

Leveraging the Clouds for improving P2P Content Distribution Networks Performance

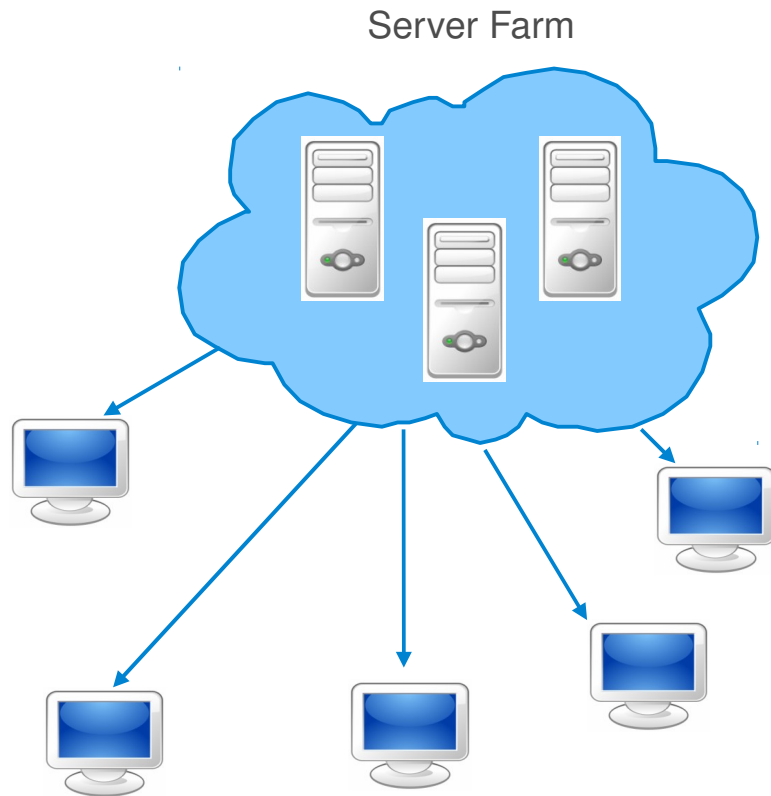
Amir H. Payberah
amir@sics.se



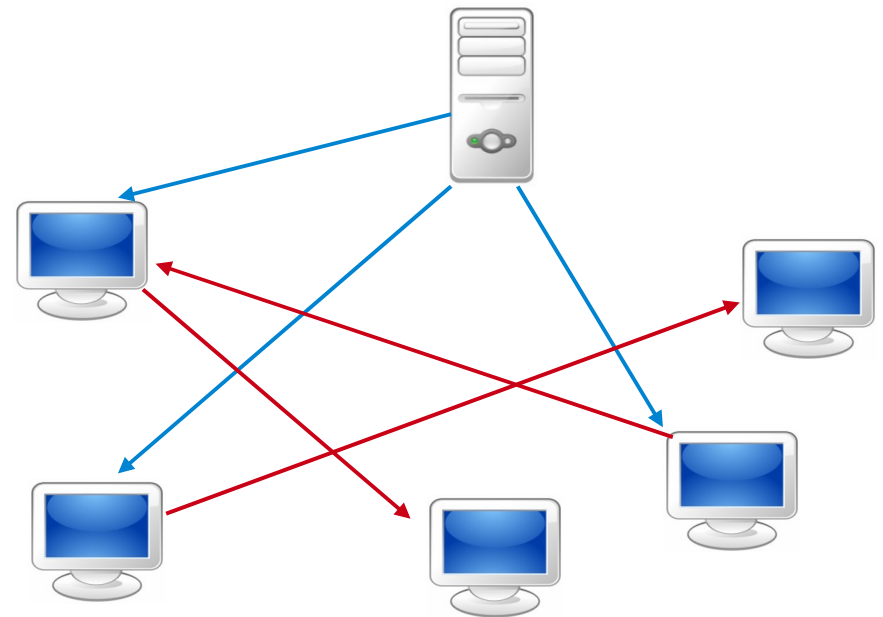
Big Picture

Big Picture

Client-Server



Peer-to-Peer



Big Picture

- How to leverage the cloud computing for improving the performance of P2P content delivery networks.

CloudMedia

How to configure the content provider utility to meet the highly dynamic demands of large-scale applications at a modest cost.

CloudAngels

How to use cloud resources in a content provider to accelerate peer-assisted content distribution.

CloudMedia

**CloudMedia: When Cloud on Demand Meets Video on Demand
(ICDCS'11)**

Boston University

Motivation

- How can a content provider learn the **dynamic demands** from users and **relay them to the cloud** service accordingly?
- How can a content provider can effectively **configure the cloud utility** to achieve the **best performance** at a **reduced cost**.

Contribution

- Queuing network modelling
 - Introducing a **queuing network model** to characterize the **dynamic viewing behaviour** of **VoD users**, and derive the **server capacity** needed to support smooth playback.
- Dynamic cloud provisioning algorithm
 - Formulating the **VM provisioning and storage rental** as two **optimization problems**, and proposing the algorithms to dynamically configure cloud resources.
- CloudMedia

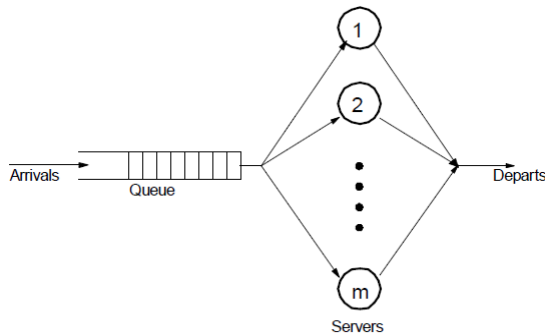
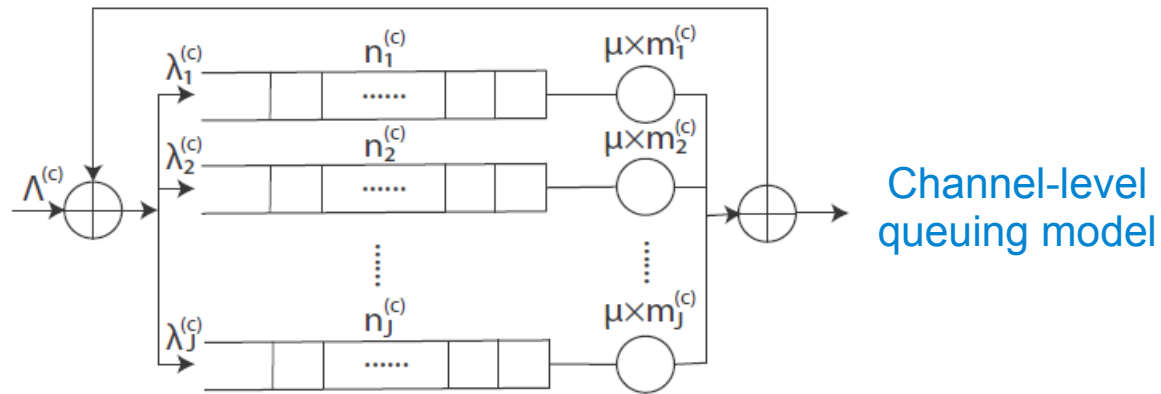
Contribution

- Queuing network modelling
 - Introducing a **queuing network model** to characterize the **dynamic viewing behaviour** of **VoD users**, and derive the **server capacity** needed to support smooth playback.
- Dynamic cloud provisioning algorithm
 - Formulating the VM provisioning and storage rental as two optimization problems, and proposing the algorithms to dynamically configure cloud resources.
- CloudMedia

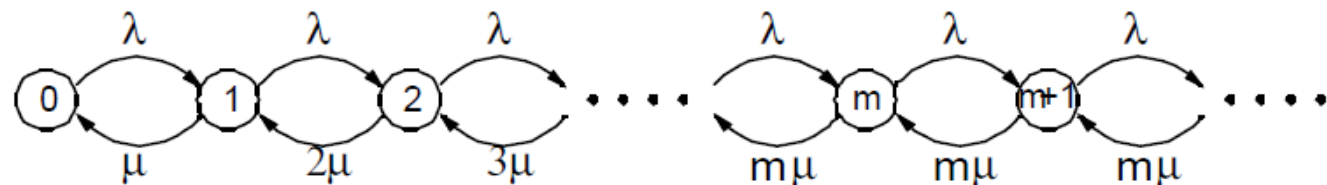
Queuing Network Modelling

- Jackson network

- A network of queues where the arrivals at each queue form a Poisson process, and the job service times are exponentially distributed.
- The video is divided into a number of chunks.
- One M/M/m queue model for each chunk.



Reminder (M/M/m)



Upload Bandwidth Estimation

- $\Delta_i^{(c)}$:
 - Client-Server: $R * m_i$
 - P2P: $R * m_i - L_i$
- $\Delta_i^{(c)}$: the total upload BW to serve chunk i in channel c in the server side.
- R : the allocated BW of each VM.
- m_i = the number of servers (m) in the queue of chunk i .
- L_i = the BW contributed to upload chunk i from the P2P overlay.

Contribution

- Queuing network modelling
 - Introducing a queuing network model to characterize the dynamic viewing behaviour of VoD users, and derive the server capacity needed to support smooth playback.
- Dynamic cloud provisioning algorithm
 - Formulating the **VM provisioning and storage rental** as two **optimization problems**, and proposing the algorithms to dynamically configure cloud resources.
- CloudMedia

Optimal Storage Rental

- Formulating as an optimization problem:
- Objective function:
 - Maximizing the aggregate performance for retrieving all chunks in all videos from the storage.
- Subject to:
 - Keep only one copy of each chunk in the storage.
 - The total amount of the chunks on each storage shouldn't exceed the storage capacity.
 - All the cost should be less than our budget.

Optimal Storage Rental - Heuristic

- We seek to place the most popular chunks on the storage at the highest performance level with the most economic budget expenditure.
- Heuristic:
 - Sort all chunks in all channels in decreasing order of upload BW the cloud needs to supply.
 - Sort the storages in decreasing order of the marginal utility per unit cost.
 - Starting with the chunk with the highest demand, we store it in the best storage, as long as the storage is not full, or move on to the second best cluster otherwise.
 - This process repeats for all the chunks in the ordered list, as long as the total storage budget spent does not exceed the budget .

Optimal VM Rental

- Formulating as an optimization problem:
- Objective function:
 - Maximizing the aggregate performance for serving all chunks on all videos from the VMs requested.
- Subject to:
 - Total upload BW of VMs requested for each chunk should be sufficient to serve the demand for the chunk.
 - The total number of requested VMs should be less than the maximum number of available VMs.
 - All the cost should be less than our budget.

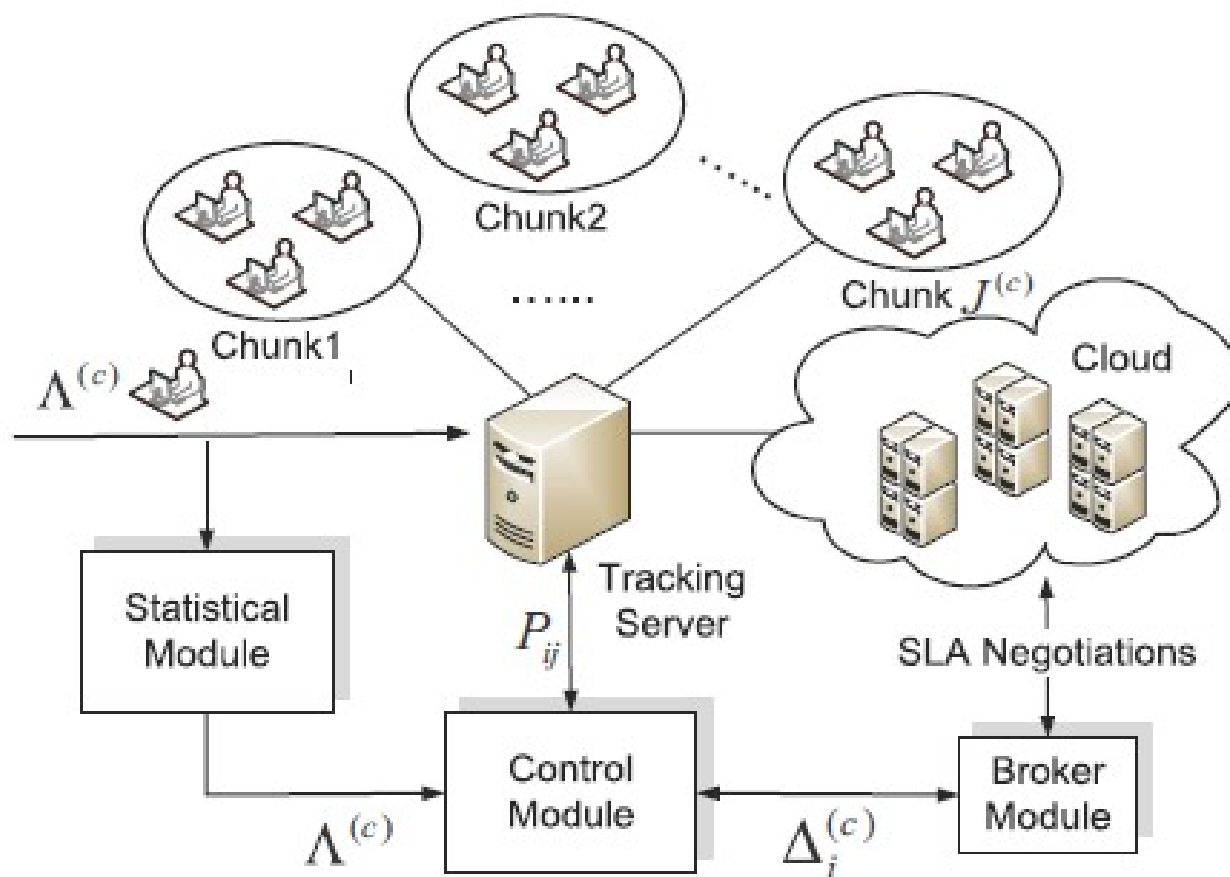
Optimal VM Rental - Heuristic

- We seek to maximally place the chunks on the VM with the best configuration at the modest budget.
- Heuristic:
 - Sort the VMs in decreasing order of the marginal utility per unit cost.
 - For any chunk i in channel c , the total number of VMs it needs is $\Delta_i^{(c)}/R$ (total BW demand on the cloud for chunk i / BW of a VM).
 - Allocate as many VMs as possible to serve each chunk demand from the best VM, if it is still available, or move on the second best VM otherwise.
 - This process repeats for all the chunks, as long as the total VM rental is not exceed .

Contribution

- Queuing network modelling
 - Introducing a queuing network model to characterize the dynamic viewing behaviour of VoD users, and derive the server capacity needed to support smooth playback.
- Dynamic cloud provisioning algorithm
 - Formulating the VM provisioning and storage rental as two optimization problems, and proposing the algorithms to dynamically configure cloud resources.
- CloudMedia

CloudMedia Architecture



CloudAngles

**Angels In the Cloud
(CLOUD'11)**

The University of Hong Kong

Motivation

- How to minimize the time it takes to distribute a chunk of a content to all the peers in a swarm?
- How to provide **minimum distribution time (MDT)** for **bulk-synchronous** content distribution?

Contribution

- Presenting an **analytical model** for the **lower bound of MDT** of a swarm in the presence of angels.
- Using the **on-demand deployment of cloud resources** (**angels**) to alleviate the swarm upload capacity bottleneck.
- CloudAngels

Contribution

- Presenting an **analytical model** for the **lower bound of MDT** of a swarm in the presence of angels.
- Using the on-demand deployment of cloud resources (angels) to alleviate the swarm upload capacity bottleneck.
- CloudAngels

MDT Problem Statement

- Formulating as an optimization problem:
- Objective function:
 - Minimizing the maximum time of delivering of the chunk i to all the peers in the swarm.
- Subject to:
 - Total consumed upload BW of the provider/each peer/each angel shouldn't exceed their maximum upload capacity.
 - Total received content in each peer from the provider, an angel and other peers shouldn't exceed its download capacity.
 - Total received content in each angel from the provider shouldn't exceed its download capacity.

MDT Problem Statement - Result

- In the presence of angels, under a fluid uplink-sharing model, the MDT has a lower bound given by:

$$T_{\min} \geq F / \min\{A, B, C\}$$

- **F**: file size
- **A**: upload capacity of the provider.
- **B**: download capacity of the slowest peer.
- **C**: aggregate upload capacity of the swarm.

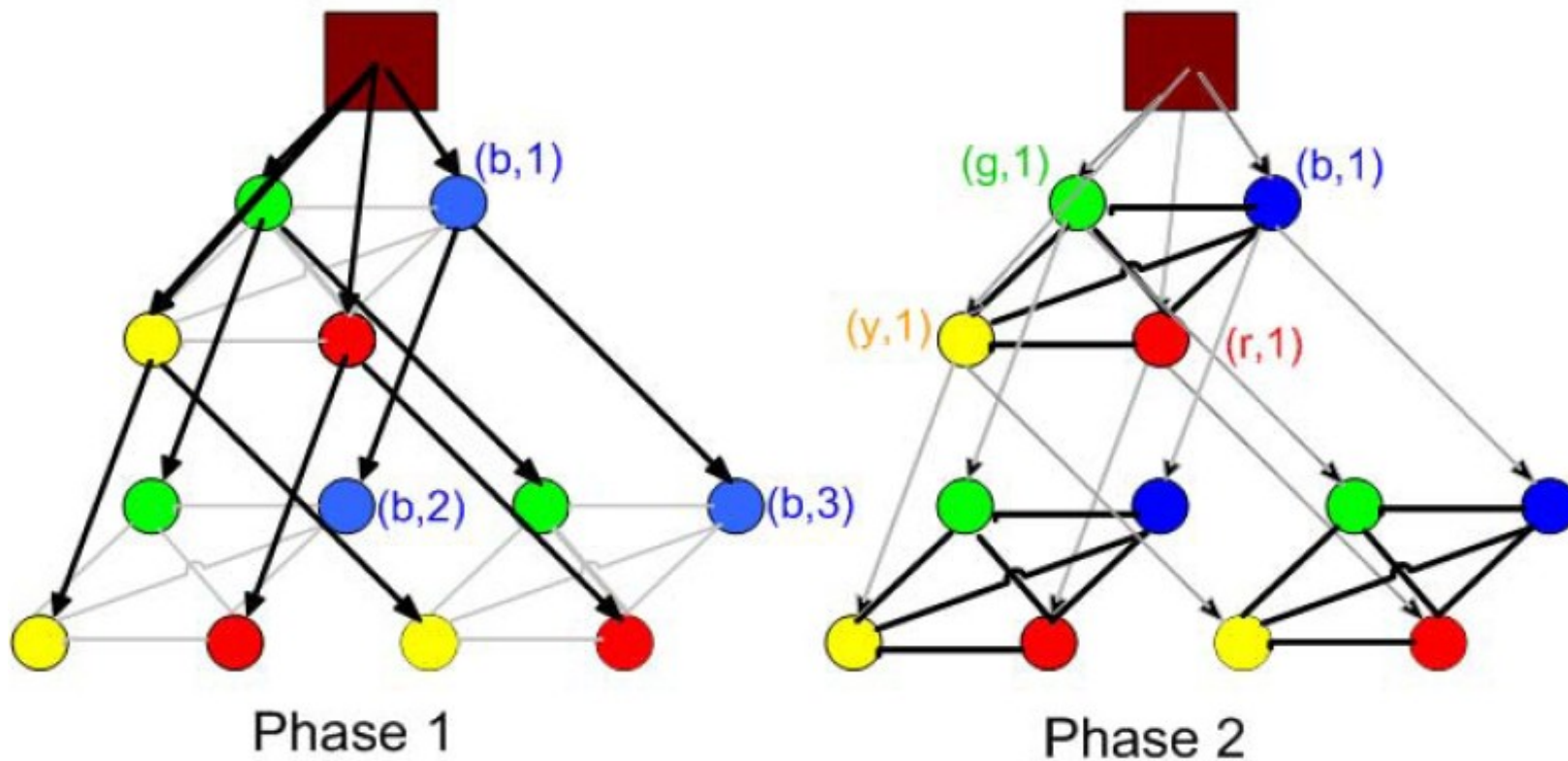
MDT Problem Statement - Result

- When the **bottleneck** lies in the aggregate upload capacity of the swarm and not in the provider's upload capacity, adding upload capacity to the swarm through the use of angels will necessarily improve MDT.
- **But ...**
- If angels download more than $1/(\text{num. of peers})$ of what they upload, they will not be able to upload the excess amount to clients.

Contribution

- Presenting an analytical model for the lower bound of MDT of a swarm in the presence of angels.
- Using the **on-demand deployment of cloud resources (angels)** to alleviate the swarm upload capacity bottleneck.
- CloudAngels

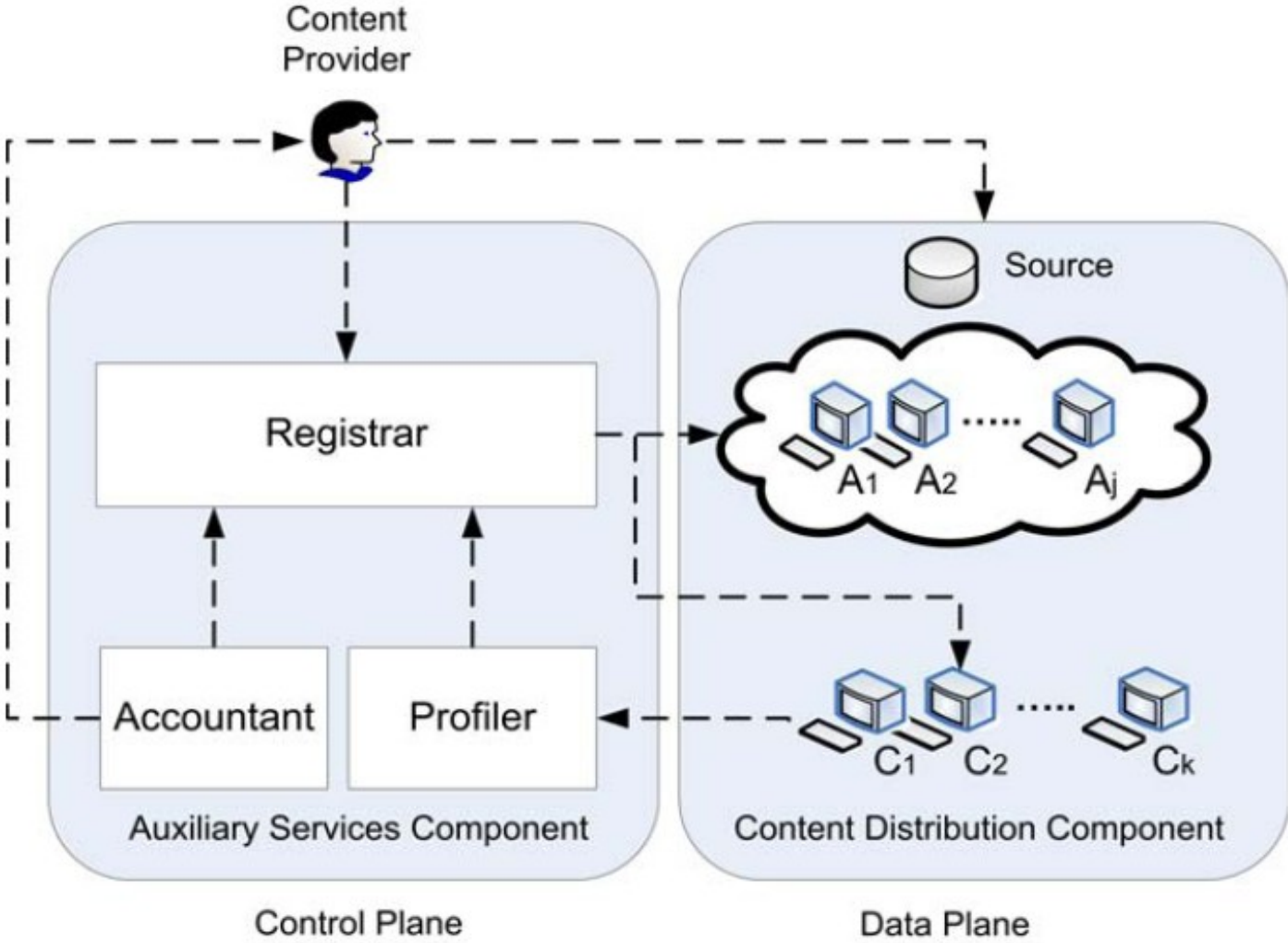
The Group Tree Coordinated Swarming Strategy



Contribution

- Presenting an analytical model for the lower bound of MDT of a swarm in the presence of angels.
- Using the on-demand deployment of cloud resources (angels) to alleviate the swarm upload capacity bottleneck.
- CloudAngels

CloudAngels Architecture



Conclusion

Conclusion

- **CloudMedia** and **CloudAngles** are two systems that leverage the cloud computing power for increasing their performance.
- **CloudMedia:**
 - Providing smooth playback of video on peers.
 - Providing an elastic source.
- **CloudAngles**
 - Minimizing the content delivery time to all the peers.
 - Adding/removing angles to/from the swarm.

Future Work

- The decision points in these system have a **global view** → **decentralized model with partial view**
- The current solutions are for **VoD** and **bulk-synchronization** → **using the idea for our current live solutions: sepidar/glive.**

Question?