Flexible Scheduling in Middleware for Distributed Rate-Based Real-Time Applications
Christopher D. Gill
Dissertation Supervisors: Dr. Ron K. Cytron, Dr. Douglas C. Schmidt
Department of Computer Science
Washington University, St. Louis, MO
cdgill@cs.wustl.edu

Historical Challenges
- Many mission-critical distributed applications require real-time QoS guarantees
  - e.g., combat systems, online trading, telecom
- Building QoS-enabled applications manually is tedious, error-prone, & expensive

New Challenges
- Many mission-critical distributed applications require real-time QoS guarantees
  - e.g., combat systems, online trading, telecom
- Building QoS-enabled applications manually is tedious, error-prone, & expensive
- Conventional middleware does not support real-time QoS requirements effectively

Overview of Research Areas

<table>
<thead>
<tr>
<th>Technical Challenge</th>
<th>Research Approach</th>
<th>Research Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Efficient and Safe Systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customizable Middleware</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nimble Adaptation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Awareness</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inclusive Systems Approach</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

These technical challenges raise crucial systems issues in both theoretical and empirical dimensions

Motivating Applications

Boeing Bold Stroke Middleware Infrastructure Platform
- Operations well defined
- Event-mediated middleware solution
- Previous-generation systems static
- Next-generation systems dynamic and adaptive

Limitations With Existing Approaches

Historically, each application has solved scheduling on its own
- Tedious, error-prone
- Costly over system lifecycles
- Single-paradigm approaches

Current middleware lacks hooks for key domain-specific features, e.g.:  
- Optimized integration with higher level managers
- Hybrid static-dynamic scheduling strategies
- Strategies built from primitive elements
- Adaptive domain-specific optimizations

Research Contributions

Systems Architecture and Framework
- Extends open-source middleware

Patterns for Adaptive Scheduling
- Capture design experience and solutions

Empirical Evaluation of Adaptive Scheduling
- Practical benefits for real-world applications
Integration with Event Service

- Separates dispatching mechanism from scheduling policy
- Dispatcher consults runtime scheduler for priorities
- Flexibility for different scheduling policies

Research Approach: the *Kokyu* Flexible Middleware Scheduling/Dispatching Framework

Scheduler assigns rates & priorities per topology, scheduling policy

- Defines necessary dispatch configuration

Implicit projection

- Of specific scheduling policy into generic dispatch infrastructure

Rate and priority assignment policy

Application specifies characteristics

- e.g., criticality, periods, dependencies

Scheduler consults run-time scheduler for priorities

Flexibility for different scheduling policies

Dispatcher consults run-time scheduler for priorities

Greater Utilization with Criticality Isolation

<table>
<thead>
<tr>
<th>Technical Challenge</th>
<th>Research Approach</th>
<th>Research Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Efficient and Safe Systems</td>
<td>Arbitrary strategies that hybridize static/dynamic scheduling/dispatching</td>
<td>Increased utilization, critical operations still meet their deadlines</td>
</tr>
<tr>
<td>Customizable Middleware</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nimble Adaptation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Awareness</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inclusive Systems Approach</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Safety: Meet Critical Deadlines

Dynamic

Static

Increased utilization, critical operations still meet their deadlines

Arbitrary strategies that hybridize static/dynamic scheduling/dispatching

No One Strategy is Optimal

<table>
<thead>
<tr>
<th>Technical Challenge</th>
<th>Research Approach</th>
<th>Research Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Efficient and Safe Systems</td>
<td>Arbitrary strategies that hybridize static/dynamic scheduling/dispatching</td>
<td>Increased utilization, critical operations still meet their deadlines</td>
</tr>
<tr>
<td>Customizable Middleware</td>
<td>Dispatching composed from primitive elements</td>
<td>Supports tailored &quot;fit&quot; of scheduling/dispatching</td>
</tr>
<tr>
<td>Nimble Adaptation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Awareness</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inclusive Systems Approach</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Case for Multi-Paradigm Scheduling

Solution: Compose Scheduling Heuristics from Dispatching Primitives

Unobtrusive Monitoring and Control Feedback

Solution: Kokyu Real-Time Metrics Monitoring Framework

Experimental Test-Bed

Popular Scheduling Strategies

<table>
<thead>
<tr>
<th>Technical Challenge</th>
<th>Research Approach</th>
<th>Research Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Efficient and Safe Systems</td>
<td>Arbitrary strategies that hybridize static/dynamic scheduling/dispatching</td>
<td>Increased utilization, critical operations still meet their deadlines</td>
</tr>
<tr>
<td>Customizable Middleware</td>
<td>Dispatching composed from primitive elements</td>
<td>Supports tailored “fit” of scheduling/dispatching</td>
</tr>
<tr>
<td>Nimble Adaptation</td>
<td>Integrated rate/ratio priority selection mechanisms</td>
<td>$O(n \log n)$ bound on adaptation</td>
</tr>
<tr>
<td>Performance Awareness</td>
<td>Time and space efficient data collection and storage framework</td>
<td>Provides run-time observable info for control, post-analysis</td>
</tr>
<tr>
<td>Inclusive Systems Approach</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Application
- Research Version of Operational Flight Program for AV-8B Aircraft
- Added navigation route computations to ramp non-critical load
- Added critical and non-critical computations to inject execution time jitter

Middleware
- The ACE ORB (TAO)
- TAO Real-Time Event Channel
- Kokyu Framework: Scheduling, Dispatching, and Metrics

Operating System and Hardware
- VxWorks RTOS on the PPC boards
- 200 MHz Motorola PPC 604 card
- two 100 MHz Dy4-177 PPC 603 cards
- Dy4-783 memory mapped display processor
- Commercial VME-64 chassis with all boards
- Switched Ethernet, MIL-STD-1553 MUX Bus on one Dy4-177 card

Rate Monotonic Scheduling (RMS)
- Assigns thread priorities by rate
- Operations at each priority handled in FIFO order

RMS + Minimum Laxity First (MLF)
- Critical operations managed as in RMS
- Non-critical operations managed in single lowest priority
- Non-critical operations handled in minimum laxity (slack time) order

Maximum Urgency First (MUF)
- Thread priority per criticality level
- Operations in each priority level handled in laxity order
Experimental Benchmarks

Measure Performance of Heuristics over Execution Time Jitter & Load
- Each numbered operating region is stable
- Transitions between operating regions: changes in SRT load, HRT+SRT jitter
- Performed using realistic hardware, OS, middleware, OFP application

Region 7 Performance (MUF does Better)

Region 8 Performance (RMS+MLF does Better)

Adaptation over Scheduling Heuristics

A Map of the Best Performing Heuristic over Execution Jitter & Load
- RMS performs best if system is under-loaded (theory predicts this)
- RMS+MLF performs best in overload if jitter is very high or very low
- MUF performs best if jitter is moderate

A Basis for Adaptive Control
- Run-time observable measure correlates with performance: operation latency
- A simple automation could be constructed

My Contribution: A Unified Middleware Approach

Technical Challenge Research Approach Research Impact
Efficient and Safe Systems
Customizable Middleware
Nimble Adaptation
Performance Awareness
Inclusive Systems Approach
Real QoS problems require both theoretical and empirical perspectives
- Scheduling theory generalized over OS/middleware primitives → heuristic space
- Empirical study of specific heuristic (sub-)spaces is crucial
- Analogy: theory/studies of Ethernet behavior: bin-exp-backoff vs. congestion collapse

Research Impacts and Collaborations

Topics Publications, Systems, & Middleware
Kokyu
- IEEE Proceedings special issue (submission in progress)
- Distinct open-source framework (Kokyu) – early 2002
Integrated Middleware
- With Boeing, Honeywell: DASC, 1999
- With Boeing, BBN: ICDCS, 2001
- With Boeing: DASC, 2001
- WORDS 2002
- Boeing: ASTD, WSOA, Bold Stroke
Resource Management
- With Boeing, ASFD, WSOA, Bold Stroke (SEC, MoBIES)
Real-Time Metrics & Visualization
- DASC, 1999
- DASC, 2000
- Boeing: ASFD, WSOA, ASTD II, Bold Stroke (in progress)
Infrastructure
- Integrated Middleware Resource Management
- IEEE Proceedings special issue (submission in progress)
- Boeing: ASTD, WSOA, Bold Stroke
## Future Research Objectives

<table>
<thead>
<tr>
<th>Topics</th>
<th>Problems, Approaches, &amp; Investigations</th>
</tr>
</thead>
</table>
| **Extremely Small Footprint DRE Firm/Soft Middleware** | - Dispatch/comm/addressing in micro-niches (downward scalability)  
- Interesting design tensions between time/space/power/...  
- "Just enough" middleware: e.g., from Jini-like backbone to an ORB  
- DARPA ITO NEST Program: OEP middleware |
| **Advanced Techniques for QoS Mechanism Instantiation** | - Composing middleware scheduling/dispatching points end-to-end  
- Heterogeneous: multiple layers and paths  
- Multi-dimensional resource management: memory, network, CPU  
- Discover/apply good heuristics and domain-specific optimizations  
- Experimental/theoretical study/construction of adaptive decision lattices  
- AOP, domain-specific type systems |
| **Coordinated Multi-Layer Multi-Agent QoS Management** | - Integration, cooperation and co-design  
- Resource managers, schedulers, dispatchers, feature sets  
- Across hardware, firmware, OS, middleware, application layers  
- Toward generalized techniques, patterns, and a "complete" theory and practice of QoS composition for real-world systems  
- E.g., real-time + anytime + adaptive control + decision-aiding + power-awareness + footprint-awareness +... |

## Concluding Remarks

**Empirical Evaluation**
- Validates adaptive/hybrid scheduling approach  
- Quantifies cost/benefits of discrete alternatives  
- Powerful when combined with theoretical view  
- "Mining" technique for problems & properties

**Composable Scheduling/Dispatching**
- Enables domain-specific optimizations, especially when design decisions are aided by empirical data

**Heuristic Space Experiments**
- Offer a quantitative blueprint for adaptation

**Open-Source Code**
- All software described here that is uniquely a part of my research will be made available in the ACE_wrappers distribution  
- Kokyu framework (early 2002)  
- Dispatching for new TAO Event Channel

## Thanks

**Mentors**
- Dr. Douglas C. Schmidt, Dr. David L. Levine, and Dr. Ron K. Cytron

**Colleagues and Collaborators**
- Faculty and Staff of WU CS and CoE  
- Dr. Douglas Niehaus  
- Mr. David Sharp, Mr. Bryan Doerr, Mr. Don Winter, Dr. David Corman, Dr. Doug Stuart, Mr. Brian Mendel, Mr. Greg Holtmeyer, Mr. Pat Goertzen, Ms. Jeanina Gossett, Ms. Amy Wright, Mr. Jim Unness, Mr. Tom Venturella, Mr. Russ Wolter  
- Mr. Kenneth Littlejohn (AFRL), Dr. Gary Koob (DARPA ITO)  
- Dr. Rakesh Jha, Mr. John Shackleton, Mr. Nigel Birch  
- Dr. Joseph Loyall, Dr. Richard Schantz, Dr. John Zinky, Dr. David Bakken  
- Dr. Ebrahim Moshiri, Mr. Malcolm Spence, Mr. Kevin Stanley

**Friends and Family**
- Members of the DOC Group  
- WU CS Graduate Students  
- My wife Barb and son Paul  
- My Mom Dr. Helen Gill, Dad Mr. David Gill and sister Ms. Sarah E. Gill  
- My parents and siblings-in-law

## Additional Questions ?

These slides:  
http://www.cs.wustl.edu/~cdgill/cdgill_defense.ppt  

Contact Information:  
http://www.cs.wustl.edu/~cdgill  
cdgill@cs.wustl.edu