Modeling Elasticity Trade-Offs in Adaptive Mixed Systems

Muhammad Candra, Hong-Linh Truong, Schahram Dustdar

Distributed System Group
Vienna University of Technology
Outline

• Introduction
  – Mixed System
  – Elasticity
  – Motivation
• Elasticity Profile
  – Constructs
  – Binding
• Runtime Framework
  – Adaptive Mixed System Framework
• Example
• Conclusion & Future Work
Mixed System

VieCOM
Vienna Elastic Computing Model
- Virtualization
- SCU Management
- Quality Control Strategy
- Elasticity
Elasticity in Mixed System

Elasticity dimension: Quality + Resources Scalability + Cost

- When the average utilization of the human workers on a running pool is above 8 hours per day, then additional workers must be assigned to the pool.
- A human-task requester wants to pay a cheaper price if the worker takes more than 1 hour to finish the task.
Motivation

SCU-based IT Infrastructure Monitoring and Management

We propose to model the behavior using ELASTICITY PROFILE
Constructs of Elasticity Profile

in Production Rule System:

Collection of Rules

Collection of Facts (working memory)
Elasticity Profile

- **Objects**
  - Objects represent any component of a system or a process that can behave elastically
  - MCEs: machine instances, storages, etc.
  - HCEs: human workers, human-based tasks, etc.

- **Metrics**
  - Metrics represent the quality, resource, and cost properties of the objects.

<table>
<thead>
<tr>
<th>Metric Dimension</th>
<th>Machine Metrics</th>
<th>Human Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources</td>
<td>Number of resources, utilization, storage capacity, bandwidth capacity</td>
<td>Number of resources, utilization</td>
</tr>
<tr>
<td>Quality</td>
<td>Response time, throughput, availability</td>
<td>Response time, rating, availability, throughput, task acceptance rate</td>
</tr>
<tr>
<td>Cost</td>
<td>Cost / API calls, virtual instance / hours</td>
<td>Task price, hourly price</td>
</tr>
</tbody>
</table>
Elasticity Profile

• Behavior
  – Rules for defining adaptation strategy
  – Contains condition and consequence

• Activities
  – Assignment
  – Assertion
  – Invocation
  – Exception

\[
\forall (worker, pool) \\
Worker(worker) \land ActivePool(pool) \land \\
IsMember(worker, pool) \land \\
HourUtilization(worker) \geq 8 \\
\Rightarrow AddWorker(pool)
\]
EP Grammar

\[\text{objects~statement} ::= \text{objects} \{ \text{objects~list} \} ;\]

\[\text{behavior~statement} ::= \text{behavior} \{ \text{implication~list} \} ;\]

\[\text{implication~list} ::= \text{implication} \]
\[\quad \text{implies} \text{implication} ; \text{implication~list}\]

\[\text{implication} ::= \text{check} [ : \text{priority} ] ( \text{condition} ) \{ \text{consequences} \}\]

\[\text{consequences} ::= \text{consequence} \mid \text{consequence} ; \text{consequences}\]

\[\text{consequence} ::= \text{metric~identifier} = \text{value}\]
\[\quad \text{assert} \text{instance~identifier}\]
\[\quad \text{trigger} \text{action~identifier}(\text{value~list})\]
\[\quad \text{throw} \text{exception~identifier}(\text{value})\]

\[\text{activities~statement} ::= \text{activities} \{ \text{activities~list} \} ;\]

\[\text{activities~list} ::= \text{activity} \]
\[\quad \text{activity}, \text{activities~list}\]

\[\text{activity} ::= \text{activity~identifier} (\text{activity~param~list})\]
Binding

• Profile and runtime binding are separated
• Protocol: SOAP, RESTful, Java RMI
• Objects binding
  – Subscription to event notification
• Metrics binding
  – Remote getter and setter
• Activity binding
  – Remote method invocation
Runtime Framework for Adaptive Mixed System
Example
Example

profile SCU_IT_Management {
    objects {
        Customer, Event, Warning, Incident, Analyzer,
        MonitoringAgent, ExpertSCU
    }
    metrics {
        Customer has ServiceType, ...;
        Incident has Lifetime, ...;
        Analyzer has Utilization, Type, ...;
        ExpertSCU has ExpertiseLevel, ...;
    }
    actions {
        AddAnalyzer(ANALYZER_TYPE),
        ReduceAnalyzer(ANALYZER_TYPE),
        AddMonitoringAgent(),
        ReduceMonitoringAgent(),
        UpgradeSCU(ExpertSCU, EXPERTISE_TYPE),
        TimeoutException(Incident),
    }
};
Example

```c
behavior {

  /* Dynamically scale analyzer for premium service based on the average utilization of the premium analyzers */
  check (Number(doubleValue > 0.8)
    from accumulate(
      Analyzer(Type==PREMIUM_MACHINE and u:Utilization),
      average(u))) {

    /* scale up */
    trigger AddAnalyzer(PREMIUM_MACHINE);
  };
  
  check (Number(doubleValue < 0.2)
    from accumulate(
      Analyzer(Type==PREMIUM_MACHINE and u:Utilization),
      average(u))) {

    /* scale down */
    trigger ReduceAnalyzer(PREMIUM_MACHINE);
  };
}
```
Example

/* Scale monitoring agent based on the number of queued warnings */
check (Number(intValue > 20)
    from accumulate(w:Warning(), count(w))) {
    /* scale up */
    trigger AddMonitoringAgent();
};

check (Number(intValue < 5)
    from accumulate(w:Warning(), count(w))) {
    /* scale down */
    trigger ReduceMonitoringAgent();
};

/* Upgrade SCU when the deadline is approaching*/
check (Incident(Lifetime > 2 * 3600 and
    getCustomer().ServiceType==PREMIUM and
    scu:getAssignedSCU()) and
    (scu.ExpertiseLevel < HIGH_EXPERTISE) ) {
    /* increasing expertise level,
    i.e., it will add more experts with
    higher expertise */
    trigger UpgradeSCU(scu, HIGH_EXPERTISE);
};
Conclusion

• Elasticity Profile
  – Constructs for modeling adaptation strategy in mixed systems
• Elasticity Framework
  – Mechanism for deploying and executing adaptation strategy

Future Works

• Part of VieCOM (Vienna Elastic Computing Model)
  – Quality Control Strategy for SCU
  – Discovery and negotiation on elastic human-based services
Acknowledgement
The first author of this paper is financially supported by the Vienna PhD School of Informatics
http://www.informatik.tuwien.ac.at/teaching/phdschool