Performance in ASCENS

DEPARTMENT OF DISTRIBUTED AND DEPENDABLE SYSTEMS
FACULTY OF MATHEMATICS AND PHYSICS
CHARLES UNIVERSITY IN PRAGUE

http://d3s.mff.cuni.cz
Brief Tasks Overview

**T1.4** SCEL Code Bindings

**T4.4** Performance Models

**T5.4** Implementation Conformance

**T6.3** Tools
  - Bidirectional Mapping Code SCEL
  - Instrumentation And Modeling
  - Implementation Conformance

**T7.2.5** Performance Aware Cloud
  - Resource Allocation
  - Performance Awareness
  - Performance Prediction
Cloud Use Case

Few possible events
- Configuration change
- Poor resource performance
- Poor application performance

Few possible reactions
- Application architecture adjustment
- Application component migration
- Cloud structure adjustment
Cloud Use Case

*Initial thoughts use log demo*

**Positives**
- Reasonably simple to begin with
- Resembles other scenarios
  - Shared database
  - Application monitoring

**Negatives**
- Does not really benefit from using cloud
- Coarse migration granularity
Log Demo Sketches

Log Clients
- Get reference to logger from knowledge
- Try to log, repeat everything on failure

Log Coordinator
- Publish knowledge about logger location
- Plus possible autonomy
  - Keep track of node utilization (node ensemble)
    - Computing
    - Storage
  - Keep track of traffic volumes (how ?)
  - Direct migration
Cloud Use Case

Avoid global knowledge

- Neighbor awareness coupled with DHT?
  - Truly local neighbors (communication, migration)
  - Few other neighbors (resiliency, local extremes)
- Introduce noise into decisions

Benefits mostly in engineering?

- Difficult to find completely new optimizations
- Perhaps we can write existing ones better
- Emergence?
Cloud Use Case

Provide **standard knowledge infrastructure**

- Nodes know their load and neighbors
- Applications publish their goals
  - Difficult to express competition

**Ensembles mediate** migration

- Shutdown Migration Ensemble
  - Migrate on seeing knowledge of imminent shutdown
- Node Utilization Ensemble
  - Migrate on seeing low or high node utilization
- Application Efficiency Ensemble
  - Keep components of distributed application together
- Combinations are the interesting part
Knowledge is not for free

- Performance awareness is knowledge
- Instrumentation and monitoring has overhead
- Autonomy balances optimization benefits with awareness overhead

This is **not easily expressible** in SCEL?

- No cost for using knowledge
- All knowledge always visible
How to connect cloud with components?

- Writing new components unrealistic
- Surrogate components?
- Wrapper layers?

Run components in simulation box but connect the simulation to real cloud?
Cloud And DEECO

How is knowledge distributed?

- Propagation throughput and latency
- Reliability when facing failures
  - Writing rules extremely tricky
  - Should connections use knowledge?
- Visibility in large ensembles
- Knowledge on demand (push vs pull)

How much can applications use knowledge?
Practical ensemble structure

- Membership awareness
  - Tailored knowledge
  - Are we too strict here?

- Ensemble instance interpretation

- How is membership updated?
  - Ensembles based on location
  - Ensembles based on activity coordination
  - Ensembles based on interest in knowledge
Some Things Do Not Rely On SCEL
Awareness Rules

Evaluate SPL expressions at runtime

Benefits

- More readable code
- Robust measurements
- Optimizations (cache results)
- Wild (compare measurements with models)

Challenges

- Naming for measurement points
- Instrumentation at runtime
- Managing overhead
Use DISL instrumentation framework

Benefits
- Can actually work

Challenges
- Runtime join point specification
- Loading and unloading
- Data aggregation
Performance Models

Empirical models of middleware scalability

Benefits
- Better estimates of maximum utilization
- Detection of anomalous behavior

Challenges
- Workload combinations
- Monitoring overhead
- Portability
Next Steps

• Test evaluating SPL at runtime

• Try to use DISL for instrumentation

• Construct middleware performance models

• Knowledge structure for cloud infrastructure