Outline

- Control-theoretic Framework
- Service delay control on Web servers
- On-line data migration in storage servers
- ControlWare: adaptive QoS control middleware

Online Data Migration in Storage Servers

- Enterprise storage servers need to move data
  - System expansion
  - Application changes
- **Always-on:** e-business, global data centers
  - **Online** data migration

State of Practice

- Need to bound impact on applications!
- Slow I/O’s!!!
- E-mail server; DB...
- SAN
- New device
- Storage system
- Script
- Submover
- HP-UX LVM

The Problem

- Execute a given migration plan on-line
- Challenges
  - Keep data consistent
  - Bound impact on application performance
  - Complete migration quickly

Adaptive solution

- Feedback control loop: adapts migration speed based on application I/O latency
  - Enforce latency contract: Bounded average I/O latency
  - Complete migration in shortest time allowed by contract
- Standard control-theoretic design
  - Systematic methodology
  - Robust, analytically proven performance
- Handle different workloads and devices

Aqueduct

- E-mail server; DB...
- SAN
- I/Os
- Submover
- HP-UX LVM
- Application Latency Contract
- Migration plan
Monitor

- Measure applications' average I/O latency of each store in the last sampling window
  - Current implementation: trace replayer directly monitors I/O latencies
  - Can interface with performance monitoring tools (HP Openview)

Controller

- Compute error for each store $i$
  $$E_i(k) = P \cdot L_C - L_i(k)$$
  $0 < P < 1$: safety margin, related to burstiness
  $k$: represents the $k^{th}$ sampling window
- Compute worst error
  $$E_{\text{max}}(k) = \min\{E_i(k)\}$$
- Integral controller computes new submove rate:
  $$R_m(k) = R_m(k-1) + K \cdot E_{\text{max}}(k)$$
  Control gain $K$: aggressiveness of rate change

Actuator

- Problem: fine-grained control of migration speed using HP-UX LVM
  - Divide store into small (32 MB) substores (LVs)
  - Submover moves substore using LVM silvering

  ![Silvering Diagram]

- Actuator enforces a submove rate by sleeping

  ![Sleeping Diagram]

Tuning controller parameters

- Approximate linear model
  $$\text{VL}(k+1) - \text{VL}(k) = G(R_m(k) - R_m(k-1))$$

- System profiling: Estimate $G$

- Construct transfer function

- Control Analysis
  - Compute $K$
  - Satisfy

- Stability
- Tracking: $\text{VL}(k) = P \cdot L_C$ in steady state
- Setting time

Experimental setup

- Aqueduct
  - HP-UX 11 & LVM
  - HP 9000-N4000 Server
  - 8 440MHz processors

- Fibre Channel

- Openmail I/O Trace

- Enterprise-scale storage server

Experiments

- Baselines: no sleeping between (sub)moves
  - Whole-store: move one store at a time
  - Sub-store: move one subtree at a time

- Constant: steady Poisson streams
  - Replace Logical Unit; migrate three 640-MB stores.

- Openmail: trace of an enterprise e-mail server running HP Openmail
  - Add Logical Unit; migrate a 1854 MB store and a 96 MB store
Chenyang Lu

Measure G $\rightarrow$ Tune K

Process gain G: the slope of the curves
Control gain K

Constant: $K = 1.09$
Openmail: $K = 0.36$

Openmail: victim latency

Openmail: latency

Aqueduct uniformly better than baselines, but ...

Openmail: latency & submove rate

Load highest on new LU towards end of migration
By design, submove rate must be 1 or higher $\Rightarrow$ controller is working correctly

Openmail: average latency

Openmail: latency CDF

Quality of Service in Unpredictable Computing Environments
Related work

- Simpler versions of the problem
  - Take (parts of) system offline
  - Migrate data in “quiet periods”
- Silvering in Logical Volume Manager [HP-UX LVM, VxVM]: maintain data consistency, no QoS guarantees
- Proportional I/O scheduling: hard to handle unpredictability
- MS Manners: no guarantees to important tasks
- Control-theory-based systems: distributed visual tracking, Web servers, e-mail server, database real-time processor scheduling ...

Summary

- Migration must be executed adaptively
  - Average I/O latency reduced by 76%
  - Contract violation ratio reduced by 78%
- Aqueduct is neither overly aggressive
  - Average victim latency 15% lower than latency contract
- Not overly conservative

Future

- More detailed sensitivity analysis
- Self-tuning controller
- Multi-dimensional QoS contracts

References