GENERIC AUTONOMIC SERVICE MANAGEMENT IN COMPLEX SYSTEMS

PhD candidate: Nabila Belhaj*
Advisors: Djamel Belaïd* Samir Tata†

*CNRS/SAMOVAR, Telecom SudParis
†Almaden Research Center, IBM Research, San Jose, CA, USA
GENERIC AUTONOMIC SERVICE MANAGEMENT IN COMPLEX SYSTEMS
1. INTRODUCTION
   1.1. Context
   1.2. Contribution objectives
   1.3. Related work

2. CONTRIBUTIONS
   2.1. Framework Description
       2.1.1. Structure of the Autonomic container
       2.1.2. Structure of the Analysis component
       2.1.3. Mapping decision process to MDP problem
   2.2. Use Case Study
   2.3. Evaluation

3. CONCLUSION AND WORK IN PROGRESS

4. PUBLICATIONS
INTRODUCTION

1.1. CONTEXT
1.2. CONTRIBUTION OBJECTIVES
1.3. RELATED WORK
Complex Applications

- Complex structure and increasing size
- Heterogeneous composition of distributed and interacting components
- Highly dynamic deployment contexts
- Tedious management tasks

- **Self-Reconfiguring**
  Adapting itself to changing environments “on the fly”.

- **Self-Optimizing**
  Optimizing system performance and resource utilization.

- **Self-Healing**
  Discovering, diagnosing and acting for disruption prevention.

- **Self-Protecting**
  Identifying and anticipating unauthorized accesses to protect from attacks.
CONTRIBUTION OBJECTIVES

1. Enhance traditional ACS with sophisticated learning behavior

2. Render self-adaptive the decision making for component-based applications

3. Optimization of learning performance

4. Make autonomic loops collaborate for a consistent decision making in the overall system
## RELATED WORK

<table>
<thead>
<tr>
<th>Approaches</th>
<th>Assessment criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raghavendra, R. et al., 2008</td>
<td>-</td>
</tr>
<tr>
<td>Gueye, S.M.K. et al., 2013</td>
<td>-</td>
</tr>
<tr>
<td>Cano, J. et al., 2013</td>
<td>-</td>
</tr>
<tr>
<td>Li, Z., Parashar, M, 2005</td>
<td>-</td>
</tr>
<tr>
<td>Tesauro, G.,2007</td>
<td>-</td>
</tr>
<tr>
<td>Rao, J. et al., 2011</td>
<td>-</td>
</tr>
<tr>
<td>Wang, H. et al., 2011</td>
<td>-</td>
</tr>
<tr>
<td>Wang, H. et al., 2016</td>
<td>-</td>
</tr>
<tr>
<td>Our proposal</td>
<td>X</td>
</tr>
</tbody>
</table>
CONTRIBUTIONS

2.1. FRAMEWORK DESCRIPTION
   2.1.1. STRUCTURE OF THE AUTONOMIC CONTAINER
   2.1.2. STRUCTURE OF THE ANALYSIS COMPONENT
   2.1.3. MAPPING DECISION PROCESS TO MDP PROBLEM

2.2. USE CASE STUDY

2.3. EVALUATION
Structure of the Autonomic Container
Main goal:
Find the optimal decision policy that maximizes the long-run sum of reward signals.

Markov Decision Process (MDP):
- $S$: finite set of states,
- $A$: finite set of actions,
- $P()$: transition probability between states
- $R()$: reward function on state transitions
Structure of the Analysis component
State Space: \( S = \{ s_i \mid s_i = (v_{m_{1i}}, v_{m_{2i}}, \ldots, v_{m_{ji}}) \} \)

- \( v_{m_{ji}} \): runtime value of metric \( m_j \) of \( s_i \)
- \( s_i \): should comply with \( O = \{ o_{m_1}, o_{m_2}, \ldots, o_{m_j} \} \)
- \( o_{m_j} \): objective value for metric \( m_j \)

Action Space: \( A = \{ A_{el} \cup A_{comp} \cup A_{BPM} \} \)

- \( A_{el} \): elementary management service
- \( A_{comp} \): composition of local and/or remote \( A_{el} \)
- \( A_{BPM} \): orchestration of local and/or remote \( A_{el} \)
Reward Function

\textbf{Algorithm 1. Reward Function}

1: Input: $s_t, s'_t$
2: for all metrics values $v'_{m1t}, v'_{m2t}, \ldots, v'_{mjt}$ do
3: \hspace{1em} $r_{v'_{mklt}} = \begin{cases} 0 & \text{if } v'_{mklt} = o_{mk} \text{ (border)} \\
\frac{d_{v'_{mklt}, o_{mk}}}{d_{v_{mklt}, o_{mk}}} & \text{if } v'_{mklt} \text{ satisfies } o_{mk} \text{ (good or best)} \\
-d_{v'_{mklt}, o_{mk}} & \text{otherwise (bad or worst)} \end{cases}$
4: end for
5: $C_{s'_t} \leftarrow \text{ComplianceDegree}(\min\{r_{v'_{m1t}}, r_{v'_{m2t}}, \ldots, r_{v'_{mjt}}\})$
6: $r_{s'_t} \leftarrow \text{ComputeReward}(C_{s'_t})$
7: do loop in Step 2 for metrics values $v_{m1t}, v_{m2t}, \ldots, v_{mjt}$
8: $C_{s_t} \leftarrow \text{ComplianceDegree}(\min\{r_{v_{m1t}}, r_{v_{m2t}}, \ldots, r_{v_{mjt}}\})$
9: $r_{(s_t \rightarrow s'_t)} \leftarrow \text{CompareStates}(C_{s_t}, C_{s'_t})$
10: $r_t \leftarrow r_{s'_t} + r_{(s_t \rightarrow s'_t)}$
11: assign $r_t$ to couple $(s_t, a_t)$
12: Output: $r_t$
Online RL decision making

Algorithm 2. Learning Decision Process

1: **Input:** \( A, O, ECA, \alpha_0, \gamma, \epsilon, \lambda, \epsilon_z, \xi \)
2: **Initialize** \( Q_0 \leftarrow \text{Parse}(ECA) \)
3: \( z_0(s, a) \leftarrow 0, \forall (s, a) \)
4: \( \alpha_0(s, a) \leftarrow \alpha_0, \forall (s, a) \)
5: \( t \leftarrow 0 \)
6: Repeat:
7: \( s_t \leftarrow \text{MonitorCurrentState} \)
8: \( a_t \leftarrow \text{ChooseAction}(s_t) \)
9: \( \text{ApplyAction}(a_t) \)
10: \( s'_t \leftarrow \text{MonitorCurrentState} \)
11: \( r_t \leftarrow \text{RewardFunction}(s_t, s'_t) \)
12: \( \delta_t(s_t, a_t) = r_t + \gamma \max_b Q_t(s'_t, b) - Q_t(s_t, a_t) \)
13: \( z_t(s_t, a_t) = z_t(s_t, a_t) + 1 \)
14: Update \( Q_t(s, a) \) and \( z_t(s, a) \):
15: \( (S, A)_{\text{transition}} \leftarrow (S, A)_{\text{transition}} \cup \{(s_t, a_t)\} \setminus \{\text{argmin}_{(s, a)} z(s, a)\} \)
16: for all \((s, a) \in (S, A)_{\text{transition}}\) do
17: \( Q_{t+1}(s, a) \leftarrow Q_t(s, a) + \alpha_t(s, a)z_t(s, a)\delta_t(s_t, a_t) \)
18: \( z_{t+1}(s, a) = \gamma \lambda z_t(s, a) \)
19: end for
20: \( \alpha_t \leftarrow \text{Decay}(\alpha_t) \)
21: \( s_t \leftarrow s'_t \)
22: \( t++ \)
23: Until: All \( \alpha \delta \leq \xi \)
24: Return All \( Q(s, a) \) and Retrieve \( \pi^* \)
**Metrics objective values:**
- \( v'_{\text{avai}} = \text{true}, 90\% \) of time
- \( v'_{\text{resp}} \leq 1200 \text{ ms} \)
- \( v'_{\text{calls}} \leq 10 \)

**State vector:**
\[ (v_{\text{avai}}, v_{\text{resp}}, v_{\text{calls}}) \]
Evaluation criterion of the Framework

- Transformation overhead of the application
- Self-adaptivity to context changes,
- SLA guarantee,
- Performance optimization of learning phase

Context dynamics:

- Workload (i.e., concurrent service calls) variations:
  - Light, Medium, Heavy
- Random triggering of service unavailability

Experiments:

- One-step online learning
- Multi-step online learning
Transformation overhead of the application
Self-adaptivity to context changes

- Service Availability
- Average Response Time (ms)
- Number of Service Calls
- Reward Signal

EVALUATION
SLA guarantee and Performance optimization
Convergence speed: one-step vs. multi-step online learning
CONCLUSION AND WORK IN PROGRESS
CONCLUSION

► Improve the decision making process of a traditional MAPE-K loop

► Replace the common use of inflexible hand-coded strategies for being knowledge-intensive and inadequate to dynamically changing contexts

► Design of sophisticated and better performing autonomic systems that learn based on their past experiences

► Dynamically compute a decision policy that suits the context dynamics
► Developing collaboration algorithms to make remote autonomic containers collaborate

► Make Analyzes components collaborate for a global consistent learning of decision policies

► Propose mechanisms of conflict detection and conflict resolution for the Analyzes decisions.
Thank you for your attention
Questions are welcome
Published papers:


Submitted papers: