A Highly Efficient Cloud-Based Architecture for Large-Scale STB Event Processing

Bin Cao, Jianwei Yin, et al.
Zhejiang University, China
iPanel.TV Inc. China
Set-top Box (STB)

- **STB device:**
  - Enable TV set to become a user interface to the internet
  - Enable TV set to receive and decode signals
- **DTV and IPTV**
  - Video on demand (VOD)
  - Shopping
  - E-government accessed via a TV set
Outline

- Introduction
- Scenario and Architecture
- Implementation discussion
- Experimental evaluation
- Conclusion and future work
Outline

- Introduction
- Scenario and Architecture
- Implementation discussion
- Experimental evaluation
- Conclusion and future work
Massive STB events are issued

- STB (Set-Top Box) device: DTV IPTV
  - Replacing traditional TV in many countries
  - Varieties of interactive services
  - Generating events (by remote control)
- Massive STB users
- Massive STB events

How to process these events in real-time?
Cloud computing

- Cloud computing:
  - Resources are provided on demand
  - Elastically scaled up and down
  - Drastic reduction in the cost of IT

- Connecting the STB devices to the cloud!

- After entering the cloud…
  - “Big Data” techniques
  - “Instant” response
Cloud computing

- Move data to the cloud:
  - Discovering STB devices and their event feeds.
  - Handling a variety of event types and rates.
  - Scaling to accommodate a large amount of STB devices.
  - Delivering feed updates timely and deterministic.
Pub/sub model

- Customized code
  - Limited memory and disk resources
  - Specialized processors and networking hardware

- Low-level
  - Pub/sub event

- High-level
  - Application domain
Data distribution service

- DDS: good performance:
  - Low-latency, high-throughput
Outline

- Introduction
- **Scenario and Architecture**
- Implementation discussion
- Experimental evaluation
- Conclusion and future work
A motivation scenario

- **TV voting:**
  - Millions of users
  - Massive events in high concurrency

![Diagram showing clients, players of interests, event processing servers, and processing results.](image)
The real-time architecture for cloud-based STB event processing and its event flow
Outline

- Introduction
- Scenario and Architecture
- Implementation discussion
- Experimental evaluation
- Conclusion and future work
Achieving the reliability

- **Data Durability:**
  - Data in event should be persisted
    - Relational database
    - Real-time database
    - File system
  - Asynchronous way is recommended
    - Ensuring the efficiency of the whole architecture
    - Late-joining subscriber
Achieving the reliability

- **Effective Redundancy:**
  - Preventing the events data from being lost
  - **Traditional redundancy**
    - Doubling or multiplying hardwares or softwares
  - **Data durability**
    - Maintaining each STB event in its own memory of STB device or in a remote node within the cluster.

- **Efficient Transmission**
  - **UDP VS TCP**
  - Extra QoS for UDP
Balancing the load

A solution for balancing the load

$$\begin{align*}
\text{Subscriber}_i & \in \begin{cases} 
L, (0 \leq \text{PubNum}(i) < \frac{\text{Sum}}{2T}) \\
M, (\frac{\text{Sum}}{2T} \leq \text{PubNum}(i) \leq \frac{\text{Sum}}{T})
\end{cases} \\
H, (\frac{\text{Sum}}{T} \leq \text{PubNum}(i))
\end{align*}$$
Outline

- Introduction
- Scenario and Architecture
- Implementation discussion
- Experimental evaluation
- Conclusion and future work
Setup

<table>
<thead>
<tr>
<th>Name</th>
<th>CPU (Intel)</th>
<th>Memory</th>
<th>NIC Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Xeon(R) E5365 (3.0GHz*8)</td>
<td>16GB</td>
<td>1000Mbps</td>
</tr>
<tr>
<td>Client</td>
<td>Core™ 2 E6700 (2.66GHz*2)</td>
<td>4GB</td>
<td>1000Mbps</td>
</tr>
</tbody>
</table>

- **Wait Time**
  - The duration for the events delivery to the working memory of the server.

- **Subscription rate**
  - The proportion of subscribed events to all published events

- **VM number**
Making use of multiple cores
Fixing waiting time

(a) 1s

(b) 2s
Fixing publication amount

The graph shows the relationship between the number of VMs and the subscription rate. The x-axis represents the number of VMs ranging from 6 to 20. The y-axis represents the subscription rate ranging from 0.4 to 1.0. Three lines are plotted, each representing a different subscription rate:

- Blue line (1s): subscription rate increases with the number of VMs.
- Green line (2s): subscription rate decreases with the number of VMs.
- Red line (3s): subscription rate decreases initially, then increases, and finally decreases again.

The graph indicates that the subscription rate increases as the number of VMs increases, but the rate of increase differs depending on the subscription period.
Outline

- Introduction
- Scenario and Architecture
- Implementation discussion
- Experimental evaluation
- Conclusion and future work
Conclusion

- Cloud computing + DDS
  - Is suitable for STB event processing
  - High efficiency
  - Varieties QoS
- Load balancing
  - an offline strategies
- Utilizing multiple cores
  - To gain high throughput
Future work

- Effective load balancing strategy
  - Exploring an online way
  - Ensuring the efficiency
- Reducing the meta-information
  - STB device has limited memory resource
  - Each node has a global knowledge