Video Management and Resource Allocation for a Large-Scale VoD Cloud

Chang, Zhangyu

Supervised by **Prof. Gary Chan**

September 28, 2017

Contents

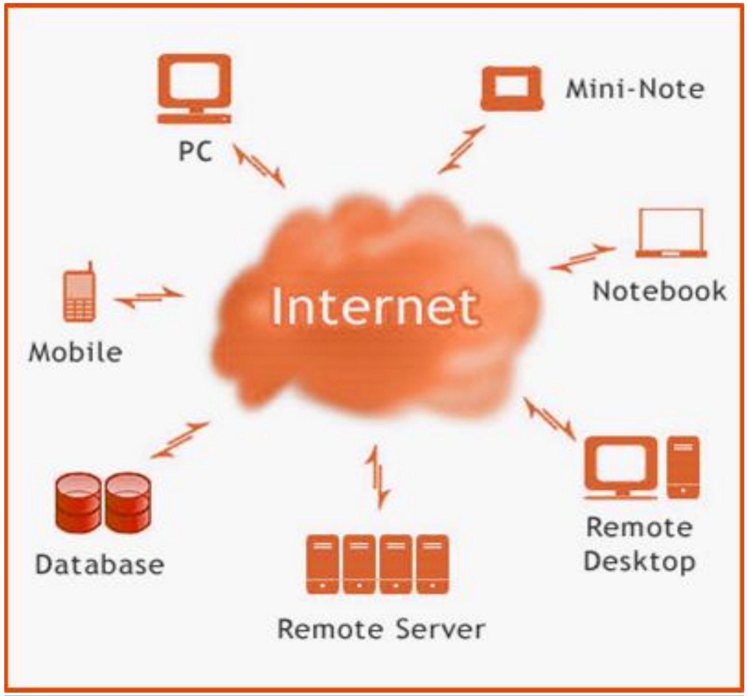
Video-on-Demand (VoD) Cloud

Video-on-Demand

- Essential Internet service for people's daily life nowadays
- Require huge amount of resource & network traffic

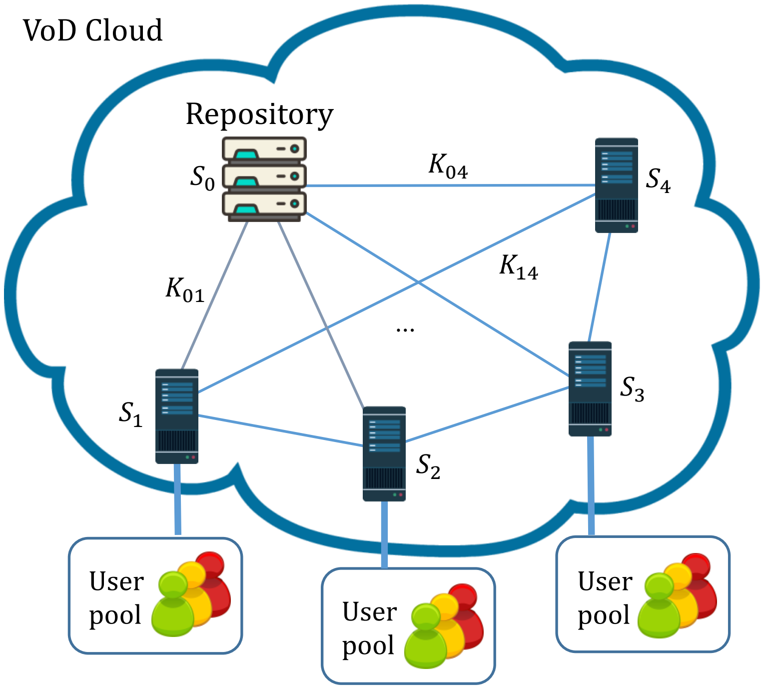
Cloud Computing

- Infrastructure as a service (IaaS)
- Reduce the cost on accessing distributed servers
- Reduce the risk of resource over-provisioning



A Typical VoD Cloud Service

Cloud Resources as Utility Service



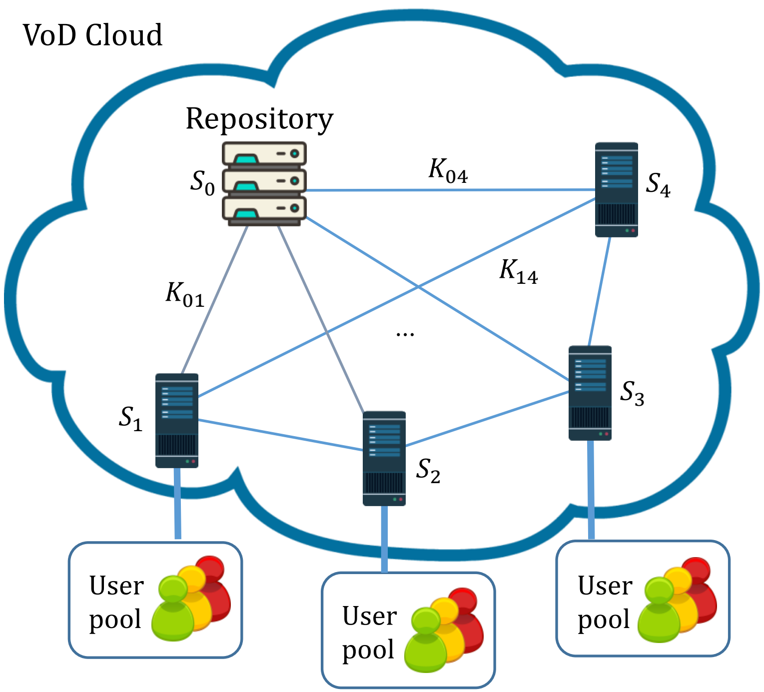
A distributed and cooperative cloud architecture for VoD service

- **Content Provider (CP)** can *rent* service from **Cloud Service Provider**
- **Content Provider** can dynamically *adjust* the resource deployment

Cloud service enables great **flexibility** on resource allocation:

- Scale up storage & streaming capacities timely
- Flexible resource allocation and provisioning
- Reduced maintenance cost

Deployment of a Distributed VoD Streaming Cloud



A distributed and cooperative cloud architecture for VoD service

	**Repository**: Complete video replication
	**Local cloud service**: Cluster of servers to serve the associated clients
	**Clients:** Geographically heterogeneous video popularities from clients

Geographic Heterogeneity
of Clients' Video Popularities

- **Local servers** may have partial video storage to save storage cost
- Reduce network load through co-operation among servers

Video Management & Resource Allocation

Video Management

- Video popularity: relatively stable and predictable in a Netflix-like VoD system
- Can be *planned* on a longer time scale (days)

Storage (content replication)

- What video to store at each server

Retrieval (server selection)

- Which servers to stream the missing video from

Resource Allocation

Server Cost

- Storage Capacity
- Processing Capacity

Cost due to the total storage and processing capacity at a server

Link Cost

- Link Capacity
- Bandwidth Utilization

Cost due to the bandwidth capacity reserved and data transmitted between pairs of servers to serve the misses

Deployment Cost vs. Quality-of-Service (QoS)

Deployment Cost

Server Cost

- Storage capacity
- Processing capacity

Link Cost

- Link capacity
- Bandwidth utilization

Quality-of-Service

Total Delay

- Due to server utilization
- Due to link utilization

Trade-off between Cost and Delay

- Satisfy the quality-of-service constraints
- Minimize total deployment cost

Bad Examples: 2 Extreme Scenarios

Full Replication
Full video storage among all local servers
+
• Minimum delay • No network cost
-
• Maximum storage cost • Cost much on cold video

Repository Only
Only video storage at the repository
+
• Minimum storage cost
-
• Maximum network cost • Huge end-to-end delay • Heavy load for repository

- Neither scenarios is efficient
- Both video management and resource allocation matters
- A joint optimization on comprehensive mode is required

Objective

Video Management and **Resource Allocation** are closely related

- Resource allocations is based on information of projected user request
- Content replication and retrieval are constrained by resource

Minimize total deployment cost

- **Server cost**: storage and processing capacity
- **Link cost**: link capacity and bandwidth utilization
- Geographically heterogeneous video popularity

Quality-of-service constraints

- Satisfactory level of end-to-end delay

Low algorithmic time complexity

- Accommodate a large video pool (in terms of video number $|V|$)

Approach

Relaxed Linear Programming

- Consider the video stored in each server as **continuous** variable
- Formulate and solve a **linear programming** (LP) problem

Quantization from Super Optimum

- Solution of the relaxed linear programming as the **super-optimum**
- **Randomized rounding** for video storage decision
- **Probabilistic** video retrieval decision
- **Resource allocation** decision based on QoE constraints

Video Clustering for Large Video Pool

- Group videos by **Spectral Clustering** to reduce the algorithmic complexity

Contributions

1	**Joint optimization formulation based on a comprehensive VoD cloud model**	**Video Management** • Server selection & content replication **Resource allocation** • Server cost (storage, processing) & link cost **Geographically heterogeneous popularity**
2	**RAVO: LP solution with quantization algorithm**	**Efficient optimization algorithm** • No extra encoding scheme • Applicable for current system • Proven optimality
3	**Video clustering method**	**Reduce the algorithmic time complexity** • Little compromise on deployment cost

Related Work

Fundamental difference:
Truly **JOINT** optimization algorithm

	Related Work	RAVO
Traditional resource allocation	• Based on heuristic approach • The optimality gap is not clear	• Discretized from LP solution • Closely optimal
Content Storage and Retrieval for VoD	• Need resource allocation result first • Rigid setting, less flexibility	• One-step offline algorithm for both resource allocation and content management • Easy to deploy in the real scenario
Current resource allocation for cloud service	• Assume full replication • Only consider bandwidth allocation	• Partial replication to lower the storage cost • Servers help each other to fully utilize the resource

Contents

Major Symbols Used

S	The set of servers (central and proxy servers)	Γ_{mn}	Average transmission rate from server m to n (bits/s)
V	The set of videos	U_m	Total upload rate of server m (bits/s)
$L^{(v)}$	Length of video v (seconds)	K_{mn}	Link capacity from server m to n (bits/s)
$P_m^{(v)}$	Access probability of video v at server m	Λ_m	Processing capacity of server m for remote streaming (bits/s)
$I_m^{(v)}$	Boolean variable indicating whether server m stores video v	C_{mn}^{N}	Link cost due to directed traffic from server m to n
H_m	Storage capacity of server m (bits)	C_m^{S}	Cost of server m
$R_{mn}^{(v)}$	Probability of streaming video v from server m to n	D_{mn}^{N}	Delay due to directed traffic from server m to n
μ_m	Request rate at server m (requests/second)	D_m^{S}	Delay due to upload streaming of server m

The Problem of Joint Optimization on Video Management and Resource Allocation

Server cost | Link cost

minimize $\sum_{m \in S} \mathbb{C}_m^{S}(H_m, \Lambda_m, U_m) + \sum_{m,n \in S} \mathbb{C}_{mn}^{N}(\Gamma_{mn}, K_{mn})$ → System deployment cost

H_m: **Storage**; Λ_m: **Processing Capacity**; K_{mn}: **Access** bandwidth (consumed)

Subject to

Storage: $I_m^{(v)} \in \{0,1\}, \forall m \in S, v \in V$ → Whether video v stored at m

Retrieval: $0 \leq R_{mn}^{(v)} \leq I_m^{(v)}, \forall m,n \in S, v \in V$ → Probability of video v retrieved from m to n

$\sum_{v \in V} I_m^{(v)} L^{(v)} \gamma^{(v)} \leq H_m, \quad \forall m \in S$ → Storage constraint at m

$\sum_{m \in S} R_{mn}^{(v)} = 1, \quad \forall n \in S, v \in V$ → A video shall be retrieved

$\Gamma_{mn} = \sum_{v \in V} p_n^{(v)} \varepsilon_n^{(v)} \mu_n R_{mn}^{(v)} L^{(v)} \gamma^{(v)}, \forall m,n \in S$ → Remote traffic

QoS: $\mathbb{D}_{mn}^{N}(\Gamma_{mn}, K_{mn}) + \mathbb{D}_m^{S}(U_m, \Lambda_m) \leq \bar{D}, \forall m,n \in S$ → Delay

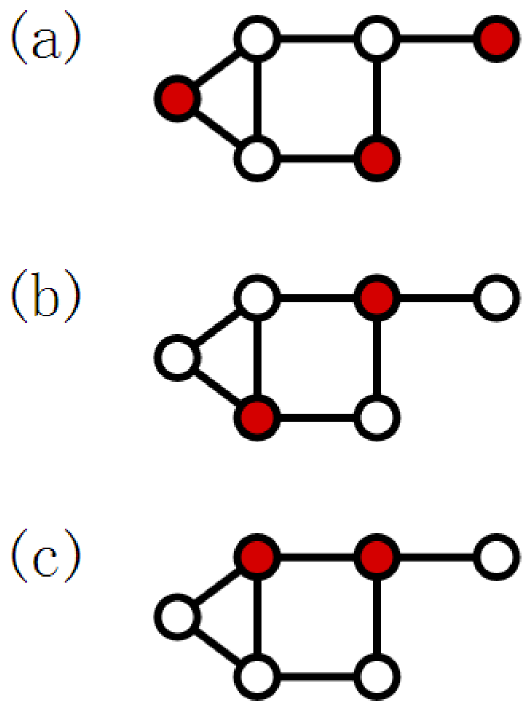
NP-hardness of Integer Programming: $I_m^{(v)} = \{0, 1\}$

The **dominating set problem**: (NP-complete)

- A **dominating set** for a graph $T = (S, E)$ is a subset D of V such that every vertex not in D is **adjacent** to at least one member of D.
- The **domination number** $\zeta(T)$ is the number of vertices in a **smallest** dominating set for T.
- The **dominating set problem** concerns testing whether $\zeta(T) \le J$ for a given graph T and input J.

The joint optimization is **NP-hard**

- The **dominating set problem** is reducible to our joint optimization problem.
- Considering that:
  - The VoD system has only one video
  - The storage cost for a replica is 1
  - No any other cost
- The servers that have the video replica form a **dominating set**.



Dominating sets (red vertices)

Contents

RAVO: Relaxing the Joint Formulation as a Linear Program and Quantization of the Solution

Step 1: Linear Program

Formulation Relaxation

- *Continuous* $\hat{I}_m^{(v)}$ $(0 \le \hat{I}_m^{(v)} \le 1)$
- $\mathbb{C}_m^{\mathrm{S}}(H_m, \Lambda_m, U_m)$, $\mathbb{C}_{mn}^{\mathrm{N}}(\Gamma_{mn}, K_{mn})$, $\mathbb{D}_{mn}^{\mathrm{N}}(\Gamma_{mn}, K_{mn})$ and $\mathbb{D}_m^{\mathrm{S}}(U_m, \Lambda_m)$ as *piecewise linear* function
- *Efficient algorithm* for solving linear programming

Solve LP for Super-optimum

- Video storage: $\hat{I}_m^{(v)}$
- Video retrieval: $\hat{R}_{mn}^{(v)}$

Step 2: Quantization

Video Management

- *Randomized round* $\hat{I}_m^{(v)}$ to get $I_m^{(v)}$
- Request from the *repository* if no other proxy server can help
- Otherwise we obtain $\forall m, n \in S$

$$R_{mn}^{(v)} = \begin{cases} 0, & \text{if } I_m^{(v)} = 0; \\ \dfrac{\hat{R}_{mn}^{(v)}}{\sum_{m \in S} I_m^{(v)} \hat{R}_{mn}^{(v)}}, & \text{if } I_m^{(v)} = 0. \end{cases}$$

Resource Allocation

- Server storage capacity as $H_m = \sum_{v \in V} I_m^{(v)} L^{(v)} \gamma^{(v)}, \forall m \in S$
- Get Γ_{mn} and U_m from $I_m^{(v)}$ and $R_{mn}^{(v)}$
- Put Γ_{mn} and U_m to equation $\mathbb{D}_m^{\mathrm{S}}(U_m, \Lambda_m) = D_m^{\mathrm{S}}, \ \forall m \in S$; $\mathbb{D}_{mn}^{\mathrm{N}}(\Gamma_{mn}, K_{mn}) = D_{mn}^{\mathrm{N}}, \ \forall m, n \in S$, and solve them to get Λ_m and K_{mn}

Algorithmic Complexity

LP	LP solver has constant expected iterations and $O(N^3)$ for each iteration (N is the number of variables)

$O(|S|^6|V|^3)$ **Dominate**

Discretize Video Storage	**Discretize Video Retrieval**
$O(|S||V|)$	$O(|S|^2|V|)$

|S|	Number of servers
|V|	Number of videos

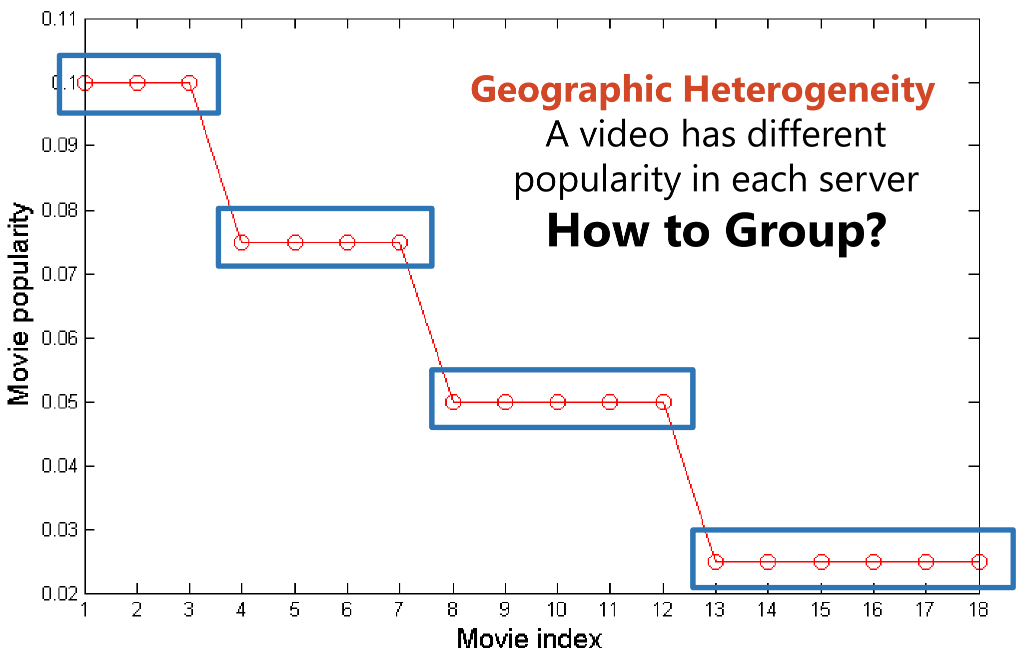
What if

$|V|$ is LARGE?

Contents

Observation on Concurrency Density

- **Concurrency density** ($b_m^{(v)} = p_m^{(v)} \varepsilon_m^{(v)}$) gives the per-storage user concurrency of a video
- Videos with *same* concurrency density result in the *same* per-bit deployment cost
- Video groups with the same concurrency density will NOT change the result of the linear programming, but the number of parameters (problem complexity) is smaller.
- Minimize



Spectral Clustering for Video Group

- Treat the concurrency density of a video v as an $|S|$ dimensional vector, namely $\boldsymbol{b}^{(v)} = (b_1^{(v)}, b_2^{(v)}, \dots, b_{|S|}^{(v)})$.
- Minimize

$$\arg_{g_i} \sum_{i=1}^{|G|} \sum_{v \in g_i} \left\| \boldsymbol{b}^{(v)} - \widetilde{\boldsymbol{b}}^{(g_i)} \right\|^2$$

- $\widetilde{\boldsymbol{b}}^{(g_i)}$ is the mean concurrency density of group g_i
- Resulting group size may not be the same
- Use spectral clustering to solve multi-dimensional *K-means*
- After solving the linear program, use *rarest first* for video placement $I_m^{(v)}$ and $\hat{R}_{mn}^{(v)} = \hat{R}_{mn}^{(g_i)}, \forall\, v \epsilon g_i$
- Then use method in RAVO for further parameter quantization

Algorithmic Complexity Reduction

LP	LP solver has constant expected iterations and $O(N^3)$ for each iteration (N is the number of variables)

|S|	Number of servers
|V|	Number of videos
|G|	Number of groups

$O(|S|^6|G|^3)$

$O(|S|^6)$ is a huge factor

Quantization on $I_m^{(v)}$ — $O(|S||V|)$

Quantization on $R_{mn}^{(v)}$ — $O(|S|^2|V|)$

Spectral Clustering — $O(|S||V|)$

Reducing complexity by $O(|V|^2)$

Contents

Simulation Environment

Video popularity

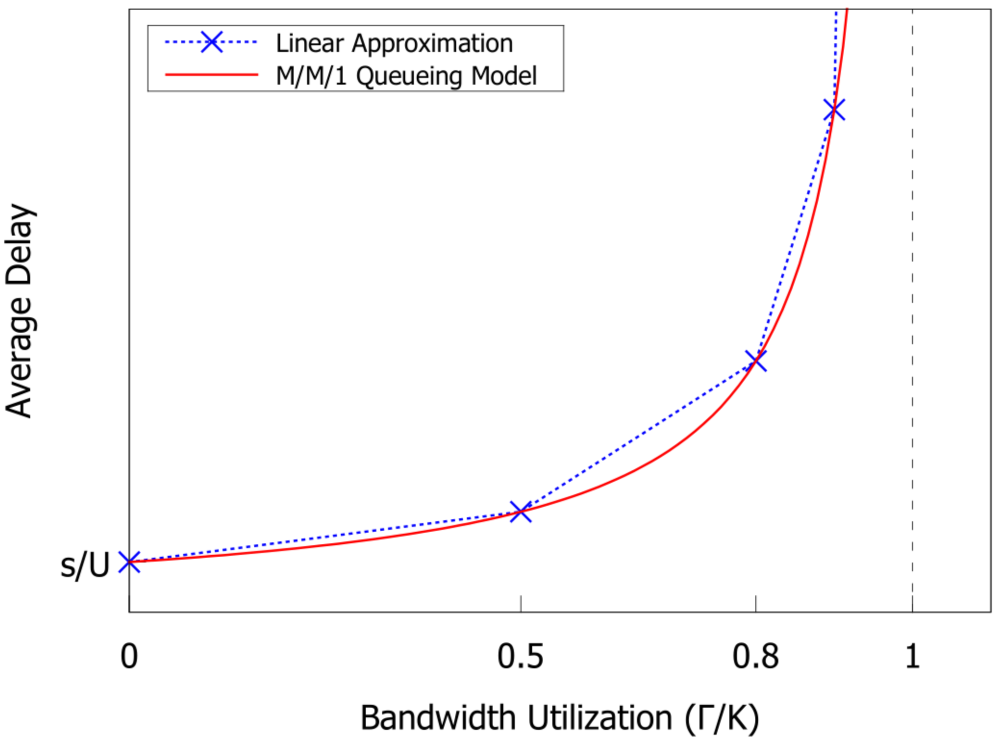
- *Zipf's* distribution: $f(i) \propto 1/i^z$
- Geographic heterogeneity
- Partially reshuffle video rank
- Trace driven based on *real data*

Cost functions

- *Proportional* to resource used
- Server cost: $C_m^{\mathrm{S}} = \sigma_m H_m + c_m \Lambda_m$
- Link cost: $C_{mn}^{\mathrm{N}} = c_{mn} K_{mn}$

Delay Function

- *M\M\1* queueing model
- *Piece-wise linear* approximation



Performance Metrics & Comparison Schemes

Performance Metrics

Total cost & components

- Server storage cost
- Server processing cost
- Link cost

Delay

- Caused by links
- Caused by servers

Running time

- Algorithmic running time

Comparison Schemes

iGreedy with optimal resource allocation

- Consider local popularity
- No cooperative replication

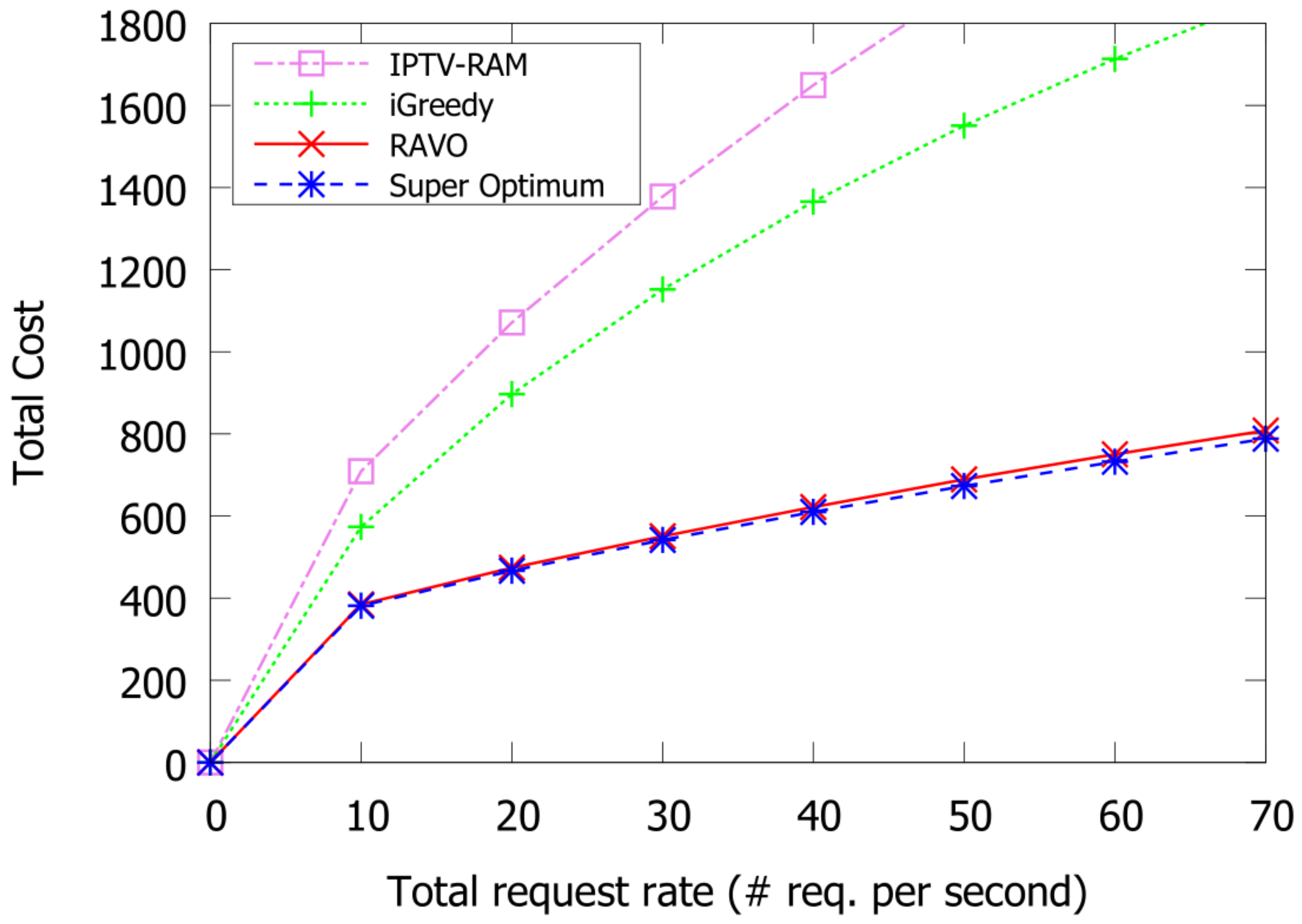
IPTV-RAM with optimal content management

- 2 video categories based on global popularity

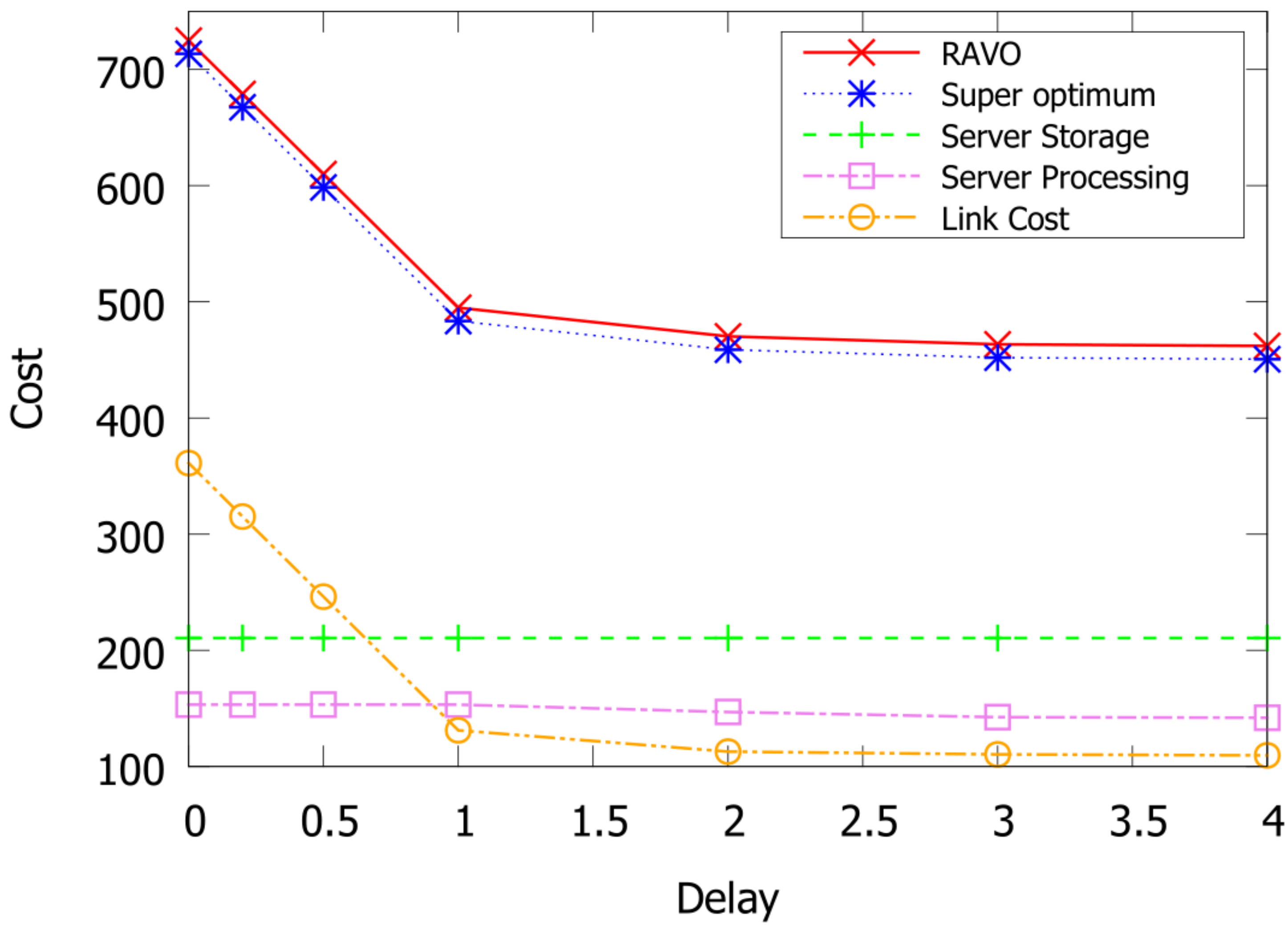
Super-optimal

- LP solution before quantization

Close to optimal performance (Cost versus Request Rate)

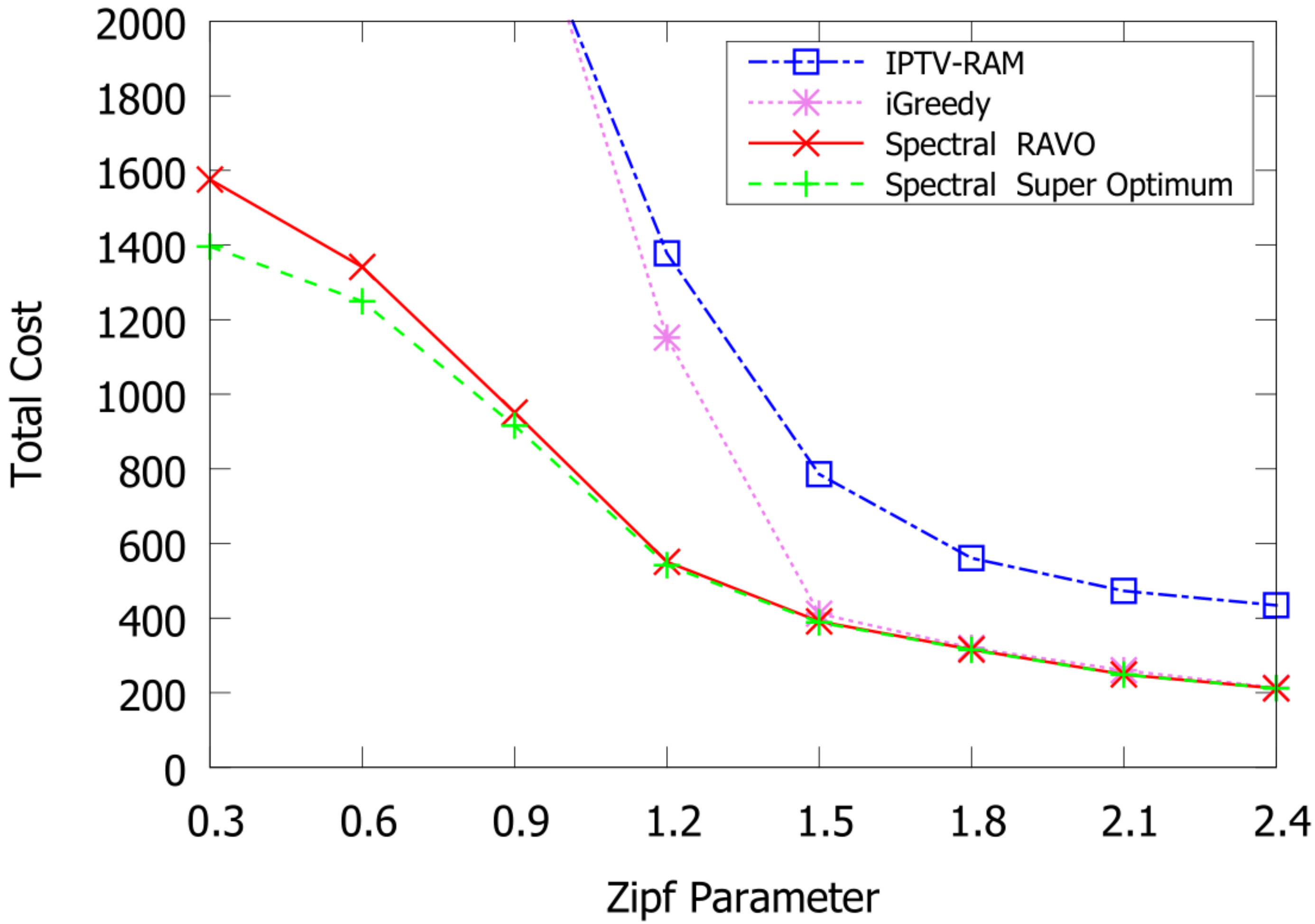


Close to optimal performance (Cost versus Delay Requirement)



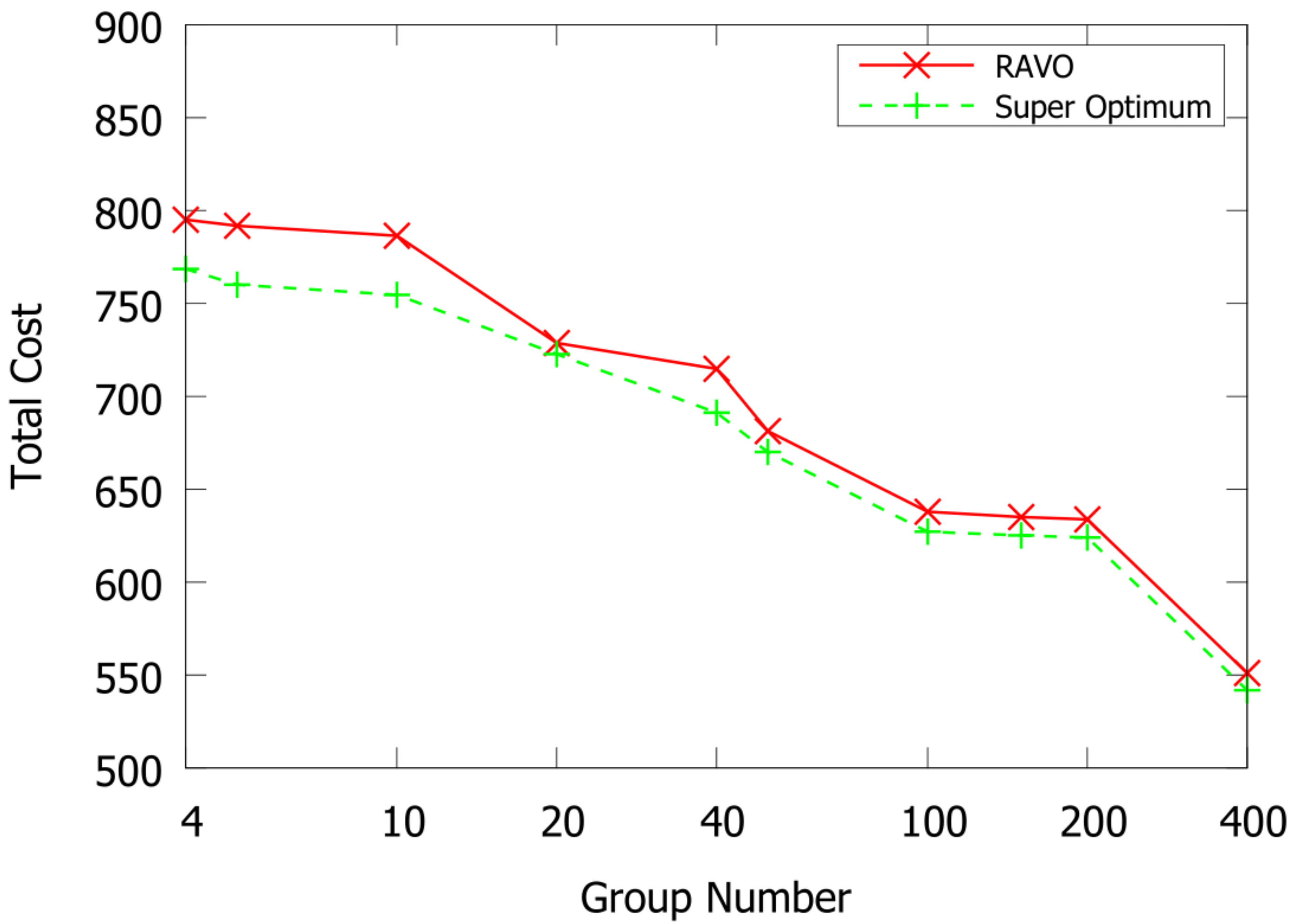
Effective Clustering Method (Cost versus *Zipf* Parameter)

- Skewness of video popularity has greater impact
- RAVO can better utilize cheap resource



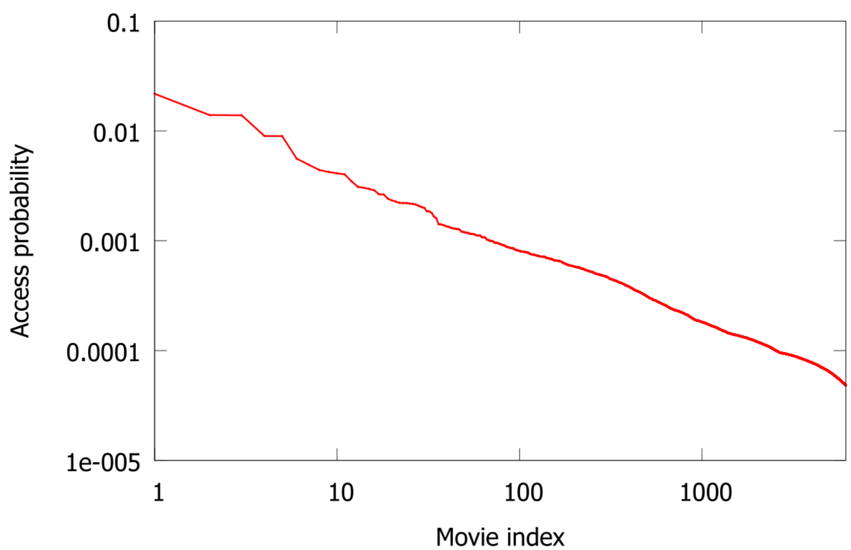
Effective Clustering Method (Cost versus Group Number)

- Longer running time for better optimality

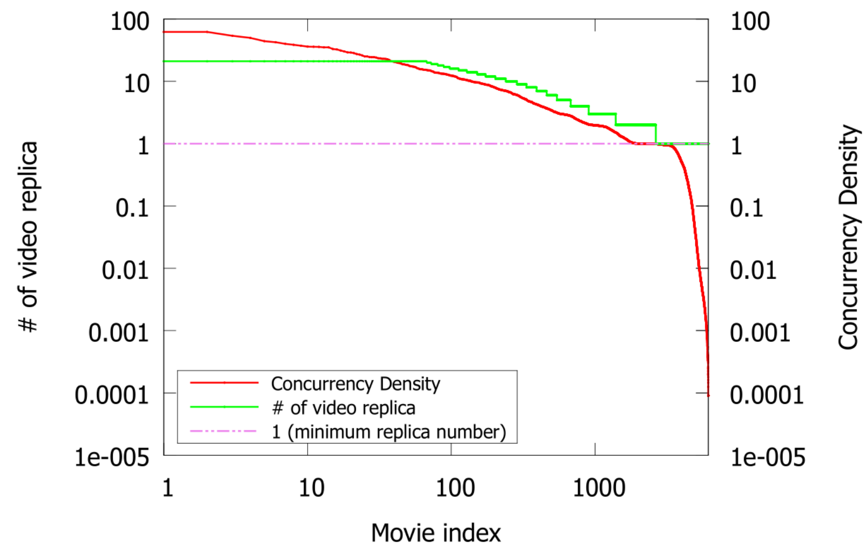


Trace-driven Simulation: Video Popularity

Movie access probability in descending order



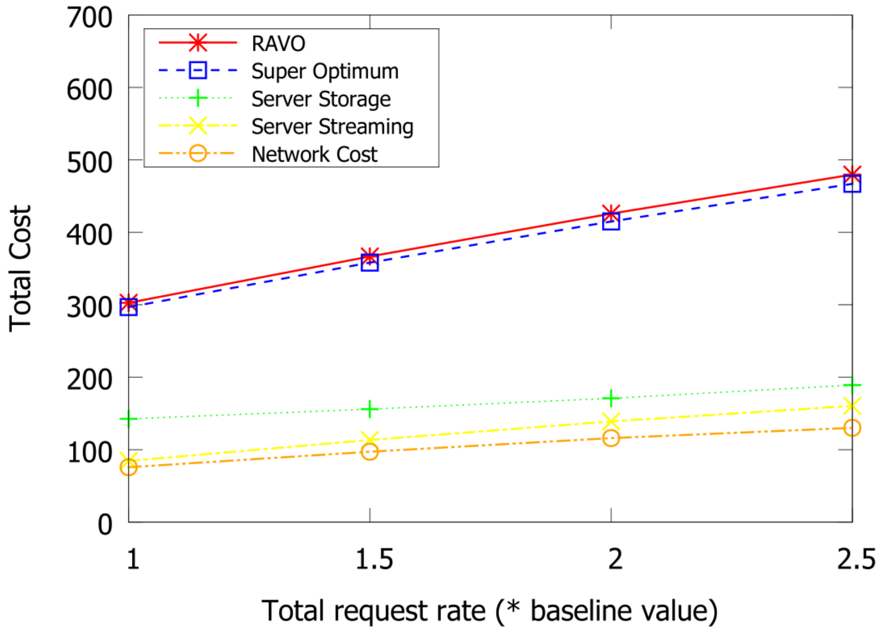
Concurrency density and replica number versus movie index



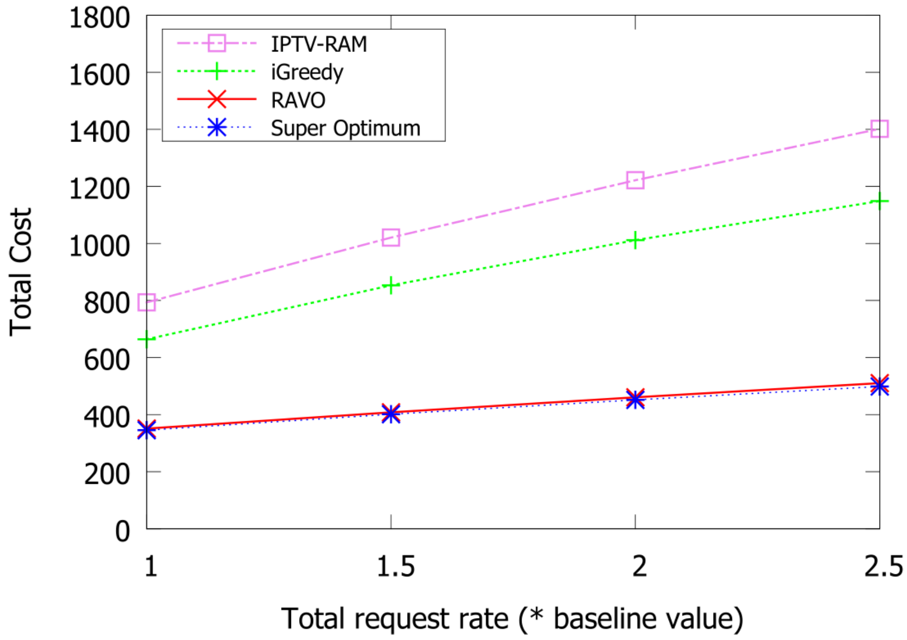
- The video access probability follows *Zipf*'s distribution
- Videos with higher concurrency density have more replicas on the cloud

Trace-driven Simulation: Performance

Deployment cost given different request rate



Deployment and component cost given different request rate



- RAVO outperform the comparison schemes with large margin
- Storage cost increases slower than the other components due to *cold video*

Contents

Conclusion

Comprehensive **Model for VoD Cloud**	• Minimize total cost: Server + Link • Video management & Resource allocation • Quality-of-service (delay) constraints • Geographic heterogeneity
RAVO **Efficient Algorithm**	• LP formulation → super optimum • Randomized rounding • Probabilistic video retrieval
Video Grouping **Spectral Clustering**	• Efficient computation • Little performance Loss • Significant time complexity reduction • Geographic heterogeneity
Extensive **Simulation Study**	• Close-to-optimum performance • Outperform the comparison scheme • Trace-driven simulation based on real data

Thank You!

Any Questions?