RELIABILITY IN CLOUD COMPUTING: ISSUES AND CHALLENGES

Dr. Bahman Javadi
School of Computing, Engineering and Mathematics
University of Western Sydney, Australia
About UWS

» One of the largest universities in Australia

» Largest undergraduate commonwealth funded university

» Over 42,000 students

» Culturally diverse student body

» ‘Bringing knowledge to life’

» Multi-campus structure
Our Proud History

A young university with a long tradition of higher education


Hawkesbury campus established

Former Australian Prime Minister Gough Whitlam announces a university in western Sydney

University of Western Sydney established

UWS College established

UWS celebrates 25 years

Bringing knowledge to life for over 200 years
Partner Institutions

» Over 104 existing international partners in 31 countries
» Collaborative agreements
» Research
» Exchange – including students and staff
» Academic cooperation
» Guaranteed admissions and articulation
» Offshore delivery
Research in School

Research Centre
- Centre for Research in Mathematics (CRM)

Three University Research Groups
- Artificial Intelligence Research Group (AI)
- Digital Humanities Research Group (DH)
- Solar Energy Technologies Research Group (SET)

School Groups
- Networking, Security an Cloud Research Group (NSCR)
- E-Health Research Group
- Computational Astrophysics
OUTLINE

- Introduction and Background
- Reliability in Cloud: Definition
- Failure Correlation
- Case Study: Failure-aware Hybrid Cloud Architecture
- Results
- Conclusions
- Issue, Challenges and Open Questions
In 1969, Leonard Kleinrock, one of the chief scientists of the original ARPA project which seeded the Internet, wrote:

- "As of now, computer networks are still in their infancy, but as they grow up and become sophisticated, we will probably see the spread of "computer utilities", which, like present electric and telephone utilities, will service individual homes and offices across the country"."
WHAT IS CLOUD COMPUTING

- Having a Car?
  - Buying a Car
    - Pay the initial cost
    - Pay the insurance cost
    - Pay the maintenance cost
  - Renting a Car
    - Pay-as-you-go

Classical Computing

Business Model

Cloud Computing
WHAT IS CLOUD COMPUTING

- New Technology  ✗
- New Architecture  ✗
- New Service  ✓

- Cloud Computing: on-demand IT Service, anywhere, anytime with a pay-as-you-go model

- The name comes from the use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams.

JEC-ECC 2015
TRADE-OFFS

Cost ($)

Cloud computing

Reliability

Performance

Supercomputers

Clusters

Desktop Grids
CLOUD INTERESTS OVER TIME

Ref: Google Trends
CLOUD SERVICES

On Premise

IaaS: Infrastructure as a Service

PaaS: Platform as a Service

SaaS: Software as a Service

Applications
Runtime
Middleware
OS
Virtualization
Servers
Storage
Networking

Applications
Data
Runtime
Middleware
OS
Virtualization
Servers
Storage
Networking

Applications
Runtime
Middleware
OS
Virtualization
Servers
Storage
Networking
Cloud Service Providers

- Cloud Applications (Apps-as-a-service)
- Cloud Platforms (Platform-as-a-Service)
- Cloud Infrastructure (Infrastructure-as-a-Service)

Private

Public
SPECTRUM OF ABSTRACTIONS

- Different levels of abstraction
  - Instruction Set VM: Amazon EC2
  - Framework VM: Google App Engine

- Similar to languages
  - Higher level abstractions can be built on top of lower ones

Lower-level, More flexibility, More management
Not scalable by default

Higher-level, Less flexibility, Less management
Automatically scalable

EC2 Azure AppEngine Google Apps

JEC-ECC 2015
CLOUD COMPUTING: PROS AND CONS

- **Pros**
  - Elasticity
  - Cost-effective
  - Quick deployment
  - Unlimited storage

- **Cons**
  - Security and Privacy
  - Vendor Lock-in
  - Lack of Control
The preparedness to support the growth of cloud computing

These 24 countries account for 80 percent of the global ICT market.
OUTLINE

- Introduction and Background
- Reliability in Cloud: Definition
- Failure Correlation
- Case Study: Failure-aware Hybrid Cloud Architecture
- Results
- Conclusions
- Issue, Challenges and Open Questions
CLOUD RELIABILITY

- A **failure** is an event that makes a system fail to operate according to its specifications.

- Resource failure is inevitable where failure is the norm rather than exception.

- **Highly Reliable Service**
  - Public Clouds
    - Redundant components to provide reliable services
  - Private Clouds
    - Frequent service failures
Hard disks are the not only the most replaced component, they are also the most dominant reason behind server failure!!

The cost of per server repair (which includes downtime; IT ticketing system to send a technician; hardware repairs) is $300. This amounts close to 2.5 million dollars for 100,000 servers.

Introduction and Background
Reliability in Cloud: Definition
Failure Correlation
Case Study: Failure-aware Hybrid Cloud Architecture
Results
Conclusions
Issue, Challenges and Open Questions
Failure Correlation

- Correlation in Failures → overlapped failures
  - Spatial
  - Temporal

- 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.
Failures in Service

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

JEC-ECC 2015
OUTLINE

- Introduction and Background
- Reliability in Cloud: Definition
- Failure Correlation
- Case Study: Failure-aware Hybrid Cloud Architecture
- Results
- Conclusions
- Issue, Challenges and Open Questions
CASE STUDY

- 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

JEC-ECC 2015
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).
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 $(l, m, h, q)$
    
    $$
    k = \frac{ql + m + (1 - q)h}{2}
    $$

JEC-ECC 2015
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.

![Graph showing cumulative distribution of request duration]

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

JEC-ECC 2015


**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

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

- Public Cloud service rate

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

*JEC-ECC 2015*
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-know 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
OUTLINE

- Introduction and Background
- Reliability in Cloud: Definition
- Failure Correlation
- Case Study: Failure-aware Hybrid Cloud Architecture
- Results
- Conclusions
- Issue, Challenges and Open Questions
**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</td>
<td>Weibull ($\alpha = 23.375, 0.2 \leq \beta \leq 0.3$)</td>
</tr>
<tr>
<td>No. of VMs</td>
<td>Loguniform ($l = 0.8, m, h = log_2 N_s, q = 0.9$)</td>
</tr>
<tr>
<td>Request duration</td>
<td>Lognormal ($2.5 \leq \mu \leq 3.5, \sigma = 1.7$)</td>
</tr>
<tr>
<td>$P_1$</td>
<td>0.02</td>
</tr>
<tr>
<td>$P_2$</td>
<td>0.78</td>
</tr>
</tbody>
</table>
Failures from Failure Trace Archive (FTA)
- Grid’5000 traces
  - 18-month
  - 800 events/node

<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., $f=1.3$)

$N_s = N_c = 64$

JEC-ECC 2015
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)

![Graphs showing slowdown vs. request arrival rate for NoR-SB, ARS-SB, and ADS-SB, for different values of \( \beta \).]

![Graphs showing slowdown vs. request arrival rate for NoR-CB, ARS-CB, and ADS-CB, for different values of \( \beta \).]

![Graph showing slowdown vs. request arrival rate for NoR-EB, ARS-EB, and ADS-EB, for different values of \( \beta \).]
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

MAR 09, 2015

HOMEPAGE

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

JEC-ECC 2015


Thank You