Amazon Web Services: EC2, S3, and SQS

Summarized by: Michael Riera

9/17/2011
Papers in discussion

- **Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud**
  - Keith R. Jackson et al...
  - Advanced Computing for Science Lawrence Berkeley National Lab, and IT Lawrence Berkeley National Lab,

- **Evaluating Caching and Storage Options on the Amazon Web Services Cloud**
  - David Chiu et al..
  - Ohio State CSE

  - Simon L. Garfinkel
  - Center of Research on Computation and Society, Harvard
Agenda

• Purpose
• Related Work
• Understanding S3, EC2, and SQS
  – Methodology / Implementation
  – AWS Utility Services and Benchmarks
• Conclusion
Introduction

• The purpose of this paper is to exercise AWS S3, EC2 and SQS utility services
  – 2007/08 Architecture
    • Supports: Linux-based running on top of the Xen virtualization engine
    • The VM provides an equivalent 1.7Ghz x86 1.75GB RAM, and 160GB of local storage, with 250Mb/s in network connectivity
    • Amazon Machine Instance (AMI) stored on S3
S3 Storage Model

• Stores data as named object, grouped in ‘Buckets’
  – Bucket names are global; (not sure if this is still the case)

• S3 is designed to quickly fail on error
  – Client’s responsible for retrying failed request
    • Happens less than 1%; when implemented 100% of availability can be reached
    • Read & Write Failures
      – Write Ack: S3 computes an MD5 for every object- Client computes MD5, if not match, object corruption. (same applies to read request)
Amazon SQS Storage Model

• ASQS allows users to create one or more named queues.
  – Supports three basic operations:
    • Send, receive, and deleting a single or multiple messages, sequentially or concurrently
  – 8KB Message size
  – Unlimited queues and messages
  – Locking Messages
Amazon Elastic Computing Services

• Instantiating Multiple instances of your Template AMI.
  – Instances ran: Linux 2.6.16- XenU kernal
• Boot up time is less than 2 minutes
• Reliability (1 year):
  – 1 unscheduled reboot, and 1 lockup
• Load balancers single points of failure
Evaluation

• Evaluation Amazon S3, SQS, EC2
  – Created throughput test in C++
    • Measured: Upload and download time of a batch of 1 Byte, 1KB, 1 MB, 16MB, and 100MB (referred to as probe files)
      – Different variety for cache busting
      – Experiment 1: Separated with a random delay governed by a poison distribution
      – Experiment 2: Repeated queries with no delay
  – Tested with multiple thin clients around the US
    • 2 computers at Harvard U, 2 computers at MIT, shared server at ISP LA, and a server at ISP Pittsburgh
Evaluation

• Evaluation Amazon S3, SQS, EC2
• 137,099 probes
  – 32,748 probes on a single EC2 instance to S3
  – 74,808 for our surge experience
  – 39,269 from the thin clients
Evaluation

- Evaluation Amazon S3, SQS, EC2

Average Read throughput 100MB probe
Evaluation

- Evaluation Amazon S3, SQS, EC2

Figure 3: Average hourly TPS as measured by 1 byte GET operations from EC2.

Average Read throughput 1 byte
Evaluation

- Evaluation Amazon S3, SQS, EC2

Bandwidth (reads from EC2 to S3) CDF plots
Evaluation

- Evaluation Amazon S3, SQS, EC2

Bandwidth (write from EC2 to S3) CDF plots
Saving to multiple locations
Evaluation

• Evaluation Amazon S3, SQS, EC2

Query Variance: making repeated calls with lower packet size