Challenges in Service-Oriented Networking

Bob Callaway

North Carolina State University
Department of Electrical and Computer Engineering
Ph.D Qualifying Examination

April 14, 2006

Advisory Committee: Dr. Michael Devetsikiotis, Dr. Yannis Viniotis, Dr. Harry Perros, Dr. Andy Rindos, Dr. Adolfo Rodriguez, Dr. Mihail Sichitiu
Presentation Outline

1. Application-Aware Networking
   - Active Networks
   - Overlay Networks
   - XML: A Standard for Data Interoperability

2. Service-Oriented Networking
   - Service-Oriented Architectures
   - Network Service Intermediaries

3. Literature Review: Internet Server Design
   - Admission Control
   - Concurrency Mechanisms

4. Performance Analysis of Different Concurrency Mechanisms
   - Experimental Design
   - Results of Experiments

5. Conclusions and Future Work
Section Outline

1 Application-Aware Networking
   - Active Networks
   - Overlay Networks
   - XML: A Standard for Data Interoperability

2 Service-Oriented Networking
   - Service-Oriented Architectures
   - Network Service Intermediaries

3 Literature Review: Internet Server Design
   - Admission Control
   - Concurrency Mechanisms

4 Performance Analysis of Different Concurrency Mechanisms
   - Experimental Design
   - Results of Experiments

5 Conclusions and Future Work
What is Application-Aware Networking?

**Definition**

Application-aware networking is an emerging technology that promises to provide increased end-to-end system performance for next-generation applications and networks by providing differential treatment of traffic dependent on application data.

**Why do we need it?**

- Application performance is limiting factor in improving end-to-end performance
- Integration of heterogeneous distributed systems is difficult and costly
- Complex algorithms are needed to secure enterprise applications
Active Networks

- Attempted to add application layer functionality by executing user-supplied bytecode in “smart” packets in specific active nodes (programmable routers, switches)
- Suffers from issues of security, resource allocation, performance, and cost of deployment
Overlay Networks

Overview

- Consist of peer nodes that self-organize into a distributed data structure based on application criteria.
- Goals are to alleviate the effects of slow or sporadic deployment of new services in the Internet, and to directly provide application-level functionality that is out-of-scope for the underlying network.
Lacking an Open Standard for Data Interoperability

Why is this an issue?

- Active and overlay networks operate under the presumption that the underlying protocol is proprietary and specific to the application.
- Difficult to build a scalable and robust optimization strategy within active or overlay networks for numerous proprietary protocols.
- Management of updates to these protocols is time consuming.
The Extensible Markup Language (XML)

What is XML?

XML is an open standard for representing self-describing application data in a textual format, enabling heterogeneous systems to easily operate on the data.

```
<xml version="1.0">
  <book>
    <title>Green Eggs and Ham</title>
    <author>Dr. Seuss</author>
  </book>
</xml>
```

Estimated Percentage of XML in Network Traffic
XML as a Catalyst for Adding Intelligence in the Network

Why should we try to make the network fabric application-aware again?

- The underlying assumptions of active and overlay networks have changed
  - Advances in hardware, software, networking technologies
  - The network fabric can now support application-awareness
- An open standard for application-to-application communication now exists and is widely adopted
  - XML enables services and SOA, and much more
Section Outline

1. Application-Aware Networking
   - Active Networks
   - Overlay Networks
   - XML: A Standard for Data Interoperability

2. Service-Oriented Networking
   - Service-Oriented Architectures
   - Network Service Intermediaries

3. Literature Review: Internet Server Design
   - Admission Control
   - Concurrency Mechanisms

4. Performance Analysis of Different Concurrency Mechanisms
   - Experimental Design
   - Results of Experiments

5. Conclusions and Future Work
Service-Oriented Architectures

Issues in IT Architectures
- Rapid Pace of Change
  - Moore’s Law exacerbates this
- Integration of Heterogeneous Systems

Loosely Coupled Services
- Represent a reusable business function
- Removes dependencies on implementation specifics through standardized interfaces
- Standardized interfaces enable the flexibility of SOA
What are Web Services?

- XML-based standard for application communication over the Internet
- Based upon a find/publish/use approach
- Many standards already exist for WS (e.g. WS-Security)

```xml
http://mygateway.com/services/createOrder
<order>
  <id>1234</id>
  <customer_id>AB35</customer_id>
  <line>
    <item>
      <part_no>127.87A</part_no>
      <quantity>2</quantity>
    </item>
  ...
```
Enterprise Service Bus (ESB)

What is an ESB?

- An ESB virtualizes the enterprise resources, allowing the business logic of the enterprise to be developed and managed independently of the infrastructure, network, and provision of those business services.

- The main functions of an ESB are to convert underlying transport protocols, transform message formats, and intelligently route requests made to services within an SOA.
Service-Oriented Networking (SON)

What is SON?
- SON enables application-awareness in the network, overcoming the previous limitations and constraints imposed by active and overlay networks, to improve performance and ease the integration of heterogeneous systems within service-oriented environments.

What benefits does SON provide?
- Service Virtualization
- Locality Exploitation
- Increased Manageability
Network Service Intermediaries (NSI)

What is an NSI?

- Network service intermediaries (NSI) are XML-enabled application-aware network appliances which are configured with accelerated hardware for optimized security and XML operations.

Examples of NSI

- DataPower XI50
- Cisco AON Blade
The NSI as a Platform for ESB Functionality

What would types of functions would an NSI perform?
- Functional Offloading
- Service Integration
- Intelligent Routing
NSI - Architectural Design Goals

- If the execution architecture is poorly implemented or architected, the NSI will become a bottleneck for the infrastructure and will actually degrade the performance of the overall system.
- We define three goals for the NSI architecture: scalability, robustness, and adaptivity.
Section Outline

1. Application-Aware Networking
   - Active Networks
   - Overlay Networks
   - XML: A Standard for Data Interoperability

2. Service-Oriented Networking
   - Service-Oriented Architectures
   - Network Service Intermediaries

3. Literature Review: Internet Server Design
   - Admission Control
   - Concurrency Mechanisms

4. Performance Analysis of Different Concurrency Mechanisms
   - Experimental Design
   - Results of Experiments

5. Conclusions and Future Work
Keys to Scalability, Robustness, and Adaptivity

Important Architectural Components

- **Admission Control Algorithms**
  - Explicitly enumerate the process of admitting (and denying) requests according to specified criteria in order to ensure that the protected resource is never overloaded.

- **Concurrency Mechanisms**
  - Operating system structures designed to provide the virtualization that multiple programs can be executed simultaneously on a single processor.
  - Manage how multiple programs can compete for the shared processor.
History of Admission Control

- The initial surge in research on admission control techniques began with the popularity of connection-oriented networks such as ATM networks.
- Admission control algorithms for the application layer began to appear, due to the increasing popularity of mission-critical applications such as e-commerce are deployed on servers that are subject to a dynamic and complex workload.
Web Server Admission Control

Algorithms Utilizing Network Bandwidth

  - Based upon concept of user-perceived bandwidth = \( \frac{\text{bytes transferred}}{\text{response time}} \)
  - Combines TCP-SYN policing and HTTP header analysis

Algorithms Utilizing Processor Utilization

- L. Cherkasova and P. Phaal, Session-based admission control: A mechanism for peak load management of commercial web sites, IEEE Transactions on Computers, 2002
  - Created static and adaptive MBAC algorithm based upon session length
### Algorithms Utilizing Response Times

  - Requests are classified into a service class based on latency target
  - Adapt token-bucket rate relative to 90th percentile of response times

### Algorithms Utilizing Other Metrics

  - Create non-linear admission control model for a G/G/1 queueing system
  - Supports various objective functions in profit maximization
Lauer/Needham argued in a 1978 paper that threads and event-driven systems are duals of one another. They stated that the correct concurrency mechanism depends on the problem and implementation constraints.
Thread-based Concurrency Mechanisms

Thread-Based Literature

  - Argued that poor performance of threads was due to poor implementation

  - Described a threading library which out performs event-driven systems

  - Developed a dynamic optimization method to determine the optimal size of the thread pool

  - Developed a Markov-Chain based model of a multithreaded system
Event-Driven Concurrency Mechanisms

Event-Driven Literature

  - Investigated the use of event-dispatching as an alternative to threading-based concurrency methods

  - Compares the implementation of a multithreaded web server with an event-driven implementation; makes first attempt at a hybrid concurrency environment
Hybrid Concurrency Mechanisms

Hybrid Concurrency Literature

  - An architecture that separates functions within applications into stages, which each have their own thread pool

  - Present an architecture that supports adaptively reconfiguring the concurrency policy

  - Proposed that stages should be able to be adaptively combined or split, which would dynamically change the concurrency model

  - Argues that the control should be an orthogonal extension to the framework, rather than tightly integrated
Section Outline

1. Application-Aware Networking
   - Active Networks
   - Overlay Networks
   - XML: A Standard for Data Interoperability
2. Service-Oriented Networking
   - Service-Oriented Architectures
   - Network Service Intermediaries
3. Literature Review: Internet Server Design
   - Admission Control
   - Concurrency Mechanisms
4. Performance Analysis of Different Concurrency Mechanisms
   - Experimental Design
   - Results of Experiments
5. Conclusions and Future Work
Experimental Motivation

Goals

- Create a prototype NSI and implement components that would likely be a part of an actual NSI
- Gain insight on how the choice of a concurrency mechanism affects overall system performance

Experiment Details

- Comparing Thread-Per-Request, Bounded Thread Pool, Event-Driven, and SEDA Concurrency Mechanisms
- Implemented in JCyclone, an open-source Java implementation of SEDA
- Performance Metrics: Average Response Time, Average Throughput, CPU Utilization
Experimental Design

Nonblocking Experiment

Blocking Experiment
**Results - Nonblocking Experiments**

![Graph showing average response time for different并发机制](image)

- **Average Response Time** vs **Number of Concurrent Users**
- **Average Response Time (ms)**
- **TPR**
- **BTP**
- **ED**
- **SEDA**

The graph illustrates the performance of different concurrency mechanisms under nonblocking implementation. The x-axis represents the number of concurrent users, while the y-axis shows the average response time in milliseconds. Different lines and markers represent various concurrency mechanisms, with TPR, BTP, ED, and SEDA having distinct characteristics in terms of response time at varying concurrency levels.
Results - Nonblocking Experiments

![Graph: Average Throughput - Nonblocking Implementation]

- **Average Throughput (requests/sec)**
- **Number of Concurrent Users**
- **TPR**
- **BTP**
- **ED**
- **SEDA**

The graph shows the average throughput for different concurrency mechanisms as the number of concurrent users increases. The mechanisms compared are TPR, BTP, ED, and SEDA.
Results - Nonblocking Experiments

![Average CPU Utilization - Nonblocking Implementation](image)

- **TPR**
- **BTP**
- **ED**
- **SEDA**

- **Average CPU Utilization (% busy)**
- **Number of Concurrent Users**
- **Average CPU Utilization**

- | Number of Concurrent Users | TPR  | BTP  | ED   | SEDA |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Results - Blocking Experiments

![Average Response Time - Blocking Implementation](chart.png)

- TPR
- BTP
- ED
- SEDA-even
- SEDA-weighted

<table>
<thead>
<tr>
<th>Number of Concurrent Users</th>
<th>Average Response Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1000</td>
<td>0.5</td>
</tr>
<tr>
<td>2000</td>
<td>1</td>
</tr>
<tr>
<td>3000</td>
<td>1.5</td>
</tr>
<tr>
<td>4000</td>
<td>2</td>
</tr>
<tr>
<td>5000</td>
<td>2.5</td>
</tr>
<tr>
<td>6000</td>
<td>x 10^5</td>
</tr>
<tr>
<td>7000</td>
<td></td>
</tr>
<tr>
<td>8000</td>
<td></td>
</tr>
</tbody>
</table>
Results - Blocking Experiments

Average Throughput – Blocking Implementation

- TPR
- BTP
- ED
- SEDA-even
- SEDA-weighted

Number of Concurrent Users

Average Throughput (requests/sec)
Results - Blocking Experiments

![Graph showing Average CPU Utilization - Blocking Implementation](image)

- TPR
- BTP
- ED
- SEDA-even
- SEDA-weighted
Section Outline

1. Application-Aware Networking
   - Active Networks
   - Overlay Networks
   - XML: A Standard for Data Interoperability

2. Service-Oriented Networking
   - Service-Oriented Architectures
   - Network Service Intermediaries

3. Literature Review: Internet Server Design
   - Admission Control
   - Concurrency Mechanisms

4. Performance Analysis of Different Concurrency Mechanisms
   - Experimental Design
   - Results of Experiments

5. Conclusions and Future Work
The adoption of XML and acceptance of Web Services and SOA have enabled network components to make intelligent decisions based on application data.

Our review of the Internet server design literature provides insight into two large components of a scalable, robust, and adaptive NSI: admission control and concurrency mechanisms.

Our experiments validated our intuition and emphasized the importance of concurrency mechanisms in the overall performance of an NSI.

We believe that SON provides exciting new multidisciplinary research opportunities in service-oriented computing, hardware, software, and networking which could have dramatic effects on the development of emerging network services.
Future Work

- An adaptive and hybrid concurrency mechanism that uses online measurements to determine the correct concurrency mechanism for each component and for the system as a whole.
- Admission control that balances the complex tradeoff between system utilization and performance.
- Investigating how SON affects other architectures, such as grid computing or P2P systems.

Publications