Douglas C. Schmidt schmidt@cs.wustl.edu

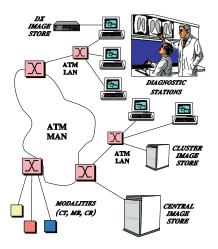
Washington University, St. Louis www.cs.wustl.edu/~schmidt/

This work was sponsored in part by Boeing and DARPA

Douglas C. Schmidt

High-performance CORBA Tutorial

Problem: High-Performance, Real-time Middleware



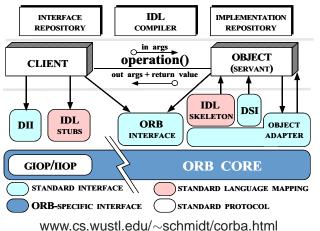
- · Many applications require high-performance
 - e.g., telecom, imaging,
- Building these applications manually is hard
- · Existing middleware doesn't support performance effectively
- e.g., CORBA, DCOM, DCE

Washington University, St. Louis

Douglas C. Schmidt

High-performance CORBA Tutorial

Candidate Solution: CORBA



Goals of CORBA

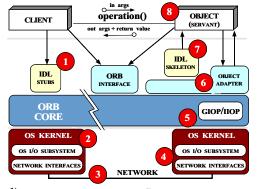
- Simplify distribution by automating
 - * Object location & activation
 - * Parameter marshaling
 - * Demultiplexing
 - * Error handling
- Provide foundation for higher-level services

2

Douglas C. Schmidt

High-performance CORBA Tutorial

Performance Challenges for ORB Middleware



- 1) CLIENT MARSHALING 2) CLIENT PROTOCOL OUEUEING
- 3) NETWORK DELAY
- 4) SERVER PROTOCOL QUEUEING 8) METHOD EXECUTION
- 5) THREAD DISPATCHING 6) REQUEST DISPATCHING 7) SERVER DEMARSHALING

- Key Challenges
 - Specifying QoS requirements
 - Determining operation schedules
 - Alleviating priority inversion and non-determinism
 - Reducing latency/jitter for demultiplexing
 - Reducing presentation laver overhead
 - Maintaining small footprint

CLIENT

RIDL

STUBS

REAL-TIME

ORB CORE

OS KERNEL

REAL-TIME I/O

SUBSYSTEM

HIGH-SPEED

TWORK INTERFACE

high-performance

* Runs on POSIX.

Win32, RTOSs

Leverages ACE

TAO Overview

A real-time,

ORB

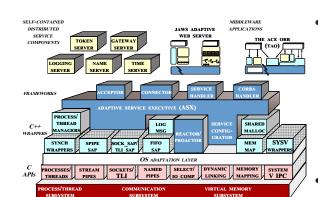
Related work

U. RI. Mitre

Mich.

QuO at BBN

ARMADA at U.



GENERAL POSIX AND Win32 SERVICES www.cs.wustl.edu/~schmidt/ACE.html

ACE Overview

- Concurrent OO networking framework
- Ported to C++ and Java
- Runs on RTOSs. POSIX, and Win32

Related work

- x-Kernel
- SysV STREAMS

Washington University, St. Louis

Washington University, St. Louis

5

Douglas C. Schmidt

High-performance CORBA Tutorial

Scope: Performance Optimizations in TAO

The ACE ORB (TAO)

RIDL

OBJECT

(SERVANT)

REAL-TIME

OBJECT

ADAPTER

RIOP

OS KERNEL

REAL-TIME I/O

SUBSYSTEM

HIGH-SPEED

NETWORK INTERFACE

in args

operation()

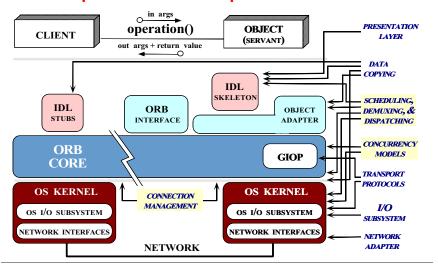
out args + return value

ORB RUN-TIME

SCHEDULER

NETWORK

www.cs.wustl.edu/~schmidt/TAO.html

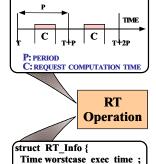


Douglas C. Schmidt

Douglas C. Schmidt

High-performance CORBA Tutorial

Problem: Providing QoS to CORBA Operations



Period period ;

Criticality criticality;

Importance importance;

PERIODIC SCHEDULING/DISPATCHING

Design Challenges

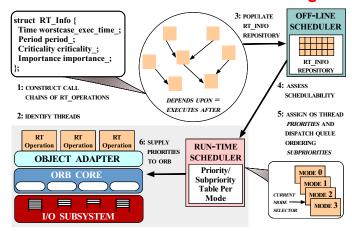
- Specifying/enforcing QoS requirements
- Focus on *Operations* upon *Objects*
 - * Rather than on communication channels or threads/synchronization
- Support static and dynamic scheduling

Solution Approach

- Servants publish resource (e.g., CPU) requirements and (periodic) deadlines
- Most clients are also servants

Washington University, St. Louis Washington University, St. Louis

Solution: TAO's Real-time Static Scheduling Service

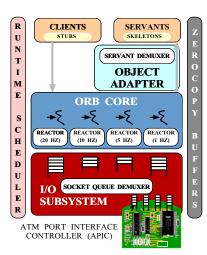


www.cs.wustl.edu/~schmidt/TAO.ps.gz

Washington University, St. Louis

8

TAO's High-Performance, Real-time ORB Endsystem



Solution Approach

- Integrate RT dispatcher into ORB endsystem
- Support multiple request scheduling strategies
 - * e.g., RMS, EDF, and MUF
- Requests ordered across thread priorities by OS dispatcher
- Requests ordered within priorities based on data dependencies and importance

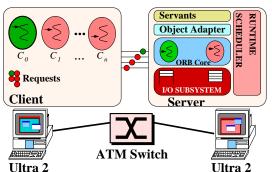
Washington University, St. Louis

9

Douglas C. Schmidt

High-performance CORBA Tutorial

ORB Latency and Priority Inversion Experiments



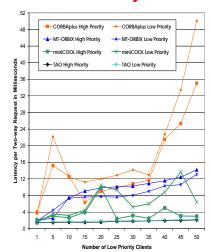
www.cs.wustl.edu/~schmidt/RT-perf.ps.gz

- Vary ORBs, hold OS constant
- Solaris real-time threads
- $\bullet \ \ \mbox{High priority client} \ C_0 \\ \mbox{connects to servant} \ S_0 \\ \mbox{with matching priorities}$
- Clients C₁...C_n have same lower priority
- Clients $C_1 \dots C_n$ connect to servant S_1
- Clients invoke twoway CORBA calls that cube a number on the servant and returns result

Douglas C. Schmidt

High-performance CORBA Tutorial

ORB Latency and Priority Inversion Results



• Synopsis of results

- TAO's latency is lowest for large # of clients
- TAO avoids priority inversion
 - * *i.e.*, high priority client always has lowest latency
- Primary overhead stems from concurrency and connection architecture
 - * *e.g.*, synchronization and context switching

Washington University, St. Louis

10

in milliseconds

SERVANT

SKELETONS

4: dispatch

upcall()

OBJECT

ADAPTER

SERVANT DEMUXER

→ SERVANT

2: enqueue(data) ORB CORE

I/O SUBSYSTEM

Common Problems

- High overhead
 - * Context switching
 - * Synchronization
- Thread-level priority inversions
 - * FIFO request queueing
 - * Improper thread priorities
- Lack of application control over concurrency model

Problem: Improper ORB Concurrency Model

 Definition Standard deviation from average latency Synopsis of results - TAO's jitter is lowest and most consistent - CORBAplus' jitter is highest and

most variable

Washington University, St. Louis

12

Washington University, St. Louis

Douglas C. Schmidt

3: dequeue.

filter

&enqueue

request, FILTER

THREAD

FILTER

FILTER

FILTER

13

Douglas C. Schmidt

High-performance CORBA Tutorial

CORBAplus High Priority

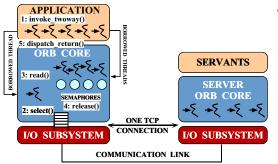
MT-ORBIX Low Priority MT-ORBIX High Priority

miniCOOL Low Priority miniCOOL High Priority

TAO Low Priority

Problem: ORB Shared Connection Model

ORB Jitter Results



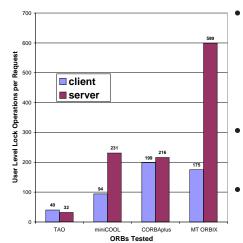
Common Problems

- Request-level priority inversions
 - * Sharing multiple priorities on a single connection
- Complex connection multiplexing
- Synchronization overhead

Douglas C. Schmidt

High-performance CORBA Tutorial

Problem: High Locking Overhead

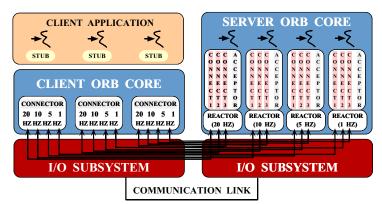


- Locking overhead significantly affects latency and jitter
 - Memory management commonly involves locking
- RT ORBs should minimize or eliminate all locking operations
- TAO is carefully designed to minimize locking and memory allocation

Washington University, St. Louis

14

Solution: TAO's Inter-ORB Connection Topology

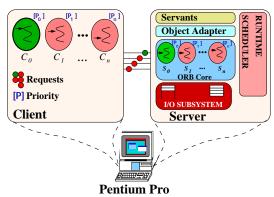


www.cs.wustl.edu/~schmidt/RT-middleware.ps.gz

Washington University, St. Louis

16

Real-time OS/ORB Performance Experiments



www.cs.wustl.edu/ \sim schmidt/RT-OS.ps.gz

- · Vary OS, hold ORBs constant
- Single-processor Intel Pentium Pro 200 Mhz, 128 Mbytes of RAM
- Client and servant run on the same machine
- Client C_i connects to servant S_i with priority P_i
 - i ranges from $1 \dots 50$
- Clients invoke twoway CORBA calls that cube a number on the servant and returns result

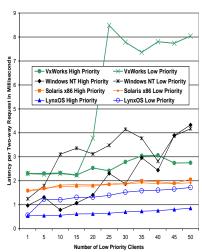
Washington University, St. Louis

17

Douglas C. Schmidt

High-performance CORBA Tutorial

Real-time OS/ORB Performance Results



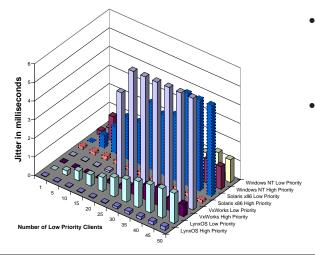
• Synopsis of results

- LynxOS yielded very good latency and deterministic behavior
- Erratic behavior and high latency are a problem for Windows NT
- Windows NT also showed priority inversion at 50 low priority clients
- VxWorks performs surprisingly erratically
- Solaris' latency is high but predictable

Douglas C. Schmidt

High-performance CORBA Tutorial

Real-time OS/ORB Jitter Results



Definition

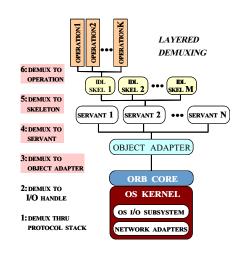
- Standard deviation from average latency
- Synopsis of results
 - Some RTOS's provide low jitter
 - ORB (TAO) doesn't introduce jitter

(D) DE-LAYERED ACTIVE DEMUXING

index(object key/operation)

OBJECT ADAPTER

Problem: Reducing Demultiplexing Latency



Design Challenges

- Minimize demuxing layers
- Provide O(1) operation demuxing
- Avoid priority inversions
- RemainCORBA-compliant

Washington University, St. Louis

20

• Results at www.cs.wustl.edu/~schmidt/ieee_tc-97.ps.gz

SERVANT 1

Solution: Demultiplexing Optimizations

IDL SKEL 1 SKEL 2

(B) LAYERED DEMUXING,

DYNAMIC HASHING

(C) LAYERED DEMUXING,

SERVANT 2 ••• SERVANT 500

hash(object key)

OBJECT ADAPTER

• Linear search based on Orbix demuxing strategy

• Perfect hashing based on GNU gperf

(A) LAYERED DEMUXING,

LINEAR SEARCH

SERVANT 1 SERVANT 2 ••• SERVANT 500

OBJECT ADAPTER

www.cs.wustl.edu/~schmidt/gperf.ps.gz

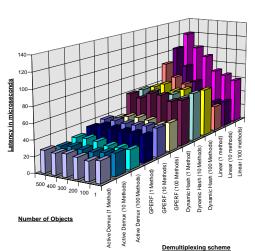
Washington University, St. Louis

21

Douglas C. Schmidt

High-performance CORBA Tutorial

Demultiplexing Performance Results



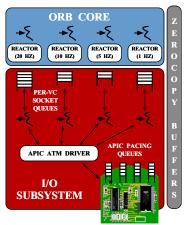
• Synopsis

- Linear search is far too costly
- Dynamic hashing can be unstable
- gperf solution is 100% compatible, but static
- Optimal active demuxing may not be 100% compatible, but is dynamic
- Strategy pattern facilitates flexibility

Douglas C. Schmidt

High-performance CORBA Tutorial

Next Steps: Integrating TAO with ATM I/O Subsystem



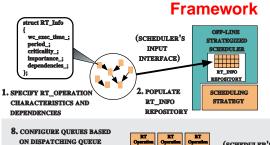
Key Features

- Vertical integration of QoS through ORB, OS, and ATM network
- Real-time I/O enhancements to Solaris kernel
- Provides rate-based QoS end-to-end
- Leverages APIC features for cell pacing and zero-copy buffering

Washington University, St. Louis

22

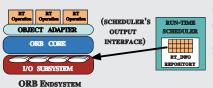
Next Steps: Strategized Scheduling Service



OFF-LINE

- 3. ASSESS SCHEDULABILITY
- 4. ASSIGN STATIC PRIORITY AND STATIC SUBPRIORITY
- 5. MAP STATIC PRIORITY, DYNAMIC SUBPRIORITY, AND STATIC SUBPRIORITY INTO DISPATCHING PRIORITY AND DISPATCHING SUBPRIORITY
- 6. ASSIGN DISPATCHING OUEUE CONFIGURATION

- 8. CONFIGURE QUEUES BASEL
 ON DISPATCHING QUEUE
 CONFIGURATION
 10. DUNANTO OUTSIDES ASSIGN.
- 10. DYNAMIC QUEUES ASSIGN DYNAMIC PORTIONS OF DISPATCHING SUBPRIORITY (AND POSSIBLY DISPATCHING PRIORITY)



ON-LINE 7. SUPPLY DISPATCHING QUEUE CONFIGURATION TO THE ORE

9. SUPPLY STATIC PORTIONS OF DISPATCHING PRIORITY AND DISPATCHING SUBPRIORITY TO THE ORB

www.cs.wustl.edu/ \sim schmidt/dynamic.ps.gz

Washington University, St. Louis

24

Principles for High-Performance, Real-time ORBs

- Avoid dynamic connection management
- Minimize dynamic memory management and data copying
- Avoid multiplexing connections for different priority threads
- Avoid complex concurrency models
- Integrate ORB with OS and I/O subsystem and avoid reimplementing OS mechanisms
- Guide ORB design by empirical benchmarks





Washington University, St. Louis

25

Douglas C. Schmidt

High-performance CORBA Tutorial

TAO Project Research Summary

- Current focus: real-time ORBs
 - Developed first deployed real-time CORBA scheduling service and first POA
 - Minimized ORB Core priority inversion and non-determinism
 - Reduced end-to-end latency via demuxing optimizations
 - Applied optimizations to IIOP protocol engine
 - Co-submitters to OMG's real-time CORBA RFP
- Future work
 - Dynamic and hybrid scheduling of CORBA operations
 - Distributed QoS and integration with real-time ATM I/O Subsystem
 - Optimizing IDL compiler
 - Technology transfer with DARPA Quorum program

Douglas C. Schmidt

High-performance CORBA Tutorial

Web URLs for Additional Information

These slides:

www.cs.wustl.edu/~schmidt/PDF/tutorial4.ps.gz

- More information on TAO: www.cs.wustl.edu/~schmidt/RT-ORB.ps.gz
- TAO Event Channel: www.cs.wustl.edu/~schmidt/JSAC-98.ps.gz
- TAO static scheduling: www.cs.wustl.edu/~schmidt/TAO.ps.gz
- TAO dynamic scheduling:

www.cs.wustl.edu/~schmidt/dynamic.ps.gz

• ORB Endsystem Architecture:

www.cs.wustl.edu/~schmidt/RT-middleware.ps.gz

Web URLs for Additional Information (cont'd)

- Performance Measurements:
 - Demuxing latency: www.cs.wustl.edu/~schmidt/GLOBECOM-97.ps.gz
 - SII throughput: www.cs.wustl.edu/~schmidt/SIGCOMM-96.ps.gz
 - DII throughput: www.cs.wustl.edu/~schmidt/GLOBECOM-96.ps.gz
 - Latency, scalability: www.cs.wustl.edu/~schmidt/ICDCS-97.ps.gz
 - IIOP optimizations: www.cs.wustl.edu/~schmidt/JSAC-99.ps.gz
- More detail on CORBA: www.cs.wustl.edu/~schmidt/corba.html
- ADAPTIVE Communication Environment (ACE):

www.cs.wustl.edu/~schmidt/ACE.html

• The ACE ORB (TAO):

www.cs.wustl.edu/~schmidt/TAO.html