AGILE: Elastic distributed resource scaling for Infrastructure-as-a-Service

Hiep Nguyen, Zhiming Shen, Xiaohui (Helen) Gu
North Carolina State University

Sethuraman Subbiah
NetApp

John Wilkes
Google
Elastic resource scaling for Infrastructure-as-a-Service

- Elasticity: grow/shrink resource as required
Design goals

- Application agnostic
  - Easy to deploy
  - Support different applications

- Effective overload handling
  - Predict overload accurately
  - Minimize SLO violations
  - Minimize resource cost

- Low overhead
  - Light-weight
  - Little interference
State of the art

- **Reactive resource scaling** [e.g., Amazon EC2]
  - Performance degradation due to long instantiation latency ($\approx 2$ minutes)

- **Trace-driven resource scaling** [e.g., Chandra et al. IWQoS 2003, Gong et al. CNSM 2007, Shen et al. SOCC 2011]
  - Focus on short-term prediction or need to assume cyclic workload patterns

- **Model-driven resource scaling** [e.g., Zhu et al. ICAC 2008, Kalyvianaki et al. ICAC 2009, Padala et al. Eurosys 2007]
  - Have parameters that need to be specified or tuned offline for different applications/workloads
AGILE system overview

- Resource usage monitoring
- Resource demand prediction
- Resource pressure modeling
- Server pool prediction
- Server pool scaling manager

**Key Processes:**
- **Overload starts**
- **Overload stops**
- **Advance alert**
- **SLO violation feedback**
- **Future resource demand**
- **Resource to maintain**
- **When to scale**
- **How many VMs**

**Actions:**
- **VM add/remove**
Pre-copy live VM cloning

- **Design goals**
  - Immediate performance scale-up
  - Avoid storing and maintaining VM snapshots

Pre-copy memory

Disk Image
Pre-copy live VM cloning

- **Design goals**
  - Immediate performance scale-up
  - Avoid storing and maintaining VM snapshots
Pre-copy live VM cloning

- **Dynamic copy-rate configuration**
  - Minimum copy-rate (e.g., little interference)
  - Finish cloning within the overload pending time

\[
\text{CopyRate} = \frac{\text{MemorySize}}{t_{\text{clone}}} + r_{\text{dirty}}
\]
Performance scale-up comparison

Immediate performance scale-up

Application starts

New VM request

Cassandra throughput (ops/sec)

Time (s)

Boot from scratch
Cold cloning
AGILE live cloning

Immediate performance scale-up
Wavelet-based medium-term prediction

Original signal

Decompose

Detail signals

Synthesize

Approximation signals

Scale 1

Scale 2

Scale 3

Scale 4
Online resource pressure modeling

- Mapping function between:
  - Resource pressure
  - SLO violation rate
Optimizations for AGILE cloning

- **Post-cloning auto-configuration**
  - Event driven auto-configuration
  - Application VMs can subscribe to critical events

- **False alarm handling**
  - Continuously check predicted overload state
  - Cancel cloning triggered by the false alarm
Experimental evaluation

- **Implemented on top of KVM**
  - Modified KVM to support pre-copy live cloning

- **Test bed:**
  - 10 cloud nodes running CentOS 6.2 with KVM 0.12.1.2

- **Benchmark systems**
  - RUBiS driven by four real workload traces
    - WorldCup'98, EPA, Nasa, ClarkNet (one day traces)
  - Google cluster data: 100 CPU usage and 100 Memory usage traces (29 days)
Wavelet-based prediction accuracy

- RUBiS traces
Wavelet-based prediction accuracy

- RUBiS traces
Wavelet-based prediction accuracy

- Google CPU traces
Overload handling

- Web server and database server scaling ($\approx 2$ hours, scale from 1 to 2 servers)
Overload handling

- Web server: during scaling
Dynamic copy-rate configuration

Accurately control the cloning time under different deadlines
Conclusion

- Prediction-driven elastic distributed resource scaling:
  - Accurate medium-term prediction based on wavelet transforms
  - Adaptive copy-rate to minimize interference
  - Application-agnostic performance model
- Immediate performance scale-up with little overhead

Thank you!

http://dance.csc.ncsu.edu
Acknowledgement

- This work was sponsored in part by NSF CNS0915567 grant, NSF CNS0915861 grant, U.S. Army Research Office (ARO) under grant W911NF-10-1-0273, and Google Research Awards