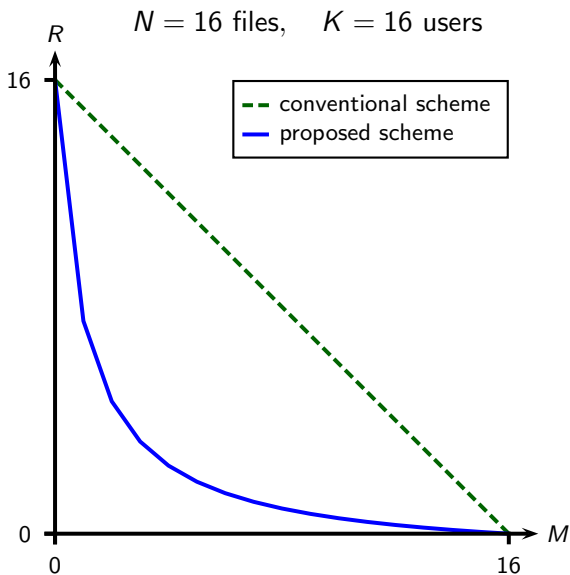
Proposed Caching Scheme



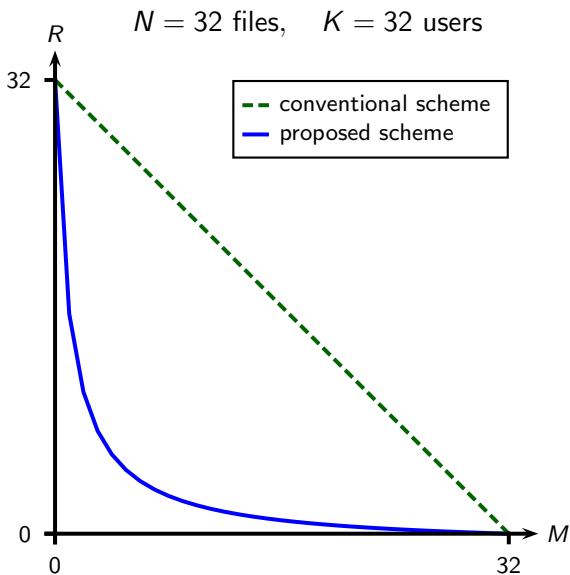
Proposed Caching Scheme



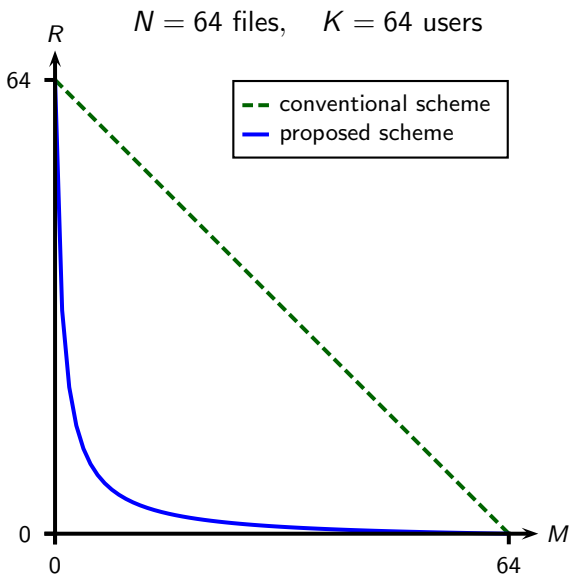
Proposed Caching Scheme



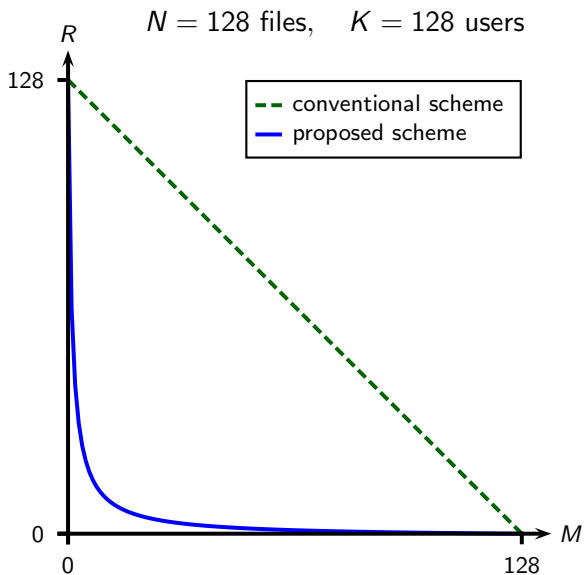
Proposed Caching Scheme



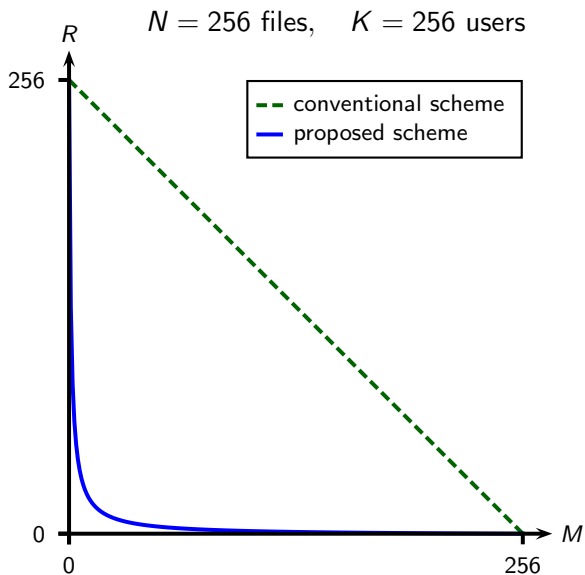
Proposed Caching Scheme



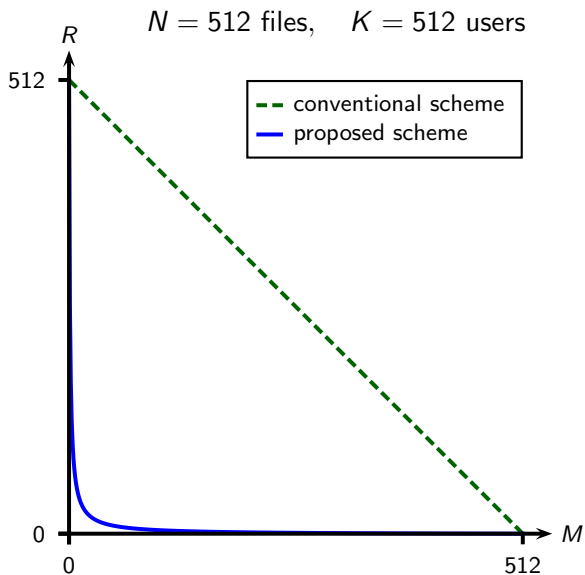
Proposed Caching Scheme



Proposed Caching Scheme

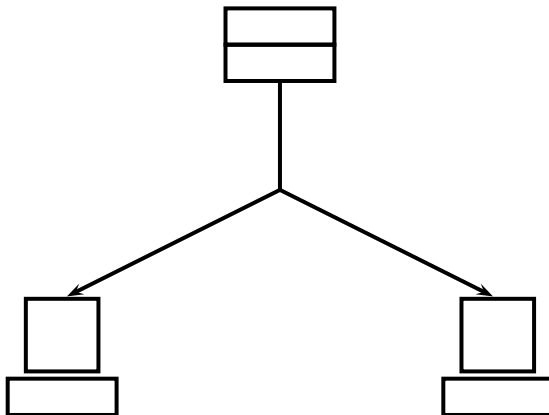


Proposed Caching Scheme



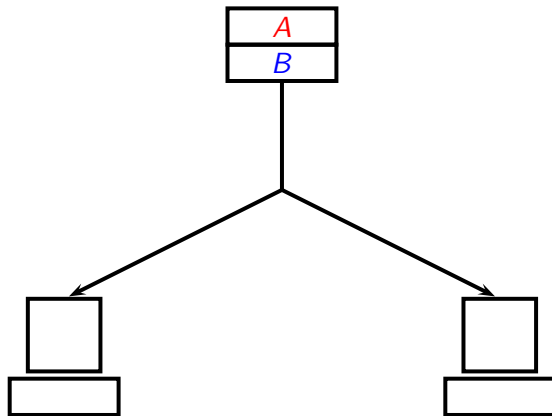
Recall: Conventional Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



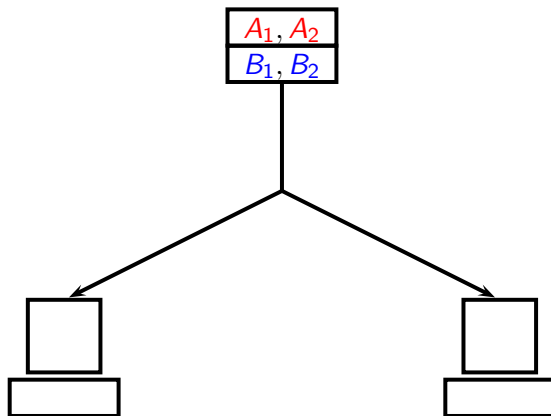
Recall: Conventional Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



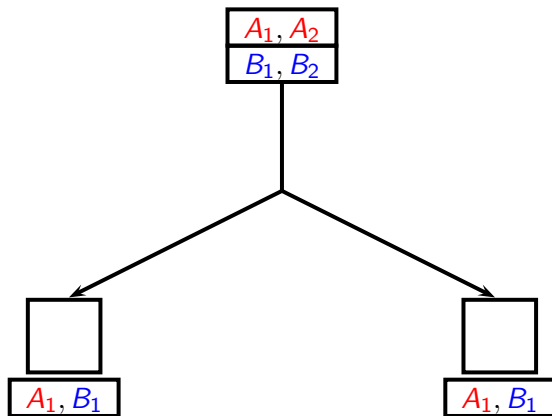
Recall: Conventional Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



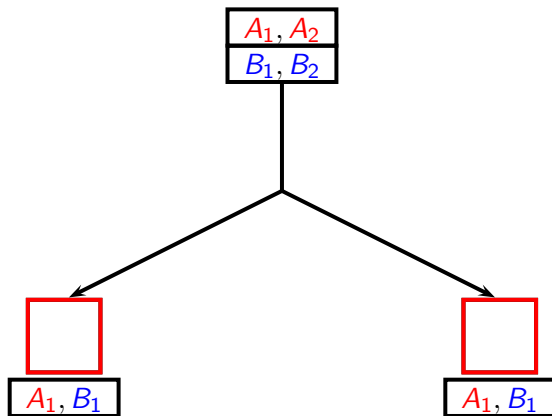
Recall: Conventional Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



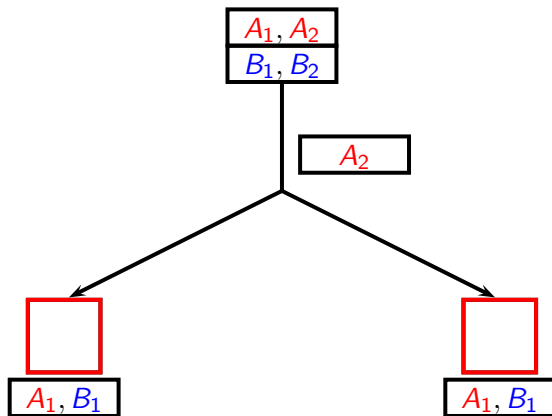
Recall: Conventional Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



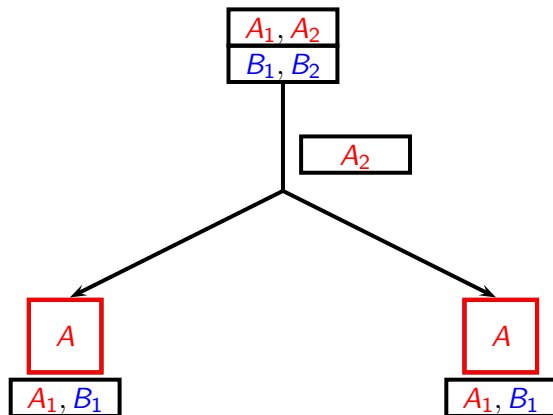
Recall: Conventional Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



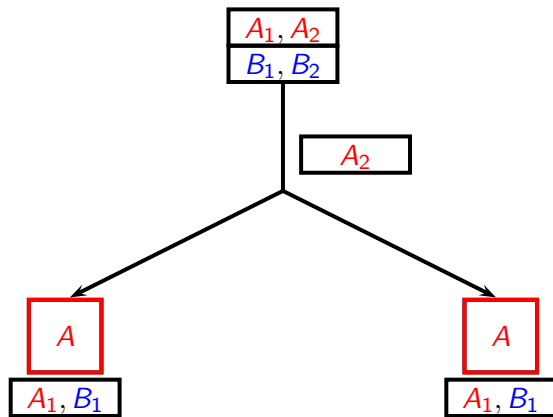
Recall: Conventional Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



Recall: Conventional Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$

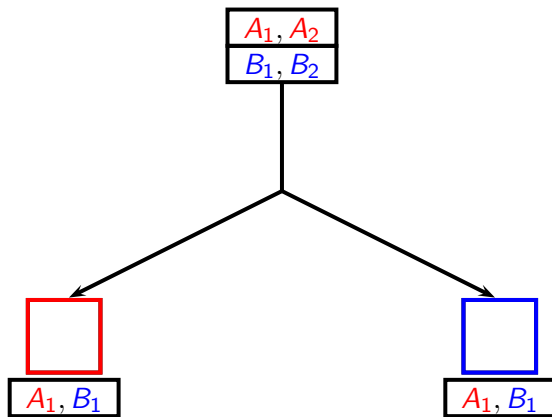


⇒ Identical cache content at users

⇒ Gain from delivering content locally

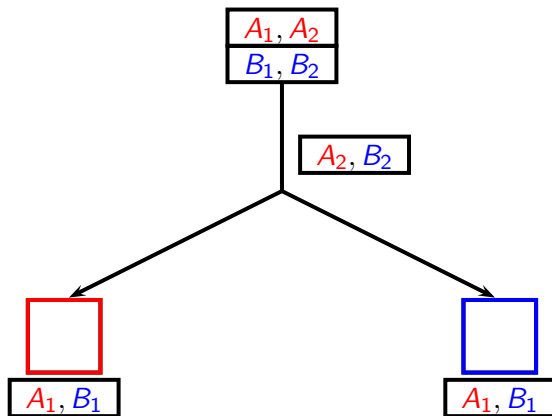
Recall: Conventional Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



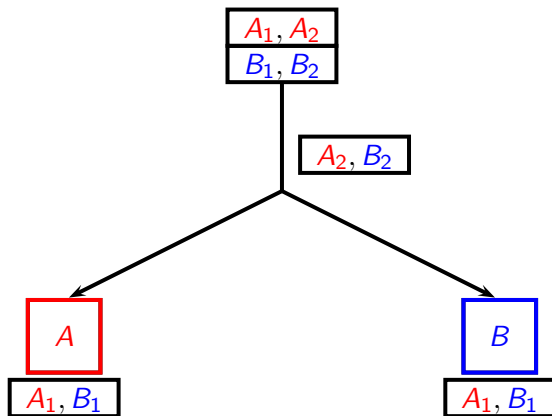
Recall: Conventional Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



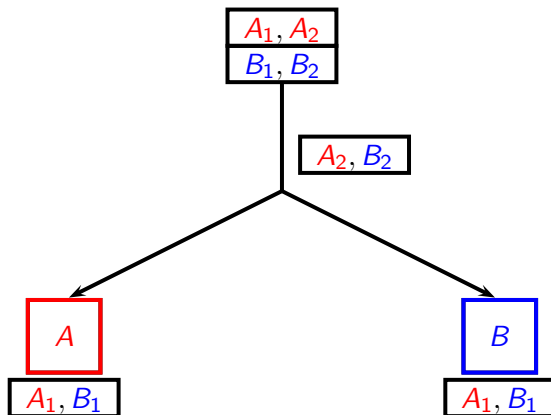
Recall: Conventional Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



Recall: Conventional Scheme

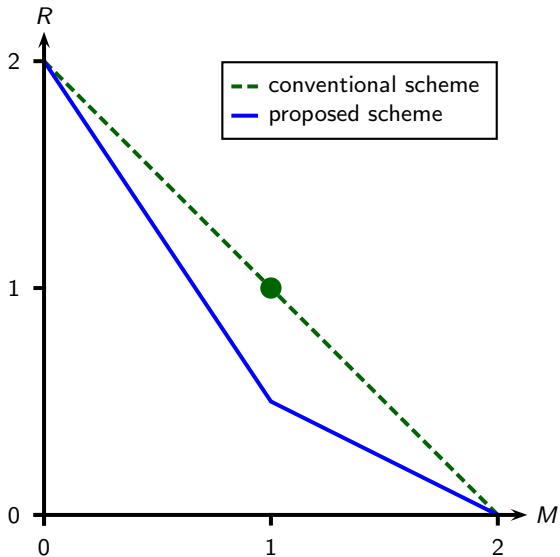
$N = 2$ files, $K = 2$ users, cache size $M = 1$



⇒ Multicast only possible for users with **same** demand

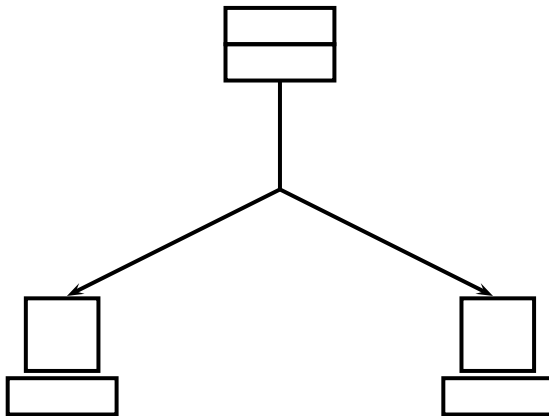
Recall: Conventional Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



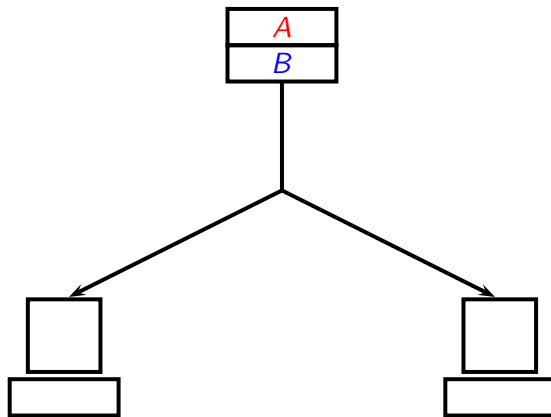
Proposed Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



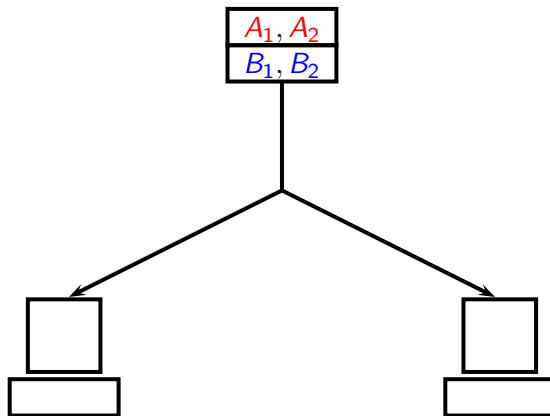
Proposed Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



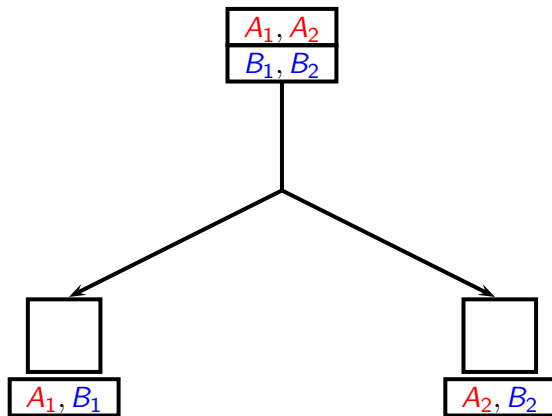
Proposed Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



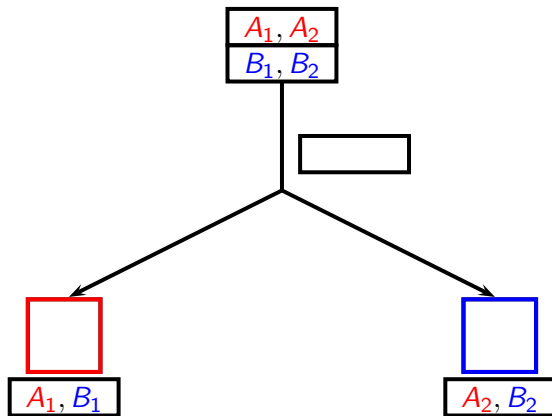
Proposed Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



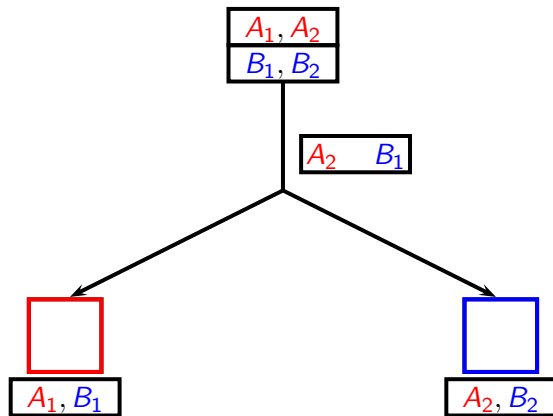
Proposed Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



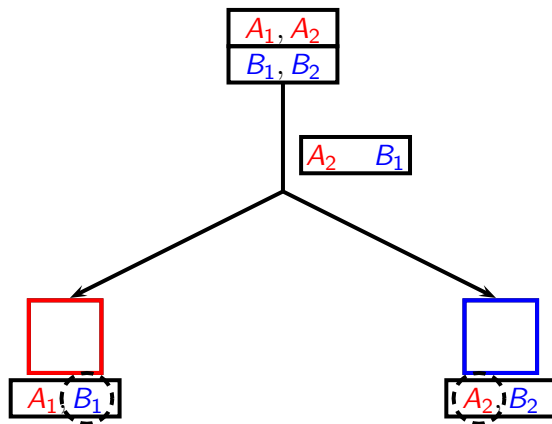
Proposed Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



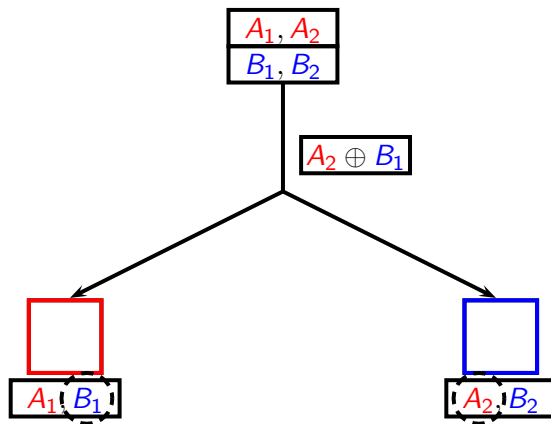
Proposed Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



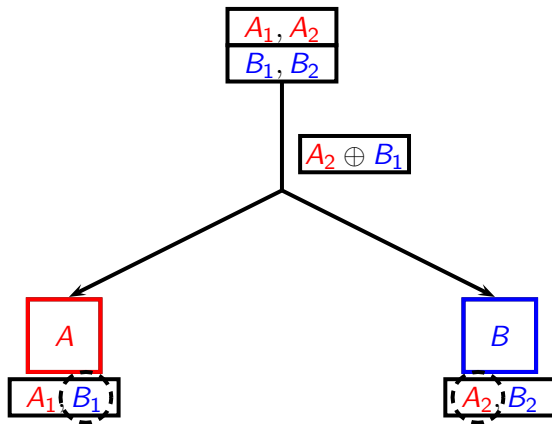
Proposed Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



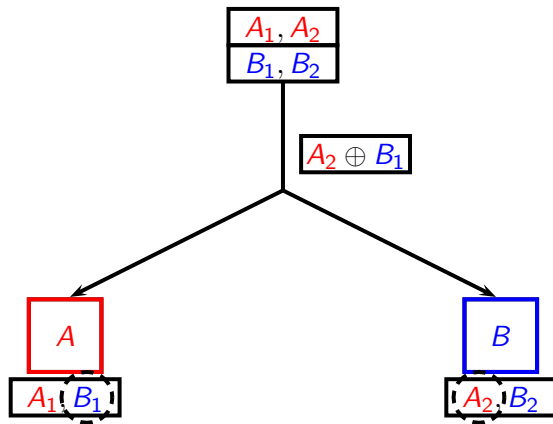
Proposed Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



Proposed Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$

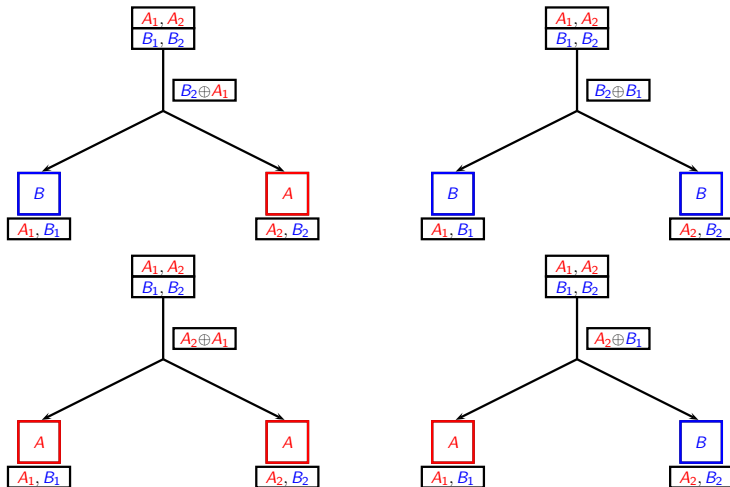


⇒ Different cache content at users

⇒ Multicast to 2 users with different demands

Proposed Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$

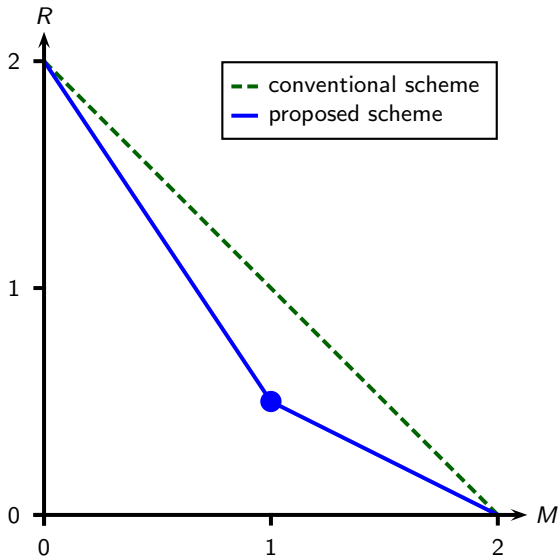


⇒ Works for all possible user requests

⇒ **Simultaneous** multicasting gain

Proposed Scheme

$N = 2$ files, $K = 2$ users, cache size $M = 1$



Proposed Scheme

N files, K users, cache size M

- Scheme can be generalized to arbitrary:
 - Number of files N
 - Number of users K
 - Cache size M

Proposed Scheme

N files, K users, cache size M

- Scheme can be generalized to arbitrary:
 - Number of files N
 - Number of users K
 - Cache size M
- Enables multicast to $KM/N + 1$ users with different demands

Comparison of the Two Schemes

N files, K users, cache size M

- Conventional scheme: $R(M) = K \cdot (1 - M/N)$
- Proposed scheme: $R(M) = K \cdot (1 - M/N) \cdot \frac{1}{1 + KM/N}$

Comparison of the Two Schemes

N files, K users, cache size M

- Conventional scheme: $R(M) = K \cdot (1 - M/N)$
- Proposed scheme: $R(M) = K \cdot (1 - M/N) \cdot \frac{1}{1 + KM/N}$

- Rate without caching K

Comparison of the Two Schemes

N files, K users, cache size M

- Conventional scheme: $R(M) = K \cdot (1 - M/N)$
- Proposed scheme: $R(M) = K \cdot (1 - M/N) \cdot \frac{1}{1 + KM/N}$

- Rate without caching K
- **Local caching gain** $1 - M/N$
 - Significant when **local** cache size M is of order N

Comparison of the Two Schemes

N files, K users, cache size M

- Conventional scheme: $R(M) = K \cdot (1 - M/N)$
- Proposed scheme: $R(M) = K \cdot (1 - M/N) \cdot \frac{1}{1+KM/N}$

- Rate without caching K
- Local caching gain $1 - M/N$
 - Significant when local cache size M is of order N
- Global caching gain $\frac{1}{1+KM/N}$
 - Significant when global cache size KM is of order N

Comparison of the Two Schemes

N files, K users, cache size M

- Conventional scheme: $R(M) = K \cdot (1 - M/N)$
- Proposed scheme: $R(M) = K \cdot (1 - M/N) \cdot \frac{1}{1+KM/N}$

- Rate without caching K
- Local caching gain $1 - M/N$
 - Significant when local cache size M is of order N
- Global caching gain $\frac{1}{1+KM/N}$
 - Significant when global cache size KM is of order N

⇒ Global gain can be $\Theta(K)$ smaller than local gain

Can We Do Better?

Theorem

*The proposed scheme is **optimal** to within a **constant** factor in rate.*

Can We Do Better?

Theorem

*The proposed scheme is **optimal** to within a **constant** factor in rate.*

⇒ Information-theoretic bound

Can We Do Better?

Theorem

*The proposed scheme is **optimal** to within a **constant** factor in rate.*

⇒ Information-theoretic bound

⇒ Constant is independent of problem parameters N, K, M

Can We Do Better?

Theorem

*The proposed scheme is **optimal** to within a **constant** factor in rate.*

- ⇒ Information-theoretic bound
- ⇒ Constant is independent of problem parameters N, K, M
- ⇒ No other significant gain besides local and global

Conclusions

A New Approach to Caching

Conclusions

A New Approach to Caching

- The main gain in caching is global
 - ⇒ Multicast to users with **different** demands

Conclusions

A New Approach to Caching

- The main gain in caching is global
 - ⇒ Multicast to users with **different** demands
- **Global** cache size matters

Conclusions

A New Approach to Caching

- The main gain in caching is global
 - ⇒ Multicast to users with **different** demands
- **Global** cache size matters
- Statistically identical users ⇒ **different** cache content

Conclusions

A New Approach to Caching

- The main gain in caching is global
 - ⇒ Multicast to users with **different** demands
- **Global** cache size matters
- Statistically identical users ⇒ **different** cache content
- Significant improvement over conventional caching schemes
 - ⇒ Reduction in rate up to order of number of users

Conclusions

A New Approach to Caching

- The main gain in caching is global
 - ⇒ Multicast to users with **different** demands
- **Global** cache size matters
- Statistically identical users ⇒ **different** cache content
- Significant improvement over conventional caching schemes
 - ⇒ Reduction in rate up to order of number of users
- Papers available on arXiv
 - ⇒ Maddah-Ali, Niesen: “Fundamental Limits of Caching”
 - ⇒ Maddah-Ali, Niesen: “Decentralized Caching Attains Order-Optimal Memory-Rate Tradeoff”
 - ⇒ Niesen, Maddah-Ali: “Coded Caching with Nonuniform Demands”