RESOURCE PROVISIONING IN HYBRID CLOUD COMPUTING IN THE PRESENCE OF RESOURCE FAILURES

Bahman Javadi
School of Computing, Engineering and Mathematics
Western Sydney University, Australia
AGENDA

- Introduction
- Hybrid Cloud Architecture
- Proposed Approaches
- Proposed Provisioning Policies
- Performance Evaluation
- Simulation Results
- Conclusions
- References
INTRODUCTION

- Hybrid Cloud Systems
  - Public Clouds
  - Private Clouds
- Resource Provisioning in Hybrid Cloud
  - Users’ QoS (i.e., deadline)
  - Resource failures
- Taking into account
  - Workload model
  - Failure characteristics
    - Failure correlations
    - Failure model
HYBRID CLOUD ARCHITECTURE

- Based on InterGrid components
- Using a Gateway (IGG) as the broker
WORKLOAD MODEL

- Scientific Applications
  - Potentially large number of resources over a short period of time.
  - Several tasks that are sensitive to communication networks and resource failures (*tightly coupled*)

- User Requests
  - Type of virtual machine;
  - Number of virtual machines;
  - Estimated duration of the request;
  - Deadline for the request (optional).
Failures in User Requests

- Resource failure is inevitable
  - Redundant components in public Clouds
    - highly reliable service
  - Leads to service failure in private Clouds

- Correlation in Failures → overlapped failures
  - Spatial correlation means multiple failures occur on different nodes within a short time interval.
  - Temporal correlation is the skewness of the failure distribution over which means failure events exhibit considerable autocorrelation at small time lags, so the failure rate changes over time.
The sequence of overlapped failures

\[ H = \{ F_i \mid F_i = (E_1, \ldots, E_n), T_s(E_{i+1}) \leq T_e(E_i) \} \]

Downtime of the service

\[ D = \sum_{\forall F_i \in H} \left( max\{T_e(F_i)\} - min\{T_s(F_i)\} \right) \]
PROPOSED APPROACHES

- **Knowledge-free Approach**
  - No Failure Model
  - Using failure correlation
  - Three brokering policies

- **Knowledge-based Approach**
  - Failure Model
  - Generic resource provisioning model
  - Two brokering policies (cost-aware)

- **Workload model**
  - Request size
  - Request duration
PROPOSED POLICIES

- Size-based Strategy
  - **Spatial correlation**: multiple failures occur on different nodes within a short time interval
  - **Strategy**: sends wider requests to more reliable public Cloud systems
  - Mean number of VMs per request
    - $P_1$: probability of one VM
    - $P_2$: probability of power of two VMs
    
    \[ \overline{S} = P_1 + 2^{[k]}(P_2) + 2^k (1 - (P_1 + P_2)) \]
    
    - Request size: two-stage uniform distribution ($l,m,h,q$)
    
    \[ k = \frac{ql + m + (1 - q)h}{2} \]
PROPOSED POLICIES (CONT.)

- Time-based strategy
  - **Temporal correlation**: the failure rate is time-dependent and some periodic failure patterns can be observed in different time-scales
  - **Request duration**: are *long tailed*.

- The mean request duration
  - Lognormal distribution in a parallel production system
  
  \[
  \overline{T} = e^{\mu + \frac{\sigma^2}{2}}
  \]

- The performance metrics which are considered in all simulations include:
  - Bounded slowdown: the speed of the request when it is serviced by a public Cloud provider, compared to the speed when it is serviced by the private Cloud.
  - Cost: the cost of using resources from a public Cloud, including both CPU and network usage.
  - Backfilling: the process of granting reservations to a request when its expected slowdown exceeds a threshold.
  - Slowdown: the ratio of the execution time of a request on the public Cloud to the execution time on the private Cloud.
  - Expansion Factor: the ratio of the number of VMs used to the number of VMs available.
  - Completion Time: the time taken for a request to complete on the public Cloud.
  - Reliability: the probability that a request will complete within a given deadline.
PROPOSED POLICIES (CONT.)

Area-based strategy

- Making a **compromise** between the size-based and time-based strategy
- The mean area of the requests

\[
\bar{A} = \bar{T} \cdot \bar{S}
\]

- This strategy sends long *and* wide requests to the public Cloud,
- It would be more conservative than a *size-based* strategy and less conservative than a *time-based* strategy.
Knowledge-based Approach: Generic Resource Provisioning Model

- Model based on routing in distributed parallel queue

\[ \min \sum_{i=1}^{n} (K_i \cdot E[T_i]) \]

\( Pi: \) routing probability

\( Ki: \) price of provider \( i \)
**MODEL PARAMETERS**

- Using Lagrange multipliers methods, we obtained the routing probability as follows:

  \[ P_i = \frac{\mu_i}{\lambda} - \sum_{i=1}^{n} \frac{\mu_i - \lambda}{\lambda} \cdot \frac{\sqrt{K_i \eta_i}}{\sum_{i=1}^{n} \sqrt{K_i \eta_i}} \]

- Private Cloud service rate

  \[ \mu_s = \left( \frac{W}{M_s \cdot \tau_s} \cdot \frac{t_a + t_u}{t_a} + L_s \right)^{-1} \]

- Public Cloud service rate

  \[ \mu_c = \left( \frac{W}{M_c \cdot \tau_c} + L_c \right)^{-1} \]
Adaptive Policies

- Adaptive with Random Sequence (ARS)
  - Routing probabilities ($P_i$)
  - Dispatch using Bernoulli distribution
- Adaptive with Deterministic Sequence (ADS)
  - Routing probabilities ($P_i$)
  - Dispatch using Billiard sequence

\[ i_b = \min_{\forall i} \left\{ \frac{X_i + Y_i}{P_i} \right\} \]
SCHEDULING ALGORITHMS

- Scheduling the request across private and public Cloud resources
- Two well-known algorithms where requests are allowed to leap forward in the queue
  - Conservative backfilling
  - Selective backfilling

\[ XFactor = \frac{W_i + T_i}{T_i} \]

- VM Checkpointing
  - VM stops working for the unavailability period
  - The request is started from where it left off when the node becomes available again
**Performance Evaluation**

- **CloudSim Simulator**
- **Performance Metrics**
  - Deadline violation rate
  - Slowdown
    \[
    \text{Slowdown} = \frac{1}{M} \sum_{i=1}^{M} \frac{W_i + \max(T_i, \text{bound})}{\max(T_i, \text{bound})}
    \]
  - Cloud Cost on EC2
    \[
    \text{Cost}_{pl} = (H_{pl} + M_{pl} \cdot H_u) C_n + (M_{pl} \cdot B_{in}) C_x
    \]
  - Workload Model
    - Parallel jobs model of a multi-cluster system (i.e., DAS-2)

<table>
<thead>
<tr>
<th>Input Parameters</th>
<th>Distribution/Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inter-arrival time No. of VMs Request duration</td>
<td>Weibull ((\alpha = 23.375, 0.2 \leq \beta \leq 0.3)) Loguniform ((l = 0.8, m, h = log_2 N_8, q = 0.9)) Lognormal ((2.5 \leq \mu \leq 3.5, \sigma = 1.7))</td>
</tr>
<tr>
<td>(P_1) (P_2)</td>
<td>0.02 (\cdot) 0.78</td>
</tr>
</tbody>
</table>
Performance Evaluation (cont.)

- Failures from Failure Trace Archive (FTA)
  - Grid’5000 traces
    - 18-month
    - 800 events/node
  - Table I: Input parameters for the workload model

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
<th>Value (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_a$</td>
<td>Mean availability length</td>
<td>22.25</td>
</tr>
<tr>
<td>$\sigma_a$</td>
<td>Std of availability length</td>
<td>41.09</td>
</tr>
<tr>
<td>$t_u$</td>
<td>Mean unavailability length</td>
<td>10.22</td>
</tr>
<tr>
<td>$\sigma_u$</td>
<td>Std of unavailability length</td>
<td>40.75</td>
</tr>
</tbody>
</table>

- Synthetic Deadline

$$d_i = \begin{cases} 
  st_i + (f \cdot ta_i), & \text{if } [st_i + (f \cdot ta_i)] < ct_i \\
  ct_i, & \text{otherwise}
\end{cases}$$

- $f$: stringency factor
- $f>1$ is normal deadline (e.g., $f=1.3$)

- $N_s = N_c = 64$
SIMULATION RESULTS

- Violation rate (knowledge-free policies)

- Request arrival rate
- Request size
- Request duration
SIMULATION RESULTS (CONT.)

- Cloud Cost on EC2 (knowledge-free policies)

- **Request arrival rate**

- **Request size**

- **Request duration**
Simulation Results (Cont.)

- Slowdown (Knowledge-based policies)

- Request arrival rate (SB)

- Request arrival rate (CB)

- Request arrival rate (EB)
FAILURE TRACE ARCHIVE (FTA)

- 27 Failure Traces
  - Supercomputers, HPC, Grid, P2P
- FTA Format
- Simulator and Scripts

FAILURE TRACE ARCHIVE
FOR IMPROVING THE RELIABILITY OF DISTRIBUTED SYSTEMS

HOME PAGE
The Failure Trace Archive (FTA) is a centralized public repository of availability traces of parallel and distributed systems, and tools for their analysis. The purpose of this archive is to facilitate the design, validation, and comparison of fault-tolerant models and algorithms.

In particular, the FTA contains the following:
- availability traces of parallel and distributed systems, differing in scale, volatility, and usage
- a standard format for failure traces
- scripts and tools for analyzing these traces

CONCLUSIONS

○ Adaptive resource provisioning in a failure-prone hybrid Cloud system

○ *Flexible* brokering strategies based on failure correlation/model as well as workload model

○ Improve performance of hybrid Cloud
  - Knowledge-free approach: 32% in terms of deadline violation and 57% in terms of slowdown while using 135$/month on EC2
  - Knowledge-based approach: 4.1 times in terms of response time while using 1200$/month on EC2
OPEN QUESTIONS

- Recourse Failures vs. Energy Consumption for Cloud Systems
  - How they are related?

- Reliability-as-a-Service (RaaS) in Cloud Computing
  - Providing reliability on demand based on the users’ requirements (e.g., Amazon Spot Instances)

- Cost Model for Resource Failures in Cloud Systems
  - Repair .... Replacement
REFERENCES


Thank You