Performance Prediction of Component-based Applications

Jenny Liu
NICTA Empirical Software Engineering Program
Research in Empirical Software Engineering Program

• Software Process Modelling
• Software Knowledge Management
• Software Architecture
• Requirements Engineering
• Project Risk Management
• Process Improvement
• Software Product Line Development
Design for High Performance and Scalability of Web-based Applications

- Workload characterisation
- Determine the components most affected
- Architectural design/implementation
- Evaluation
Performance Prediction of Component-based Applications (PPCB) : The Challenge

- Design software architectures to meet specific performance/scalability requirements
  - Diversity of the component-technologies
  - Multiple architectural design choices
  - Many tunable parameters of the runtime environment
  - Different application behavior

- The performance of a deployed component-based application
  - the behavior of its application-specific components and their interactions
  - the particular implementation of the component infrastructure, or container
  - the selected configuration settings for the container
  - the attribute settings of both the application components and the infrastructure components
  - the simultaneous request load experienced at any given time by the application
Why is this important and useful?
- Many large scale IT applications fail or experience budget blow outs due to inability to meet performance/scalability needs

What’s our solution’s competitive edge?
- anticipating the performance of the eventual solution before it has been built – lower costs and risks
- developing a methodology that allows the system architect to make design trade-offs between alternative architectures and implementation approaches

How do we do this?
- We build a quantitative performance model for a proposed system design.
- The model requires inputs from an application-independent performance profile of the underlying component technology platform, and a design description of the application.
The Framework of PP-CB Approach

- A model-based prediction approach
  - Modeling infrastructure
  - Calibrating component architecture
  - Characterizing application behavior
  - Benchmarking middleware platform
  - Populating model parameters
- Requirements of PP-CB approach
  - Allow performance prediction under architecture changes
  - Reveal performance bottlenecks
  - Accurate
  - Cost effective
PPCB : The Solution

The imagination driving Australia’s ICT future.

Infrastructure model: a queueing network model for a single J2EE server using JMS

Performance Profile
(benchmarking)

Architecture model
(calibrating)

Service demand

Performance Model
(populating)

Performance Prediction

Performance Profile
(benchmarking)
PPCB Toolset : From Design to Prediction

- Design software engineering tools that hide the complexity of the modeling and analysis steps in our performance prediction approach from an architect
  - MDA-based automatic benchmark suite generation
  - Annotating performance attribute using UML2 Profile
  - From UML Profile to Performance Model Interchange Format
  - A Eclipse plug-in for the analysis engine
The imagination driving Australia's ICT future.

PPCB Toolset: From Design to Prediction
Strong evidence has been accumulated over last year that these techniques actually work!

We have several case studies of predicting J2EE applications:

- Two example applications (Stock-Online and J2EE 1.4 Tutorial Example Duke’s Bank)
- Two J2EE application servers (BEA WebLogic and Borland Enterprise Server)
- Three alternative architecture designs for data access
- Both synchronous and asynchronous communication protocols
- Infrastructure with a single server or a server cluster
- Statistical method to evaluate the prediction approach

We are also extending this approach to the .Net platform.
• IBM’s Autonomic Computing Initiative—take the ‘operator’ out of the loop

• Model-based architecture-driven adaptation for middleware platforms
  – The invention of predictive models for self-managing, self-optimizing applications
  – The invention of new software architectures to support self-managing, self-optimizing applications
  – The invention of mechanisms to enable dynamic monitoring and adaptation policies to be simply specified by the application designer
Models and Extensible Architectures
For Middleware Platform: Overview

- **Target**
  - Platform technologies to support next generation IT applications
  - Support NICTA Priority Research Challenges

- **Method**
  - Devise and empirically validate new models and architectures for building self-monitoring and dynamically re-configurable application server technologies
  - Use commercially available application servers as test-beds

- **Design and build infrastructure**
  - Predictive models
    - PPCB models are first example
  - Monitoring and feedback architecture framework
  - Policy specification language/tools
Models and Extensible Architectures For Middleware Platform: Some Issues

- Automatic workload characterization
  - Workload cannot be manually characterized
  - Applying clustering algorithm or fuzzy logic?
  - Determine the parameters for characterization
- Architecture designs
  - Adaptive behavior as application server plug-in
  - Computational models
- Minimize the overhead of adaptive behavior
  - Sample size and sampling period
  - Optimization of the control loop
Thank You