Resource Provisioning of Web Applications in Heterogeneous Clouds

Jiang Dejun          Vrije University & Tsinghua University
Guillaume Pierre    Vrije University
Chi-Hung Chi        Tsinghua University

2011-06-15

USENIX WebApps 2011
Background

- Web applications play an important role in our daily life.
- Web application performance is one primary concern of application providers.
  - Users leave a website if it responds slowly
  - Application providers define Service Level Objectives (e.g. average maximum response time)
  - Dynamic resource provisioning helps to guarantee Web application performance
  - Web application hosting moves to Cloud for elastic resource usage
Motivation

- Cloud resource is heterogeneous
  - Heterogeneous virtual machine types
  - Heterogeneous performance of same type
Motivation (cont.)

- Cloud resource is heterogeneous
  - Resource heterogeneity is a long-term observation
  - Resource heterogeneity is observed across Clouds (e.g. EC2, Rackspace)
Motivation (cont.)

- Cloud resource is heterogeneous
  - Current resource provisioning in Clouds (e.g. EC2)
Problem statement

- How to provision Web applications in Clouds
  - If an instance with fast CPU, it may be better to use it as an application server
  - If an instance with fast IO, it may be better to use it as a database server
  - We do not know how to use the new instance but we need to make a decision

- Difficulties
  - Unpredictable performance of new instances
  - Different performance benefits on different tiers of a new instance
Intuitive solutions

- Ignore the heterogeneous resource feature
  - Apply current resource provisioning algorithm to make decision
- Profile new instances at each tier to make decision
  - Deploy new instance as application server is fast
  - Deploy new instance as database server costs.
    e.g. DB size: 1.6GB. Dump: 190s; Transfer: 64s; Import 1530s. Total 30 min
  - This approach is inefficient and time-consuming
Outline

- Background
- Motivation
- Problem statement
- Intuitive solutions
- Our proposal
- Experimental evaluation
- Conclusion
Our proposal

- Performance correlation
  - Performance profile of a given tier is related to its resource utilization
  - Performance profiles of two different tiers (with same type resource demand) can be highly correlated
Our proposal (cont.)

- Performance prediction
  - Step 1: Employ reference applications as the calibration base
  - Step 2: Correlate resource demands of reference applications and tier services on the calibration instance
  - Step 3: Profile new instances with reference applications
  - Step 4: Derive performance of tier services on new instance
Our proposal (cont.)

- Resource provisioning
  - Obtain performance profiles of new instances
  - Apply "what-if" analysis to predict the performance of the whole application if a new instance is added to a tier
Experimental evaluation

- Experiment setup
  - Reference applications
    - a CPU-intensive application: CPU(ref)
    - an IO-intensive application: IO(ref)
  - Tested application: TPC-W (a benchmark modeling the online bookstore)
  - Weighted round-robin load balancer
  - All experiments run on Amazon EC2
Experimental evaluation

- Weighted round-robin load balancer
  - Dispatch requests among virtual instances
  - Support sticky sessions
Experimental evaluation

- Exp1: importance of adaptive load balancing

Adaptive load balancing can adapt to heterogeneous capacities of instances and enable equal response times.
Experimental evaluation

- Exp2: effectiveness of provisioning TPC-W

We have different adoptions in two groups of experiments when provisioning TPC-W on EC2 due to resource heterogeneity.
Experimental evaluation

- Exp3: comparison with other techniques

Our system achieves higher throughput using the same instances compared with other provisioning techniques
Conclusion

- Performance guarantees for Web applications are important
- **Cloud is heterogeneous**: current resource provisioning techniques are not suitable
- We propose to correlate resource demands of hosted applications with reference applications.
- One can derive the performance of Web application on new instances by just profiling new ones with reference applications.

Thank you!