Providing Monitoring-as-a-Service for Cloud Management

Shicong Meng (smeng@cc.gatech.edu)
Advisor: Prof. Ling Liu (lingliu@cc.gatech.edu)
Overview

• **Monitoring-as-a-Service (MaaS)**
  – Various Benefits for both Cloud users and service providers
  – Primitive cloud monitoring services
    • E.g. Cloud Watch, Command Center

• **State Monitoring** is one of the most widely used monitoring services
  – Continuously checking if a certain state of the monitored application/system violates a given condition
  – Examples:
    • Hotspot detection
    • Auto-scaling
    • DDoS detection
Overview

• Core functional components in state monitoring services
  – Violation Detection
  – State Information Collection
  – Multi-Tenancy Support

• Challenges
  – Violation detection
    • Accuracy, efficiency, scalability
  – State information collection
    • overhead-utility tradeoff
  – Multi-tenancy support
    • Isolation, resource management
Violation Detection

• Definition
  – Given collected monitoring data, determine whether there exists an state violation

• Existing techniques
  – Centralized detection
    • Collecting all monitoring data to a central point
    • Perform violation detection
    • Issues
      – high monitoring cost (communication)
      – Poor scalability (central point)
Violation Detection

- Existing techniques (cont’d)
  - Instantaneous distributed detection
  - Reduces communication cost
  - Issues
    - Vulnerable to transient data outliers and noises
    - Expensive counter-measures

![Diagram showing coordination and violation detection with local and global monitors](image)

- Coordinator
- \[ \sum X_i > T ? \]
- Monitors:
  - \( T_1 = 10 \)
  - \( T_2 = 10 \)
  - \( T_3 = 10 \)
  - \( T_4 = 10 \)
  - \( T_5 = 10 \)
  - \( T_6 = 10 \)

- Global poll
- E.g. \( T = 60 \)

Short-term burst
Violation Detection

- We propose distributed window based detection
  - In addition to threshold $T$, detecting continuous violation within a time window $L$
  - Robust to short-term bursts
  - Straightforward concept, but less intuitive distributed implementation…
Violation Detection

- Challenges in distributed implementation
  - Global-to-local task decoupling now involves monitoring time window (besides a threshold)
  - Ensure monitoring correctness
  - Can we also leverage monitoring time window to achieve even better communication efficiency?
Violation Detection

• Our approach
  – Detection algorithm → correctness
    • Monitor-side algorithm
    • Coordinator-side algorithm
  – Monitoring parameter tuning → efficiency
    • Global optimization based tuning
    • Local observation based tuning
Violation Detection

• Window-based monitoring algorithm
  – Coordinator side
    • State violation requires $\sum X_i > T$ to be continuous
    • “Gaps” in a time window $\rightarrow$ no violation $\rightarrow$ no need to do global poll
    • Staged global polls
Violation Detection

• **Window-based monitoring algorithm**
  
  – Monitor side
    
    • Reporting scheme and correctness
    
    • Monitors often observe continuous local violations
      
      – E.g. continuous high cpu utilization on a cluster node
    
    • Intelligently reporting *continuous* local violations
Violation Detection

- Monitoring efficiency and parameter tuning
  - The detection algorithm itself already provides considerable communication saving
    - E.g. for a window size of 15, about 33% reduction in communication cost
  - Further improvement can be achieved with parameter tuning
    - parameters: monitor-side local threshold and windows
    - Tuning is necessary for several reasons
      - Different monitored value patterns on different monitors
      - Such patterns may also change over time
Violation Detection

- Parameter tuning schemes
  - Global optimization scheme
    - Collecting monitored value distribution and perform optimization with global information
    - Good performance, limited scalability
  - Reactive turning scheme
    - React to local observations
      - Local violation report -> increase local threshold/window
      - Global poll -> reduce local threshold/window
    - Slightly worse performance, significant better scalability
Violation Detection

- A Quick look of Results
  - 50%-90% reduction in monitoring related messages
Violation Detection

- A Quick look of Results (cont’d)
  - Reactive tuning scales better than global optimization based tuning
State Information Collection

- **Periodical Collection**
  - The only option for state monitoring in most monitoring systems.
  - Cost-accuracy dilemma

- **Violation-Likelihood Based Collection**
  - Likelihood of detecting violation
  - Adjusting collection frequency based on VL
  - Maintaining a given accuracy goal
  - Benefits
    - Better service consolidation
    - Lower monitoring cost for customers

- **Results**
  - Up to 90% cost reduction in state information collection
  - Negligible mis-detection rate
Multi-tenancy Support

- Multi-tenancy in Monitoring Service
  - Indispensable
  - Challenges

- Resource-Aware Planning
  - Monitoring communication layer
  - Communication topology planning
    - Per-node available resources
    - Per-node monitoring workload
    - Minimizing duplicated workload
  - Benefits
    - Avoid inter-task interference
    - Better scalability

- Results
  - 35%-45% error reduction in attribute value collection
Conclusion and Ongoing Work

- MaaS and Cloud
  - MaaS will make Cloud management easier and more efficient
  - There are also many challenges ahead waiting us in delivering MaaS.

- Ongoing work
  - Reliability support in MaaS
  - Cloud application deployment support with MaaS

- Related publications

  [1] Shicong Meng, Ling Liu and Ting Wang "State Monitoring in Cloud Datacenters". IEEE Transactions on Knowledge and Data Engineering (TKDE), Special Section on Cloud Data Management, VOL. 23, NO. 9, SEPTEMBER 2011.
  [4] Shicong Meng, Srinivas Karshyap, Chitra Venketramani and Ling Liu, "REMO: Resource-Aware Application State Monitoring for Large-Scale Distributed Systems". Proceedings of IEEE Int. Conf. on Distributed Computing (ICDCS’09), June 22-26, in Montreal, Quebec, Canada.
Thank You!

Please visit http://www.cc.gatech.edu/~smeng for more information