Towards Model–based Management of Database Fragmentation

Asim Ali, Abdelkarim Erradi, Rashid Hadjidj

Qatar University

Rui Jia, Sherif Abdelwahed

Mississippi State University

The 8th International Workshop on Feedback Computing
San Jose, CA, USA
June 25th, 2013

MISSISSIPPI STATE UNIVERSITY

Outline

- Introduction
- Model-based Control Framework
- Control Approach
- Case Study
- Conclusion

Introduction

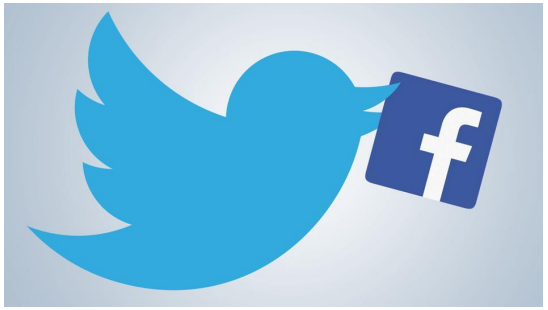
The 8th International Workshop on Feedback Computing
San Jose, CA, USA
June 25th, 2013

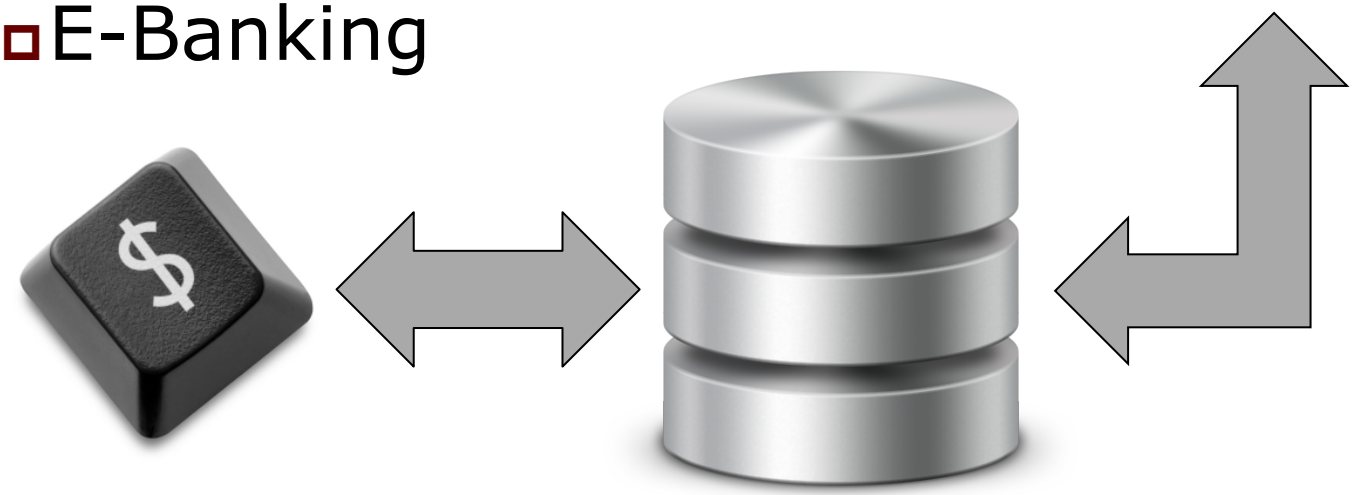


MISSISSIPPI STATE UNIVERSITY™

Database Fragmentation

- Databases are integral parts of systems such as:
  - Social Networks
  - Online trading
  - E-Banking





Database Fragmentation

- Clustered indexes
- Cause: large number of write, delete and update operations
- Internal and external

MISSISSIPPI STATE UNIVERSITY™

Database Fragmentation

Internal and External Fragmentation

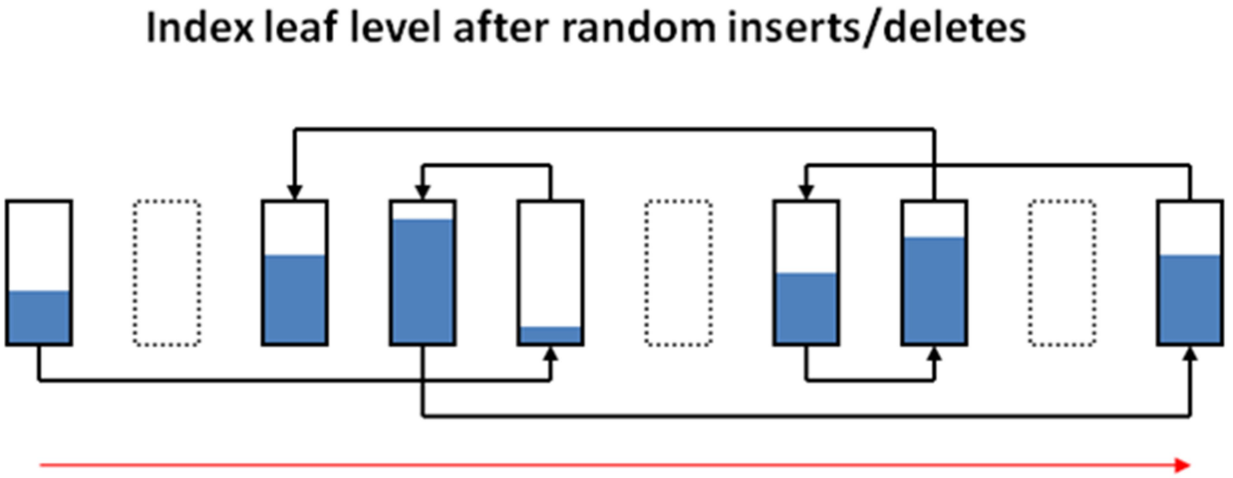


Image courtesy:*SQLSkills.com*

Fragmentation Impacts

Fragmented Database
- Internal
- Extent
- Disk

→

Increased Resource Usage
- CPU usage
- Memory Usage
- Disk IO

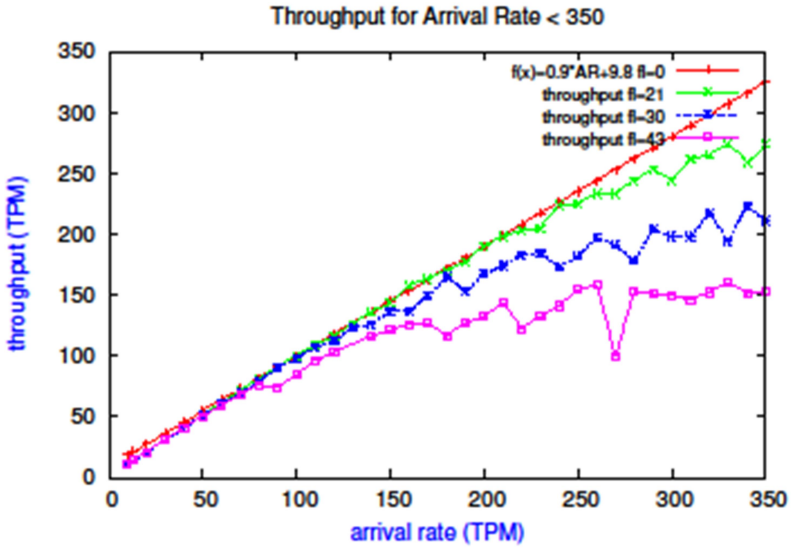
→

Degrade QoS Parameters
- Response time
- Throughput
- Availability

→

Causes Financial Overhead
- Financial Penalties due to SLA viloations

Example: Decreasing throughput with increasing fragmentation levels



MISSISSIPPI STATE UNIVERSITY

Defragmentation Approaches

Conventional

- Human expertise
- Rules of thumb decide the time and indexes to defragment
- Does not consider impact during defrag.
- Potentially suboptimal decision
- No model is required

Model-based

- Autonomous controller
- Model-based decision
- Considers impact on system performance
- Decision is optimal based on model
- System models are required

MISSISSIPPI STATE UNIVERSITY

Contribution

- A fragmentation management framework
  - Model-based
  - Considers dynamic SLAs
  - Fragmentation detection
  - Performance optimization
- Autonomic controller

Model-based Control Framework

The 8th International Workshop on Feedback Computing
San Jose, CA, USA
June 25th, 2013



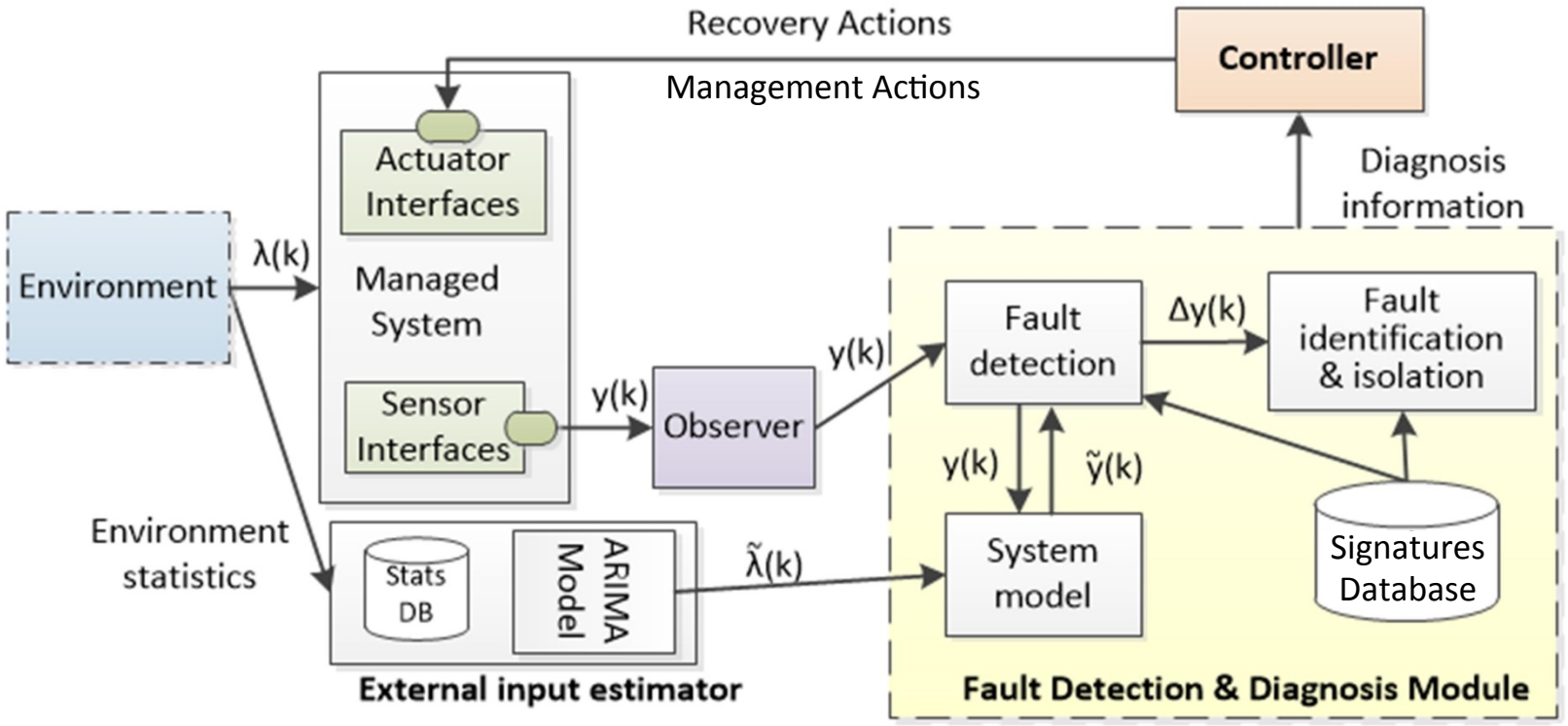
MISSISSIPPI STATE UNIVERSITY™

Model-based Control Framework

Key components

- Environment (workload)
- Managed System
- Fault Diagnosis
- Performance Controller

Model-based Control Framework



Overview of the Framework

MISSISSIPPI STATE UNIVERSITY

Model-based Control Framework

- Workload variations
  - Arrival rate (TPM)
- Fault variations
  - Fragmentation levels
- Monitored Parameters
  - CPU usage, memory usage, disk IO, response time, throughput, etc.
- Data analysis: curve fitting tools

Model-based Control Framework

Model Development Process

- Analyzing historic traces
- Simulations
- Relationships
- Benchmark workload
  - Based on OLTP workload configurations
  - Performance measured based on read operations
- Fragmentation injection script

Model-based Control Framework

- Model-based Fault Detection
  - Model based fault detection
    - Monitoring system -> Current performance
    - System model -> Expected performance
    - Threshold difference
  - Fault signature (parameters)
    - Vector of system and application parameters
    - Magnitude of deviation -> Fault intensity

Control Approach



Control Input

Available Control Options

Management Actions	Recovery Actions
Changing Database Buffer Sizes (e.g. 128MB, 256MB, and etc.)	Offline Database Rebuilding
	Online Database Reorganization

Criteria to Select the Most Optimal Control Option?

Control Input

Selection Criteria

Management Actions	Benefits	Drawbacks
Changing Database Buffer Sizes (e.g. 128MB, 256MB, and etc.)	Bigger Buffer = Less I/O Delay	Bigger Buffer=Potential Resource Shortage

Each control option has both benefits and drawbacks: a tradeoff

Control Input

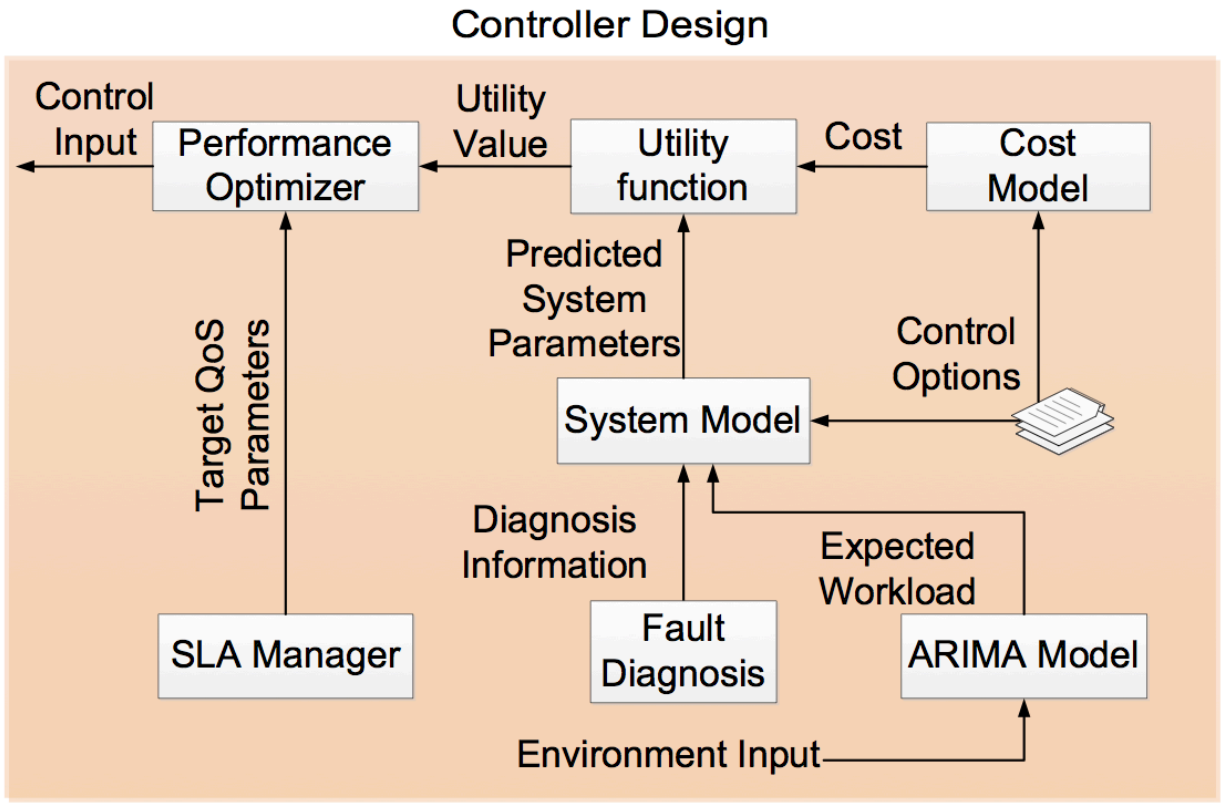
Selection Criteria

Recovery Actions	Benefits	Drawbacks
Offline Database Rebuilding	Fast Recovery	Service Blackout
Online Database Reorganization	Service Available	Slow Recovery

Each control option has both benefits and drawbacks: a tradeoff

MISSISSIPPI STATE UNIVERSITY

Controller Structure



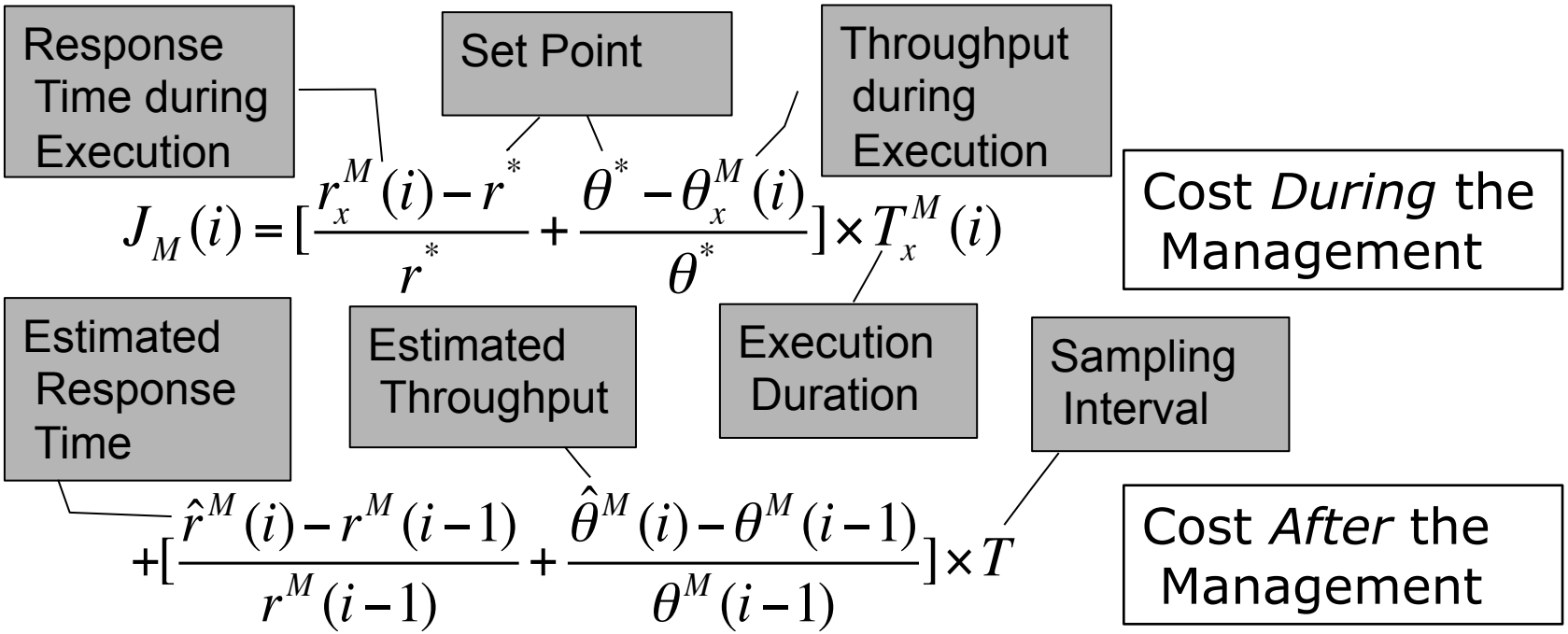
MISSISSIPPI STATE UNIVERSITY

Utility Cost Model

- Quantify the benefits and drawbacks of a control option
- Evaluate the system performance in terms of QoS and SLA violation
- Treat management and recovery actions separately

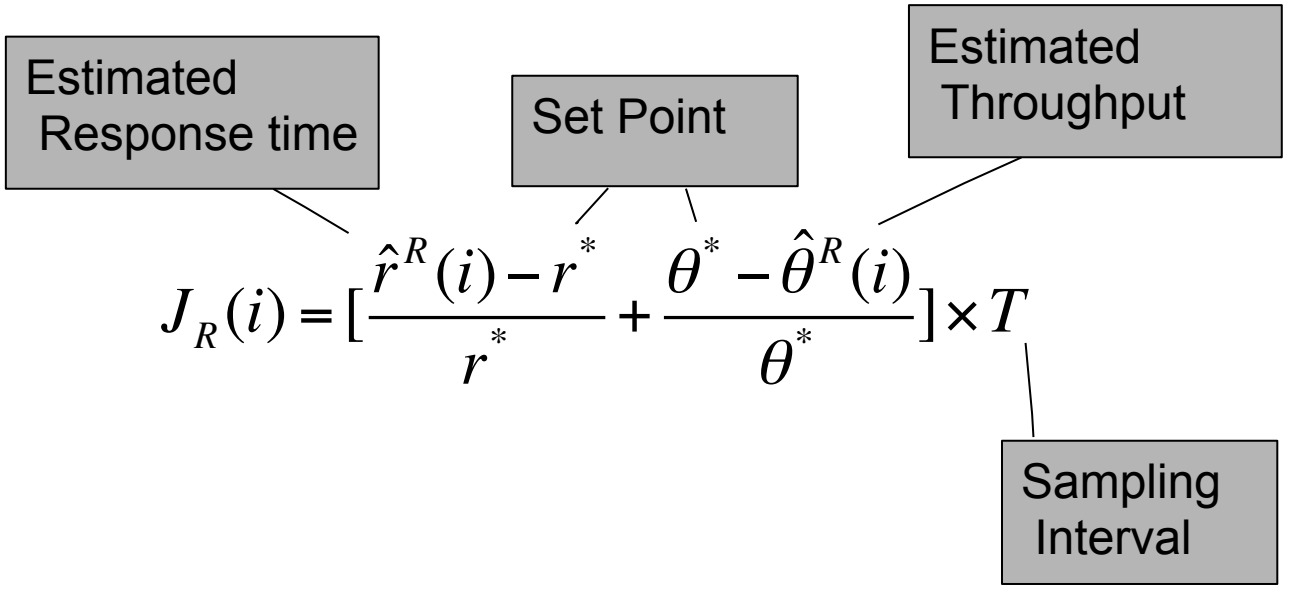
Utility Function

For management actions



Utility Function

For recovery actions



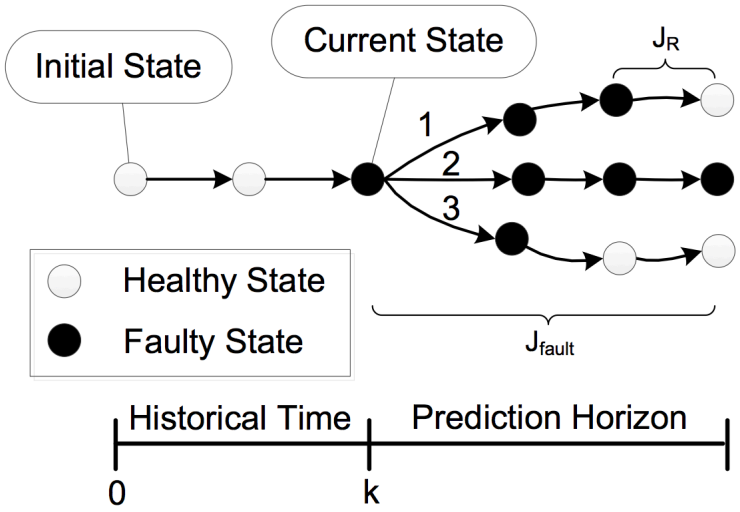
Predictive Control

Predictive Control: Select the Minimum-cost Control Plan

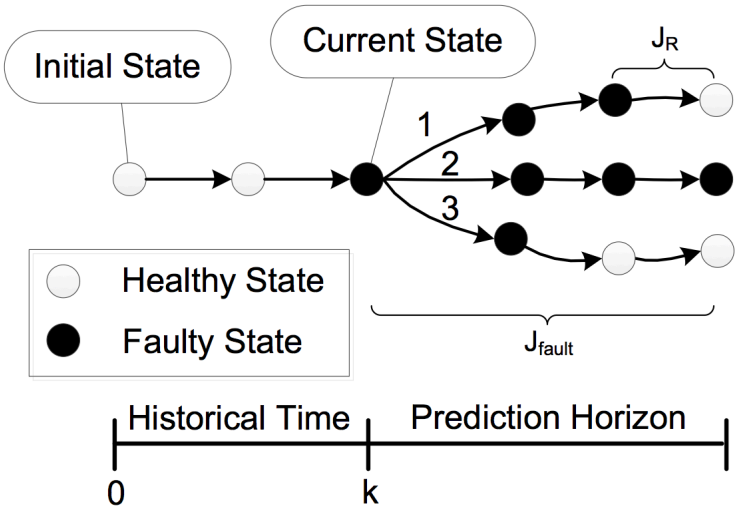
Optimization Objective:

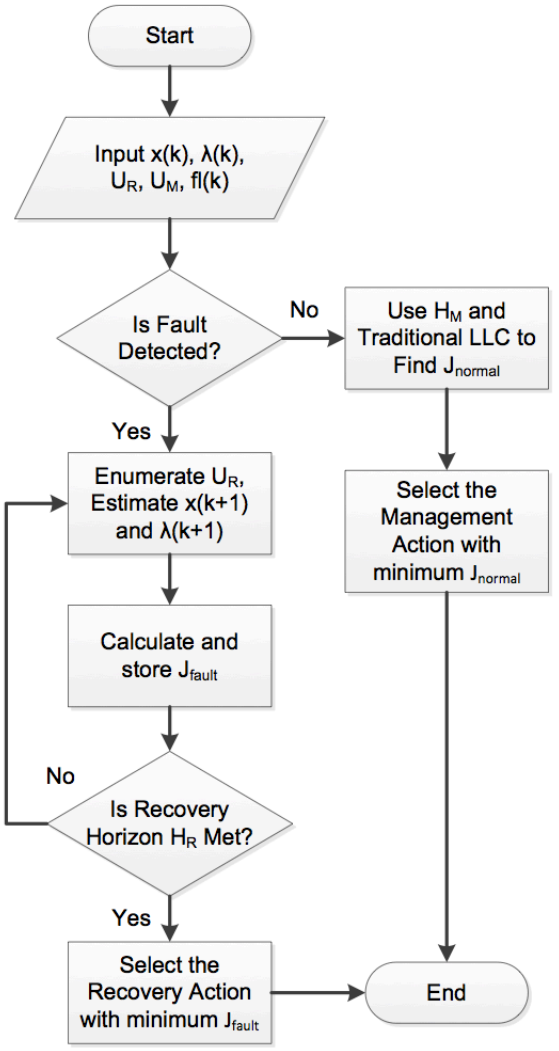
1. Minimize $\sum_{i=k}^{k+m-1} [W_M(i) J_M(i)]$

2. Minimize $W_R(j) \sum_{i=k}^{k+n-1} J_R(i) / n$



Predictive Control Algorithm





MISSISSIPPI STATE UNIVERSITY

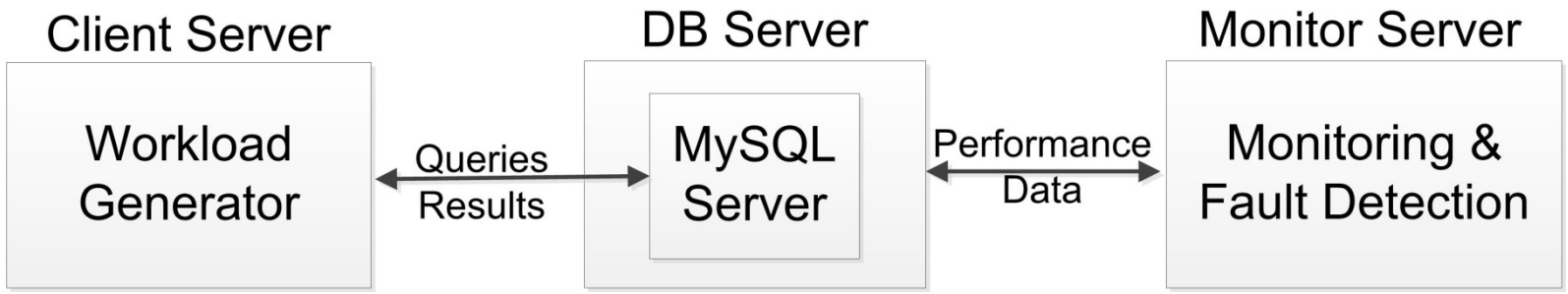
Case Study

The 8th International Workshop on Feedback Computing
San Jose, CA, USA
June 25th, 2013



MISSISSIPPI STATE UNIVERSITY™

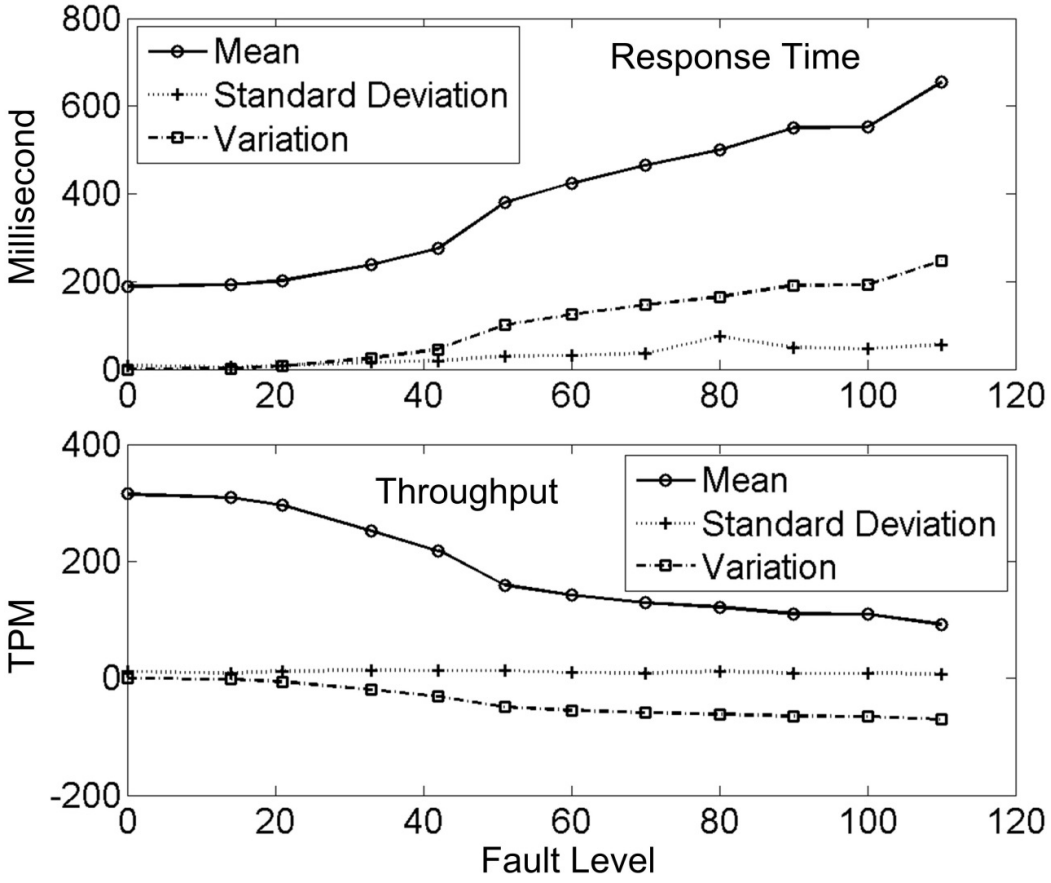
Case Study



Experimental Setup

MISSISSIPPI STATE UNIVERSITY

Case Study



Example Simulation Results

MISSISSIPPI STATE UNIVERSITY

System Parameters

- Fragmentation detection using average response times

FL	Average query response time (ms)		
	Mean	Std. Dev.	% variation
0	189	8	0.00
14	192	5	1.59
21	202	8	6.88
33	238	16	25.93
42	275	19	45.50
51	380	30	101.06
60	425	31	124.87
70	466	36	146.56
80	501	75	165.08
90	551	50	191.53
100	552	46	192.06
110	655	56	246.56

System Parameters

Arrival rate	Throughput	Avg. Response Time (ms)
$\lambda<350$	$0.9\times\lambda+8.9$	224
$\lambda\geq350$	$0.01\times\lambda+318.1$	189

System model for fragmentation level 0

MISSISSIPPI STATE UNIVERSITY

Conclusion

- The system model is developed to facilitate the fault management
- The detection method can provide the level of fragmentation
- The model-based fault management framework has been applied to solve the database fragmentation problem

Future Work

- Workload: add composition variation
- A fully autonomic control process, with more
  - Recovery actions
  - Management actions
- Applied on a distributed database system

MISSISSIPPI STATE UNIVERSITY

Thank you!

The 8th International Workshop on Feedback Computing
San Jose, CA, USA
June 25th, 2013



MISSISSIPPI STATE UNIVERSITY™