PLASMA: Programmable Elasticity for Stateful Cloud Computing Applications

Bo Sang (Purdue University, Ant Financial Services Group), Pierre-Louis Roman, Patrick Eugster (Università della Svizzera italiana), Hui Lu (Binghamton University), Srivatsan Ravi (University of Southern California), Gustavo Petri (ARM Research)
Elasticity Management for Cloud Applications
AWS Lambda Function

Each function executes independently

External storage introduce nontrivial latency
Actor-based Applications in Cloud

Scalability ✓
Low latency ✓
Elasticity ?

Orleans (SoCC’13, Eurosys’16)
EventWave (SoCC’14)
AEON (Middleware’16)
Elasticity Management for PageRank
Fine-grained Elasticity Management

- We need
  - Application information
  - User requirements
  - Server runtime information
  - Application runtime information

PLASMA Language

PLASMA Runtime
Elasticity Programming Language

- **Elasticity rules**
  - Conditions => Behaviors;
  - `server.cpu.perc > 80` or `server.cpu.perc < 60` => `balance({Partition}, cpu)`

- **Conditions**
  - **Server runtime**
  - **Actor runtime**
  - **Semantics**
  - ... 

- **Behaviors:**
  - `balance({atypes}, resource)`
  - `reserve(actor, resource)`
  - `colocate(actor, actor)`
  - `separate(actor, actor)`
  - `pin(actor)`
Elasticity Management Runtime

Profiling Runtime: collecting runtime information of actors and the server
LEM: processing rules which only require local information
GEM: processing rules which only require global information
## Evaluation: Applications

<table>
<thead>
<tr>
<th>Applications</th>
<th>Elasticity rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metadata server</td>
<td>1. Colocate Folder with Files on the same server</td>
</tr>
<tr>
<td>PageRank</td>
<td>1. Balance CPU workload</td>
</tr>
<tr>
<td>E-Store</td>
<td>1. Put hot Partitions on idle servers</td>
</tr>
<tr>
<td></td>
<td>2. Colocate parent-child Partitions</td>
</tr>
<tr>
<td></td>
<td>3. Balance CPU workload of Partitions</td>
</tr>
<tr>
<td>Media Service</td>
<td>1. Balance network workload for FrontEndsService</td>
</tr>
<tr>
<td></td>
<td>2. Provide VideoStream with enough CPU</td>
</tr>
<tr>
<td></td>
<td>3. Colocate linked VideoStream and UserInfo</td>
</tr>
<tr>
<td></td>
<td>4. Avoid migrating MovieReview</td>
</tr>
<tr>
<td></td>
<td>5. Balance CPU workload of ReviewChecker</td>
</tr>
<tr>
<td></td>
<td>6. Colocate linked ReviewEditor and UserReview</td>
</tr>
<tr>
<td>Halo Presence Server</td>
<td>1. Balance CPU workload of Routers</td>
</tr>
<tr>
<td></td>
<td>2. Colocate Session with Players in it</td>
</tr>
</tbody>
</table>
Evaluation: PageRank

Setup
- SNAP’s LiveJournal social network
- Use METIS to partition the graph into 32 partitions

24% faster with 16 vCPU (8 servers)  24 vCPU (12 servers) vs 32 vCPU (16 servers)

![Graph showing comparison between Mizan (w/ Elasticity), Mizan (w/o Elasticity), PLASMA, Orleans Elasticity]