CLive: Cloud-assisted P2P Live Streaming

Amir H. Payberah¹, Hanna Kavalionak², Vimalkumar Kumareshan²
Alberto Montresor², Seif Haridi¹
{amir,seif}@sics.se¹, {name.family}@unitn.it²
The Problem
Existing P2P Streaming Solutions

Client-Server

Server Farm

Peer-to-Peer
Problem?

- Bottlenecks in P2P video streaming systems: upload bandwidth

- A potential solution: P2P network is assisted by a cloud computing.
Problem?

- **Bottlenecks** in P2P video streaming systems: **upload bandwidth**

- A potential solution: **P2P network is assisted by a cloud computing.**

Cloud is not free
Problem?

- **Bottlenecks** in P2P video streaming systems: *upload bandwidth*

- A potential solution: **P2P network is assisted by a cloud computing.**
P2P vs. Cloud

• P2P pros and cons
  - P2P resources are cheap
  - Churn may compromise availability

• Cloud pros and cons
  - Superior availability
  - Cloud resources are not free
If You Cannot Beat Them, …? ;)

- The **cloud** as a support group for **P2P**.
- Reduce the number of (costly) **cloud interactions** as much as possible.
If You Cannot Beat Them, …? ;)

- The cloud as a support group for P2P.
- Reduce the number of (costly) cloud interactions as much as possible.

The problem to be solved becomes minimizing the economical cost, provided that a set of constraints on QoS is satisfied.
Enhanced Model
CLive Components

- P2P streaming overlay (swarm)
- Media source
- Passive Helpers
- Active Helpers
- Management component
CLive Components: P2P Streaming Component

- We assume that nodes in the system use a mesh-pull model for data distribution.

- Nodes periodically send their buffer maps to their neighbours.

- The other nodes pull the required chunks from those nodes who own the chunks.
**CLive Components: Media Source**

- A *media source* is a node that generates data chunks and **pushes** them to the swarm.
**CLive Components: Passive Helper**

- **PH** is a passive element that plays the role of a data storage, e.g., Amazon S3.

- The source pushes chunks to PH, as they are generated.

- The swarm nodes pull the missed chunks from it.

- We assume that a PH can serve as many requests as it receives.
CLlive Components: Active Helper

- AHs are active elements, e.g., Amazon EC2, that cooperate with other swarm nodes to accelerate the data dissemination.

- The source pushes chunks to AH, as they are generated.

- AHs forwards chunks to the swarm and other AH, as they are received.
CLive Components: Management

- Participates in a **gossip algorithm** to estimate the available resources in the system.

- **Adds/removes AHs** to/from the system, based on the estimation.
Two Main Questions in CLive?

- How to estimate the extra load in the overlay?
- How to relay the load to cloud with a minimum cost?
Two Main Questions in CLive?

- How to estimate the extra load in the overlay?
- How to relay the load to cloud with a minimum cost?
Main Idea

- **Infected nodes**: the number of nodes that can be served with the existing resources in the system.

- **load = swarm size – infected nodes**
The Swarm Size

- The swarm size estimation is easy: gossip-based aggregation
The Infected Nodes

- It is shown that each streamed chunk through mesh overlays follows a tree-based diffusion pattern [1].

- To compute the number of infected nodes, we model a diffusion tree for a chunk and estimate the number of nodes in that tree.

Estimate the Number of Nodes in a Tree

- Estimate the tree depth

- Estimate the fan-out distribution
Estimate the Tree Depth (1/2)

- **Last Chance Window (LCW):** A predefined number of data chunks that must be buffered ahead of the playback point.

- The missed chunks are *downloaded directly from the PH*, when the number of data chunks that are buffered ahead is *less than LCW*.

- However, if the **LCW buffer is full**, nodes pull the required chunks from the swarm.
Estimate the Tree Depth (2/2)

- $T_{\text{delay}}$: maximum acceptable latency.

- $T_{\text{life}} = T_{\text{delay}} - T_{\text{lcw}}$

- If a node cannot receive a chunk in $T_{\text{life}}$, it pulls that chunk from PH.

- The tree depth
  - $D_{\text{max}} = T_{\text{life}} / T_{\text{d}}$
  - $T_{\text{d}}$: average latency among the peers
Estimate the Fan-out Distribution

- The upload bandwidth distribution
  - Adam2
  - Gossip-based aggregation
Estimate the Number of Nodes in a Tree

- Estimate the tree depth
- Estimate the fan-out distribution

$$n_{\text{tree}}$$
The Number of Infected Nodes

- \( N_{\text{inf}} = (\text{the number of trees}) \cdot n_{\text{tree}} \)

- The number of trees: source fanout + AHs fanout
Two Main Questions in CLive?

- How to estimate the extra load in the overlay?

- How to relay the load to cloud with a minimum cost?
AH Cost

- **AH cost** in one round: $C_{ah} = C_{vm} + m.C_{chunk}$
  - $C_{vm}$: virtual machine cost
  - $C_{chunk}$: chunk transfer cost
  - $m$: number of chunks that one AH upload per round
PH Cost

- **PH cost in one round**: \( C_{ph} = C_{storage} + r \cdot (C_{chunk} + C_{req}) \)

  - \( C_{storage} \): the storage cost
  - \( C_{chunk} \): chuck transfer cost
  - \( C_{req} \): chuck request
  - \( r \): the number of retrieved chunks from PH in one round
We define $\delta$ as the number of peers that is economically reasonable to serve with PH utilization instead to run an additional AH for them.

$\delta = \frac{C_{ah}}{C_{ph}}$
Manage the Cloud Resources

- If $\text{load} > \partial$: add AH

- If $\text{load} < \partial - H$: remove AH
  - $H$: number of peers served by one AH.

- Otherwise don't change AHs.
Experiments
Different LCW size

**Playback continuity**

- Percentage of the peers (%)
- Time (s)
- LCW=40
- LCW=30
- LCW=20
- LCW=10
- LCW=0

**Playback latency**

- Avg. playback latency (s)
- Time (s)
- LCW=0
- LCW=10
- LCW=20
- LCW=30
- LCW=40
PH Load
PH Load in Different Churn Rates
Slot Distribution Estimation

[Graphs showing slot distribution estimation error over rounds and churn rate]
Cost

Number of AHs

PH load

Total cost
Cost – Real Trace Slot Distribution

Number of AHs

PH load

Total cost
Summary
Any Questions?