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
- ...

$\Rightarrow$ Places significant stress on service provider's networks

Video on Demand

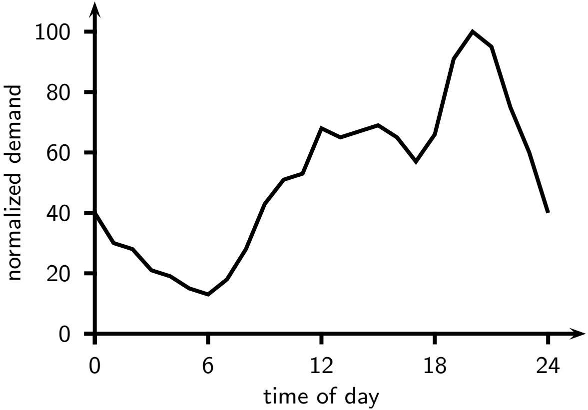
Video on demand is getting increasingly popular:

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

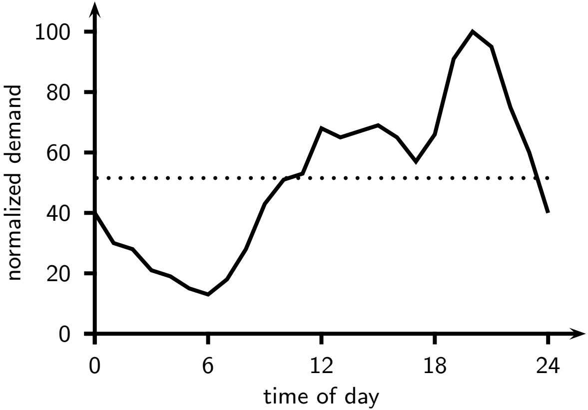
$\Rightarrow$ Places significant stress on service provider's networks

$\Rightarrow$ Caching (prefetching) can be used to mitigate this stress

Caching (Prefetching)

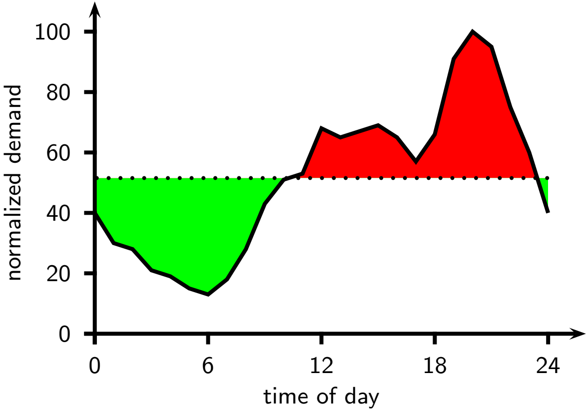


Caching (Prefetching)



- 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

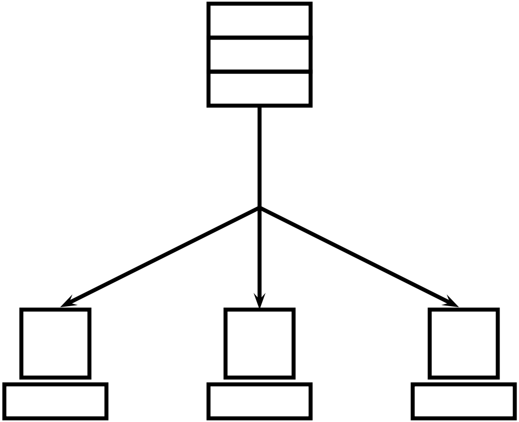
Problem Setting

server

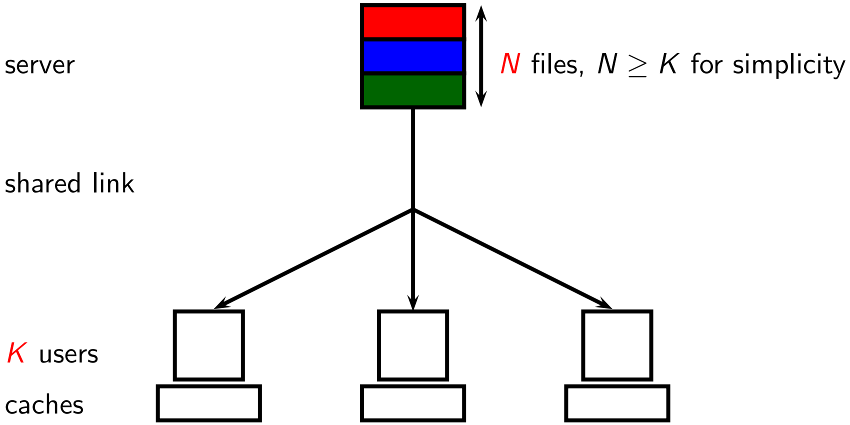
shared link

K users

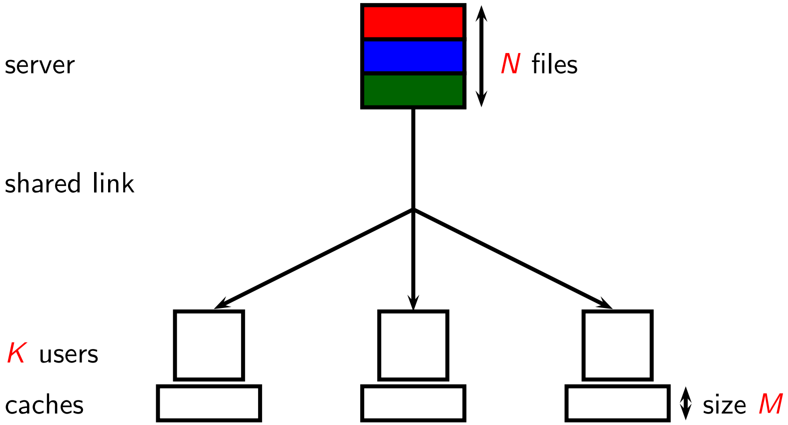
caches



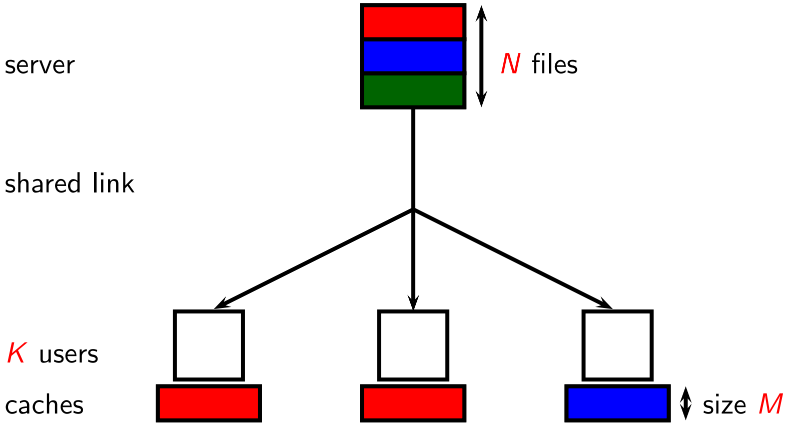
Problem Setting



Problem Setting

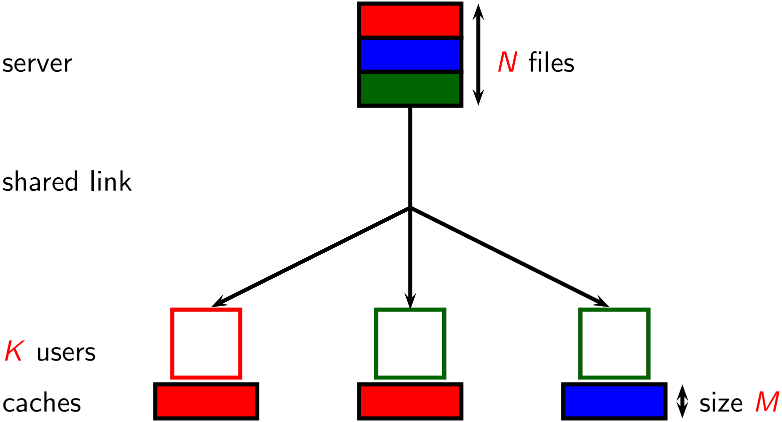


Problem Setting



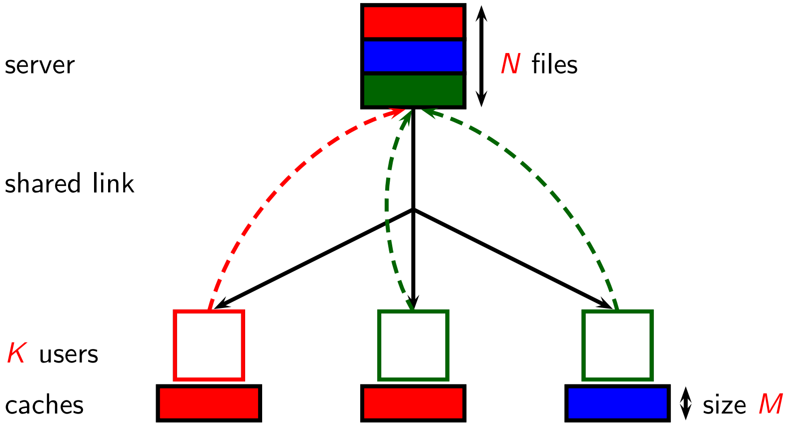
Placement: cache arbitrary function of files (linear, nonlinear, ...)

Problem Setting



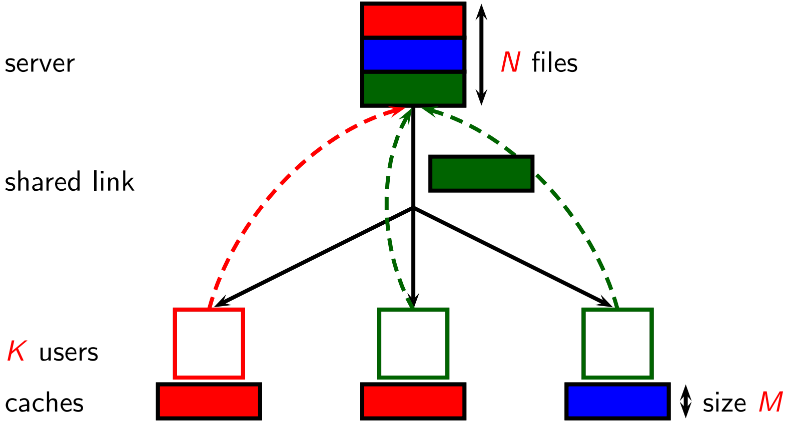
Delivery:

Problem Setting



Delivery: - requests are revealed to server

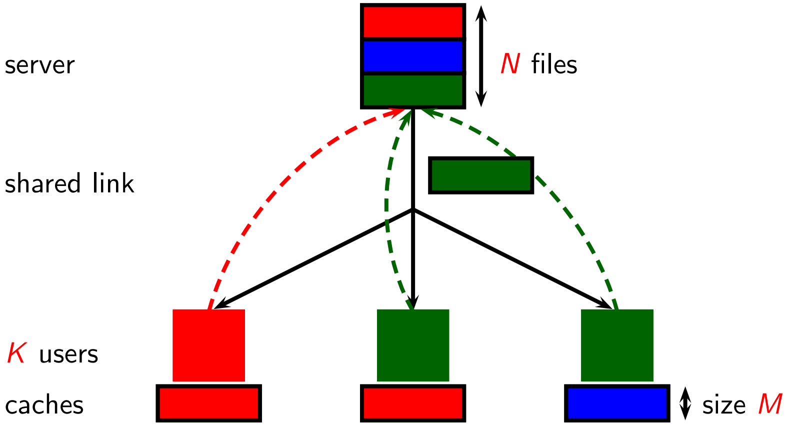
Problem Setting



Delivery: - requests are revealed to server

- server sends arbitrary function of files

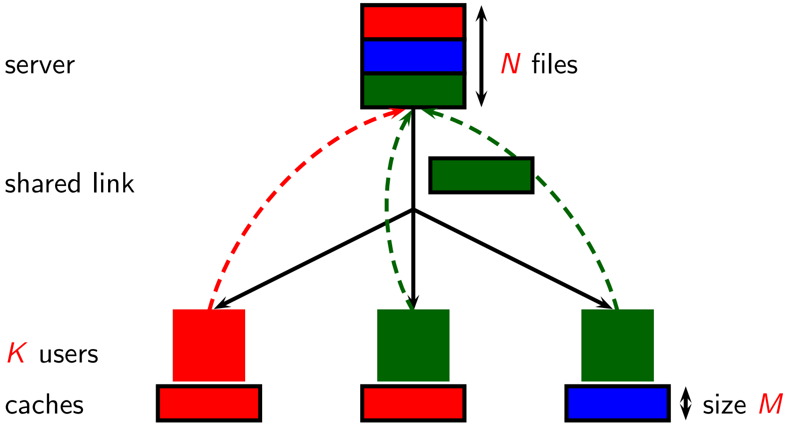
Problem Setting



Delivery: - requests are revealed to server

- server sends arbitrary function of files

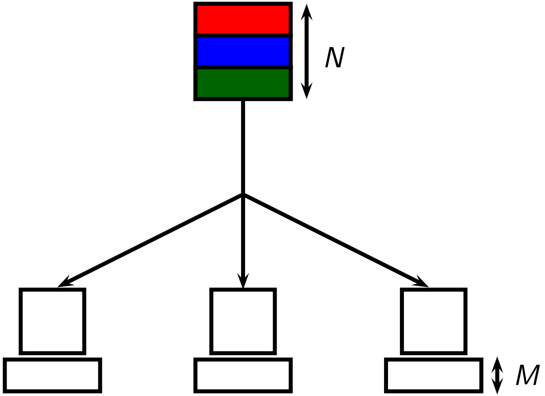
Problem Setting



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

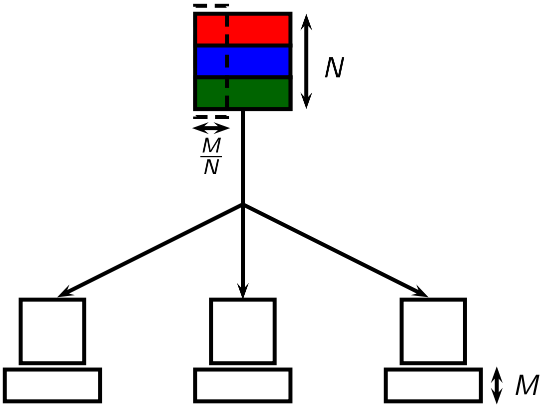
Conventional Caching Scheme

N files, K users, cache size M



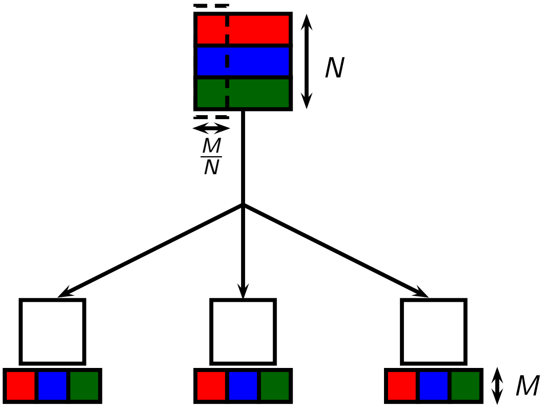
Conventional Caching Scheme

N files, K users, cache size M



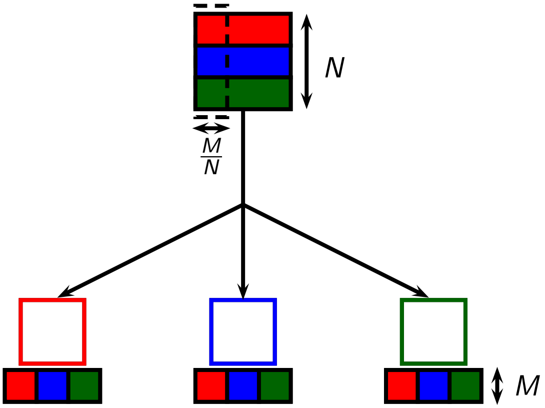
Conventional Caching Scheme

N files, K users, cache size M



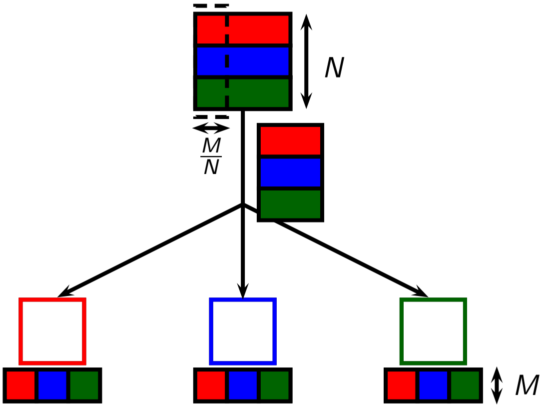
Conventional Caching Scheme

N files, K users, cache size M



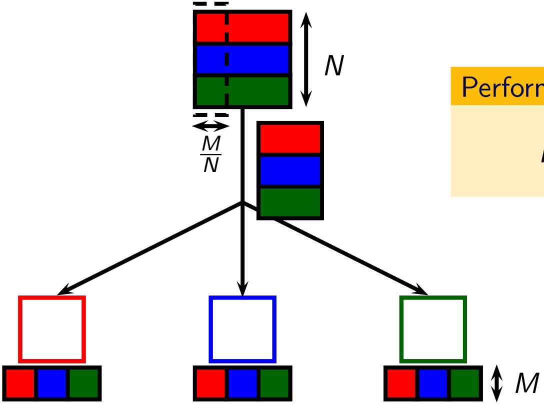
Conventional Caching Scheme

N files, K users, cache size M



Conventional Caching Scheme

N files, K users, cache size M

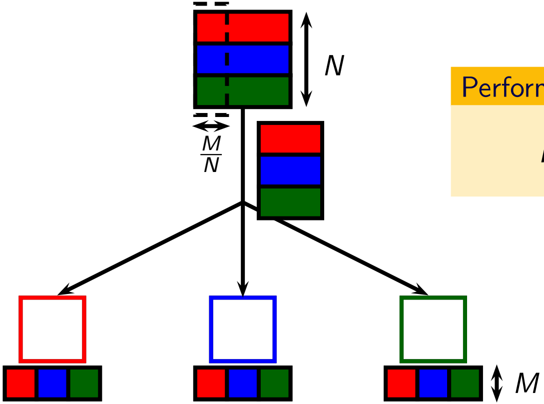


Performance of conventional scheme:

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

Conventional Caching Scheme

N files, K users, cache size M



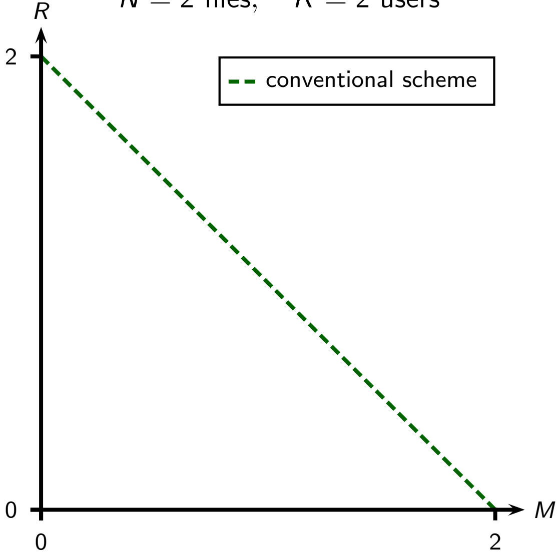
Performance of conventional scheme:

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

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

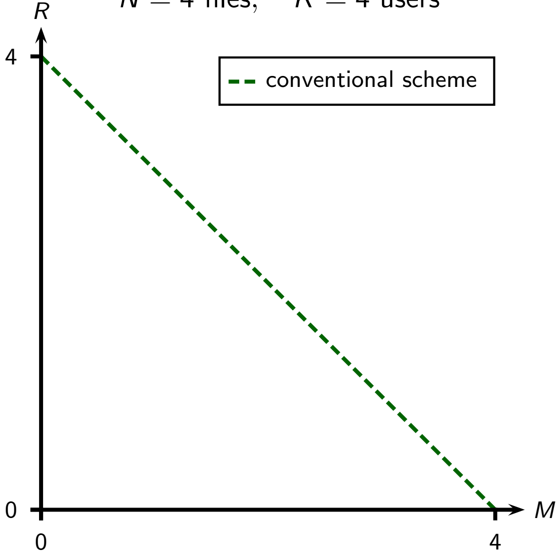
Conventional Caching Scheme

$N = 2$ files, $\quad K = 2$ users



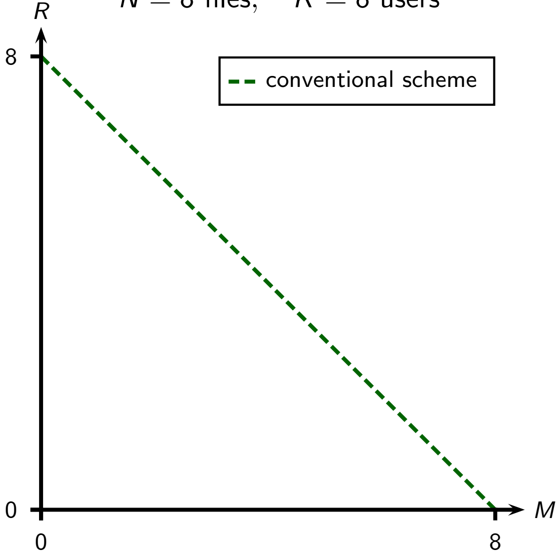
Conventional Caching Scheme

$N = 4$ files, $\quad K = 4$ users



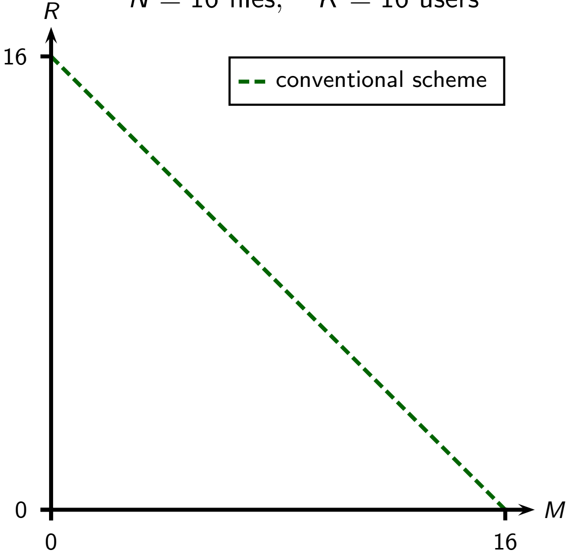
Conventional Caching Scheme

$N = 8$ files, $\quad K = 8$ users



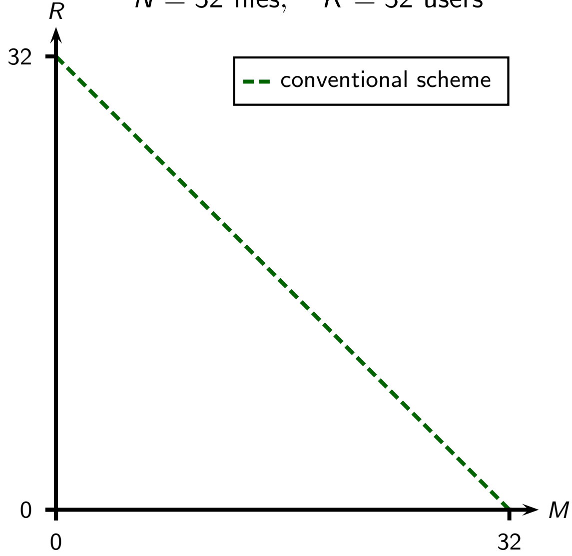
Conventional Caching Scheme

$N = 16$ files, $\quad K = 16$ users



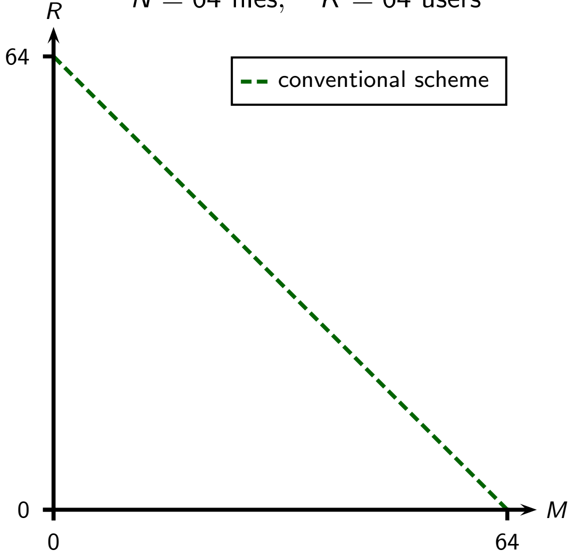
Conventional Caching Scheme

$N = 32$ files, $\quad K = 32$ users



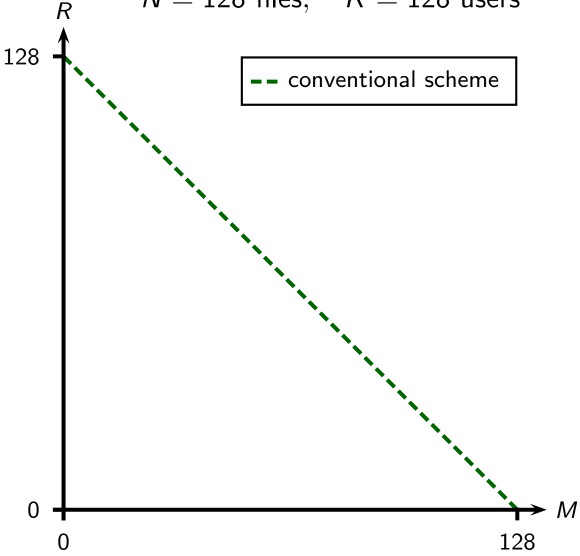
Conventional Caching Scheme

$N = 64$ files, $\quad K = 64$ users



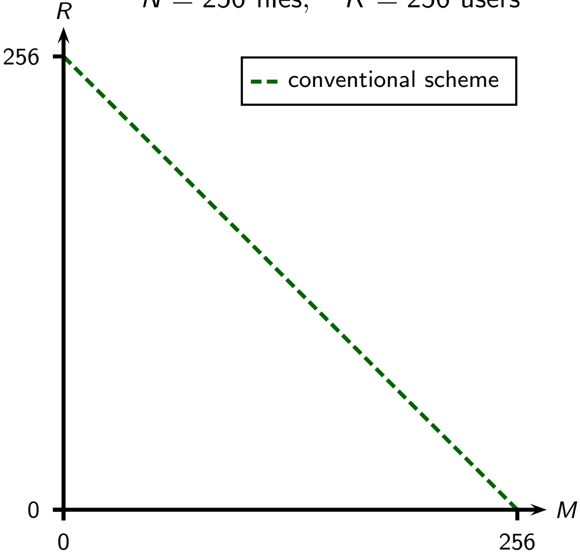
Conventional Caching Scheme

$N = 128$ files, $K = 128$ users



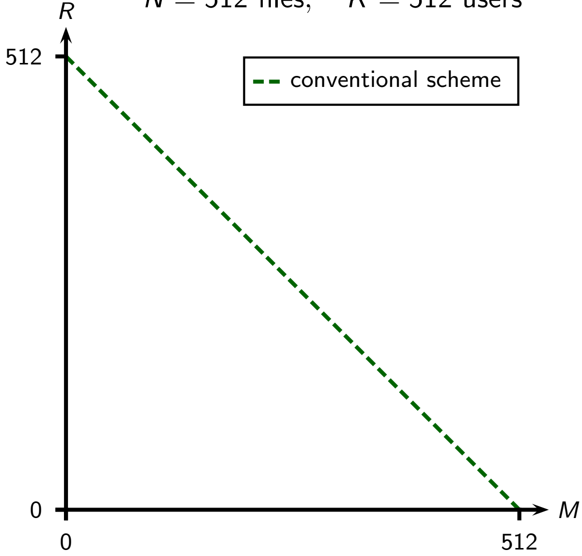
Conventional Caching Scheme

$N = 256$ files, $\quad K = 256$ users



Conventional Caching Scheme

$N = 512$ files, $\quad K = 512$ 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

Proposed Caching Scheme

N files, K users, cache size M

Design guidelines advocated in this work:

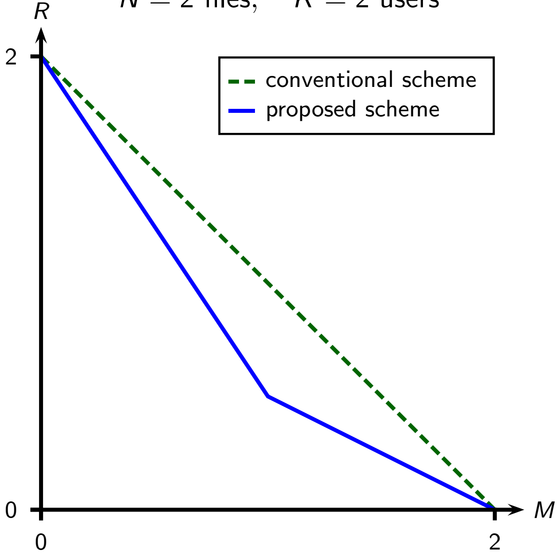
- The main gain in caching is global
- Global cache size matters
- Different cache content at users

Performance of proposed scheme:

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

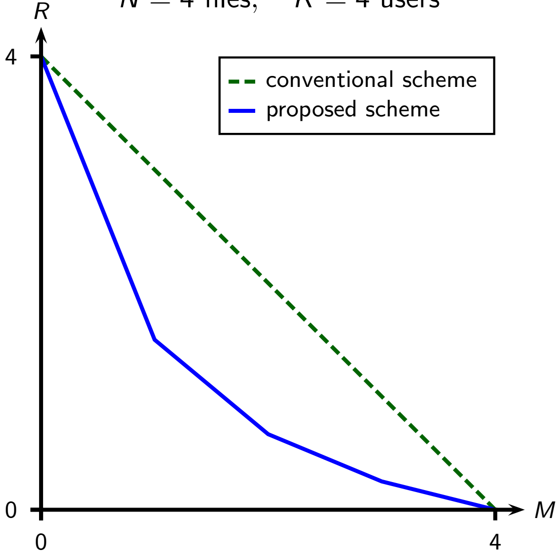
Proposed Caching Scheme

$N = 2$ files, $\quad K = 2$ users



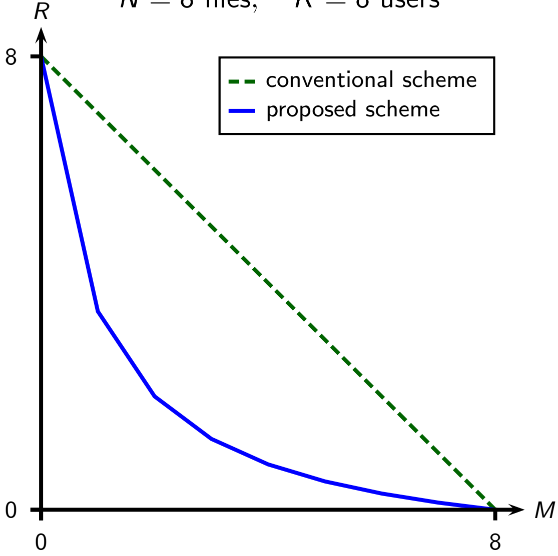
Proposed Caching Scheme

$N = 4$ files, $K = 4$ users



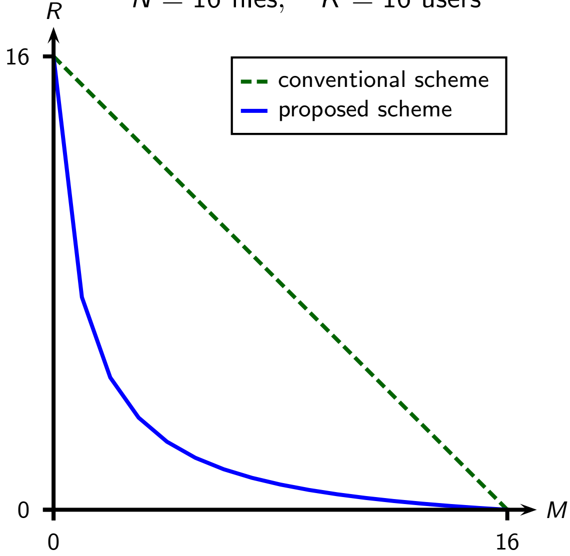
Proposed Caching Scheme

$N = 8$ files, $K = 8$ users

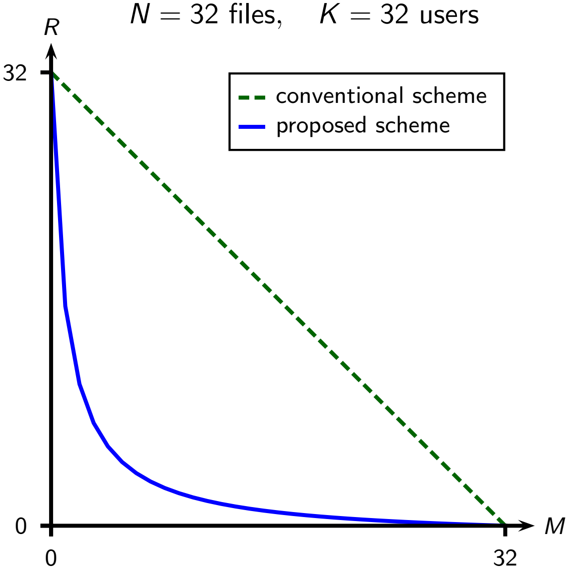


Proposed Caching Scheme

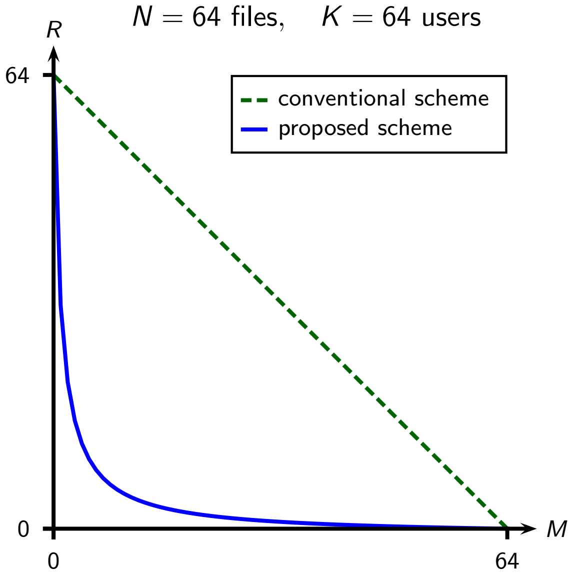
$N = 16$ files, $\quad K = 16$ users



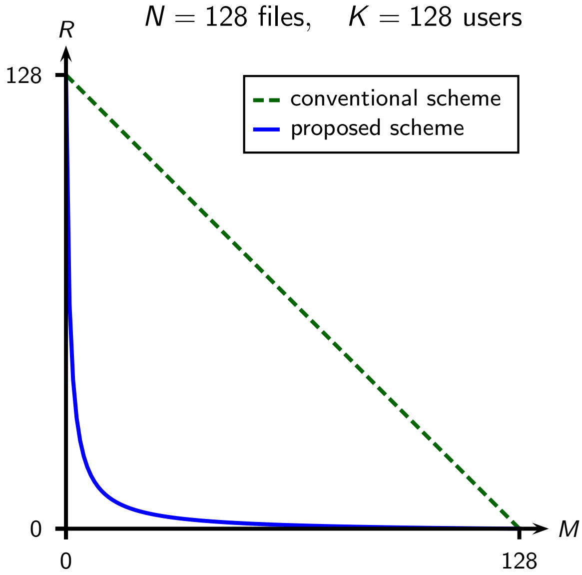
Proposed Caching Scheme



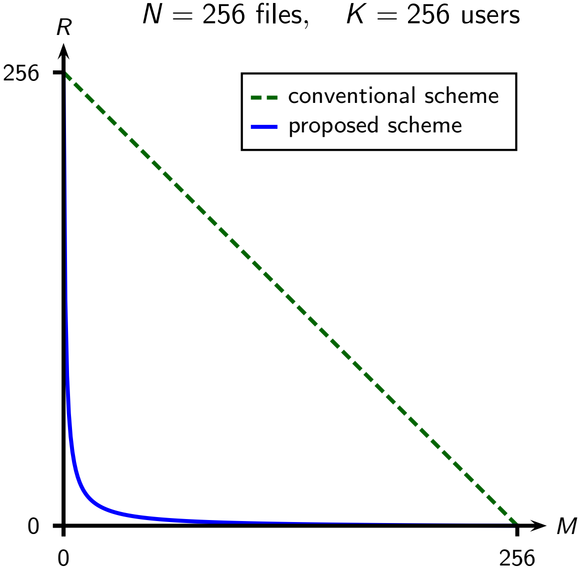
Proposed Caching Scheme



Proposed Caching Scheme

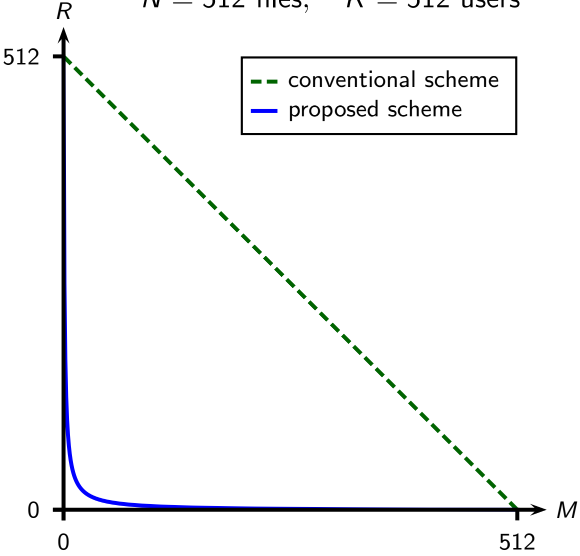


Proposed Caching Scheme



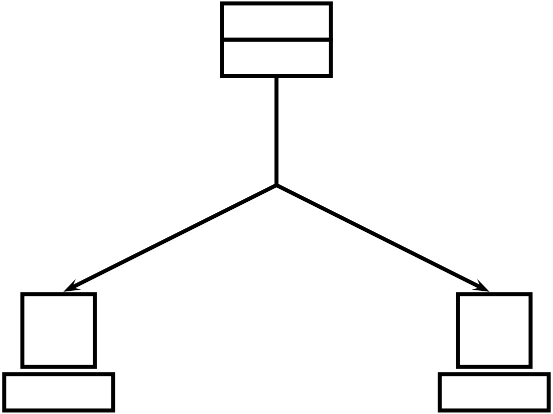
Proposed Caching Scheme

$N = 512$ files, $\quad K = 512$ users



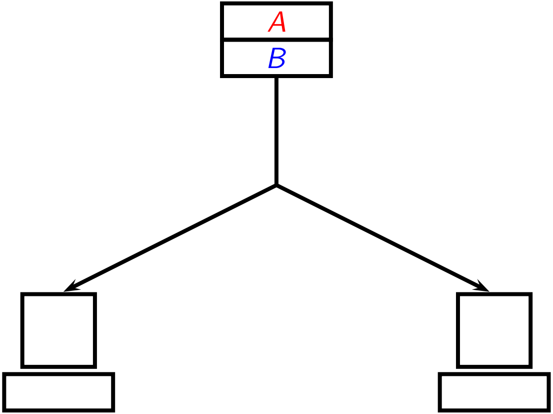
Recall: Conventional Scheme

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



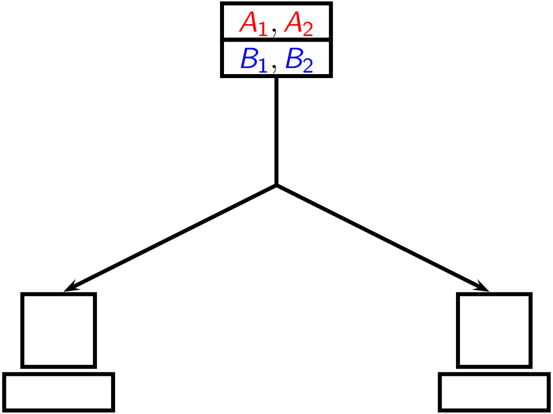
Recall: Conventional Scheme

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



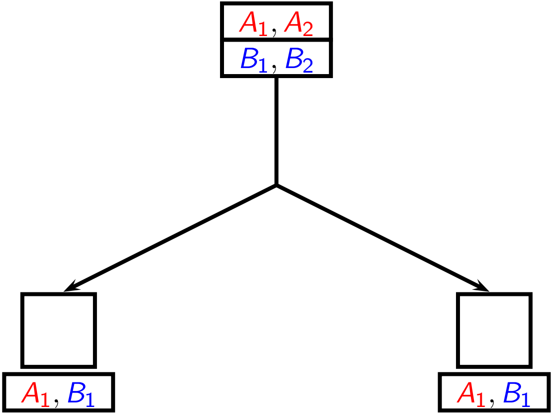
Recall: Conventional Scheme

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



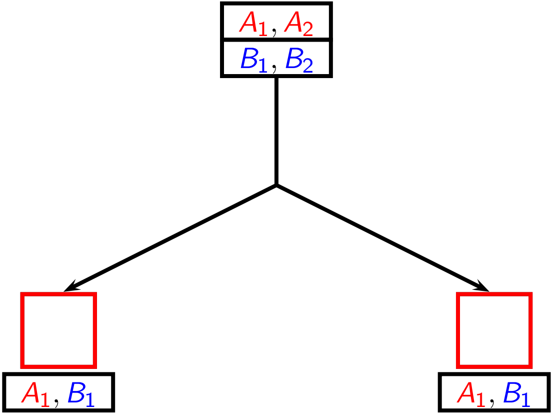
Recall: Conventional Scheme

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



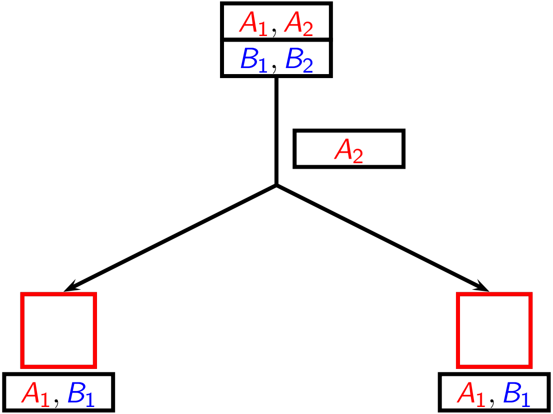
Recall: Conventional Scheme

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



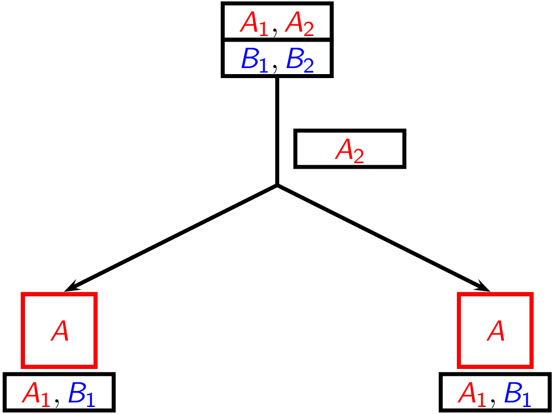
Recall: Conventional Scheme

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



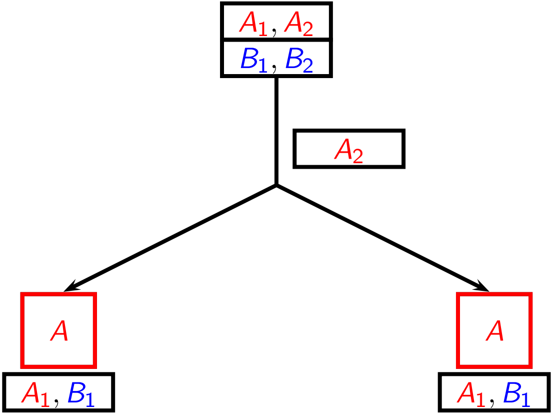
Recall: Conventional Scheme

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



Recall: Conventional Scheme

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

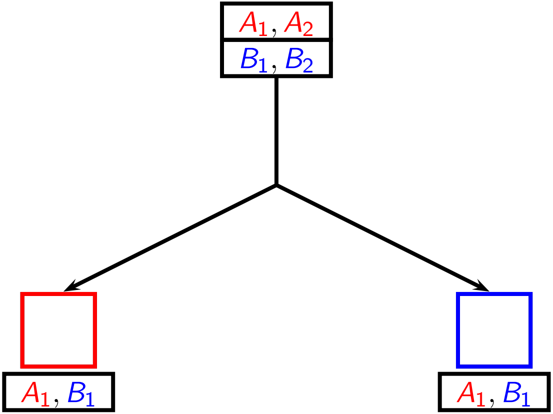


$\Rightarrow$ Identical cache content at users

$\Rightarrow$ Gain from delivering content locally

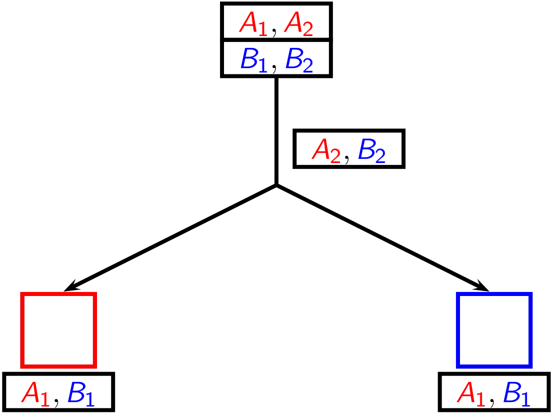
Recall: Conventional Scheme

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



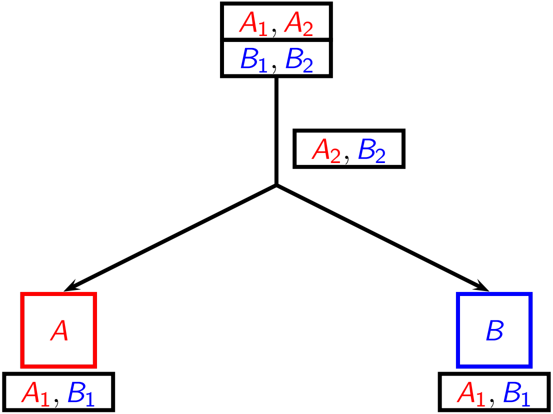
Recall: Conventional Scheme

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



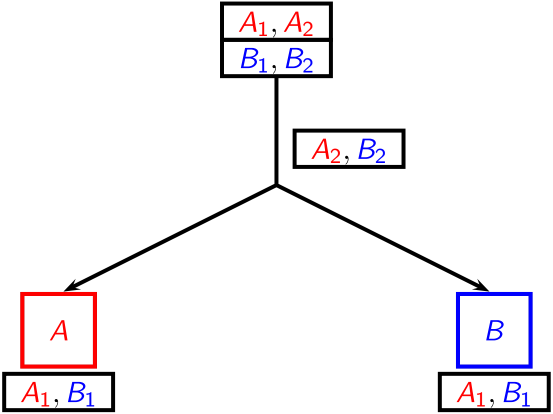
Recall: Conventional Scheme

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



Recall: Conventional Scheme

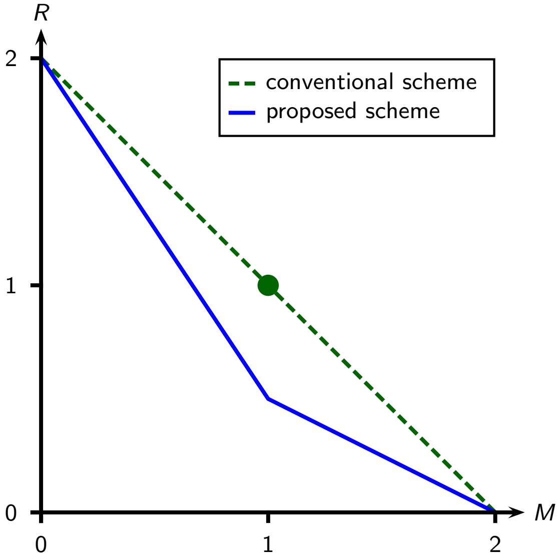
$N = 2$ files, $K = 2$ users, cache size $M = 1$



$\Rightarrow$ Multicast only possible for users with same demand

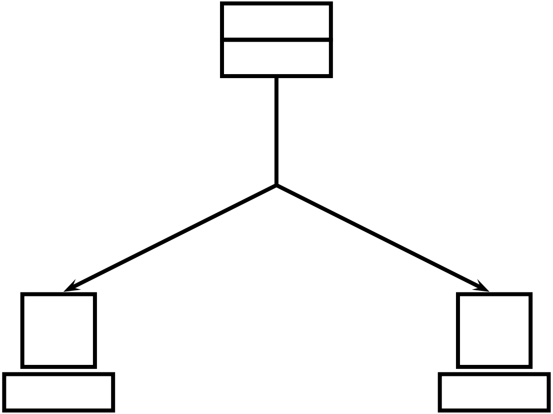
Recall: Conventional Scheme

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



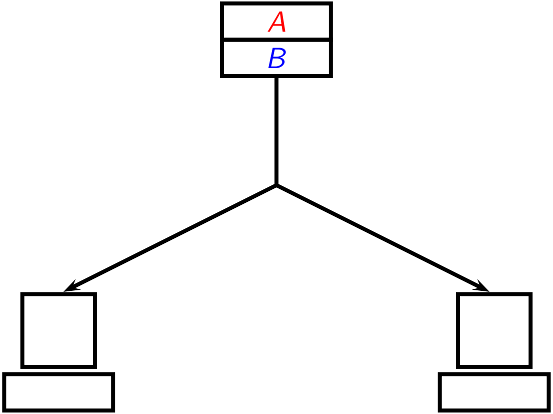
Proposed Scheme

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



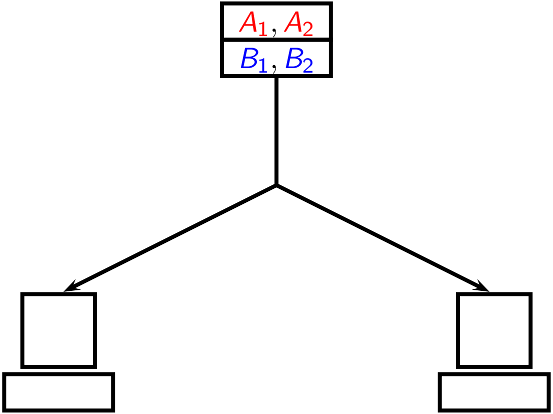
Proposed Scheme

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



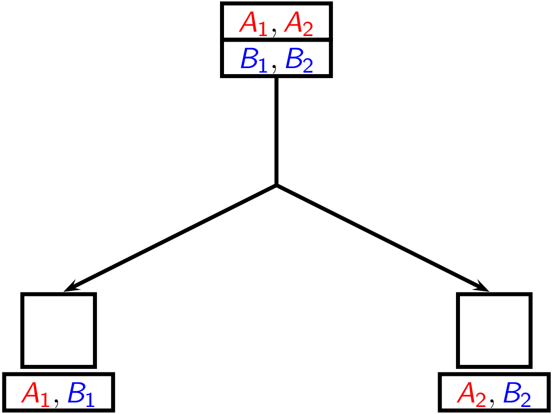
Proposed Scheme

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



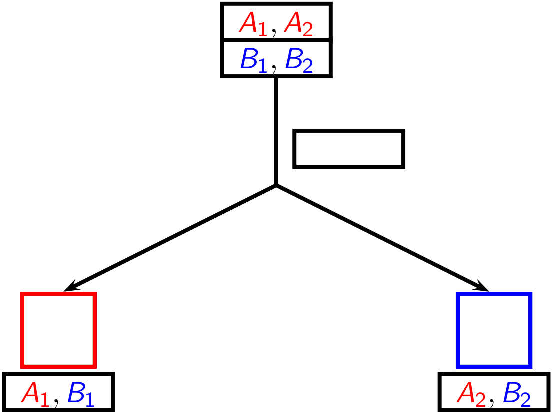
Proposed Scheme

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



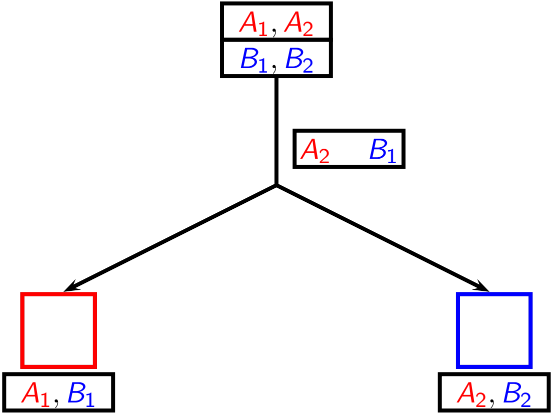
Proposed Scheme

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



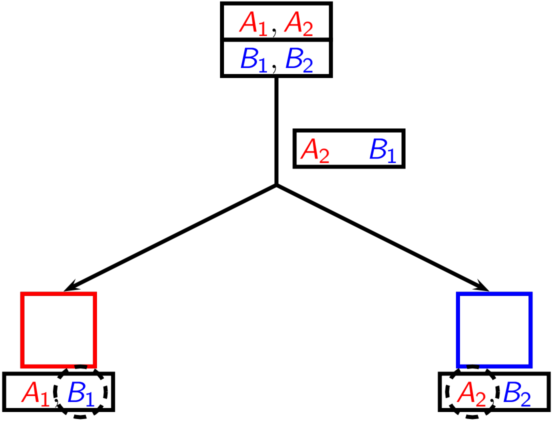
Proposed Scheme

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



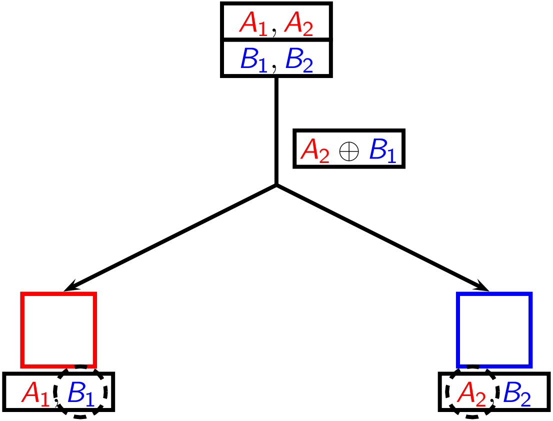
Proposed Scheme

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



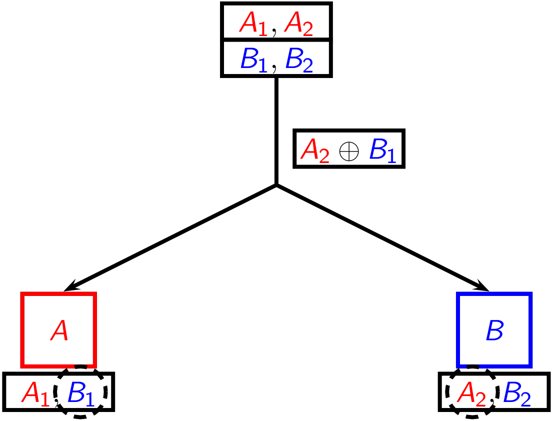
Proposed Scheme

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



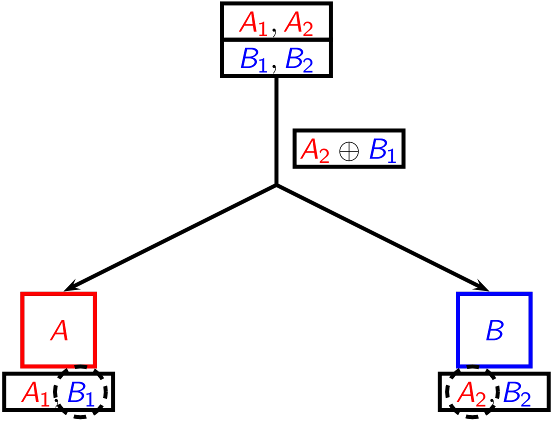
Proposed Scheme

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



Proposed Scheme

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

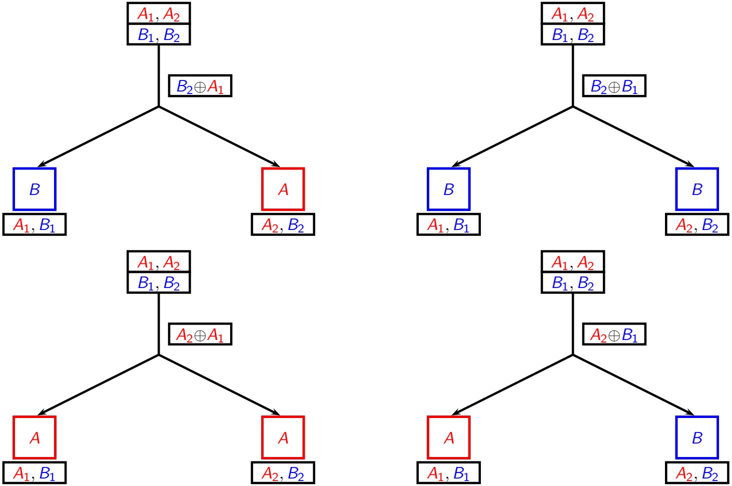


$\Rightarrow$ Different cache content at users

$\Rightarrow$ Multicast to 2 users with different demands

Proposed Scheme

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

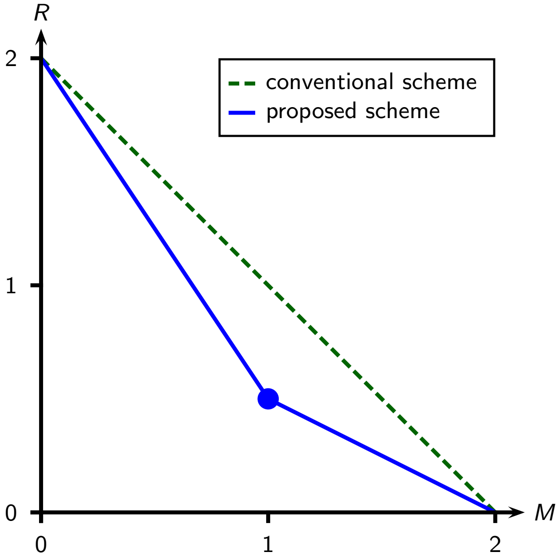


$\Rightarrow$ Works for all possible user requests

$\Rightarrow$ 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

$\Rightarrow$ 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.

$\Rightarrow$ Information-theoretic bound

Can We Do Better?

Theorem

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

$\Rightarrow$ Information-theoretic bound

$\Rightarrow$ 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.

$\Rightarrow$ Information-theoretic bound

$\Rightarrow$ Constant is independent of problem parameters N, K, M

$\Rightarrow$ 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
  - $\Rightarrow$ Multicast to users with different demands

Conclusions
A New Approach to Caching

- The main gain in caching is global
  $\Rightarrow$ Multicast to users with different demands
- Global cache size matters

Conclusions

A New Approach to Caching

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

Conclusions
A New Approach to Caching

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

Conclusions

A New Approach to Caching

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