An Empirical Analysis of CORBA and OS Support for Real-time Middleware

Sergio Flores-Gaitan
sergio@cs.wustl.edu

Advisor: Douglas C. Schmidt <schmidt@cs.wustl.edu>

Washington University, St. Louis
http://www.cs.wustl.edu/∼sergio/RTOS-benchmark.ps.gz

Sponsors
Lucent, Motorola, and Sprint

Outline of the talk

- Motivation
- Benchmarking Environment
- Key Results
- Concluding Remarks

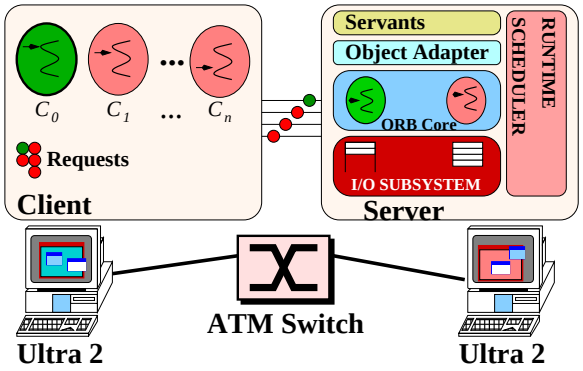
Previous Work

- Real-time results where we varied the ORBs tested
  `www.cs.wustl.edu/∼schmidt/RT-perf.ps.gz`
  - ORBs on Sun Solaris on an UltraSPARC
    - **TAO** for C++ version 0.0.42 (www.cs.wustl.edu/∼schmidt/TAO.html)
    - Chorus **miniCOOL** for C++ version r4.3 (www.chorus.com)
    - IONA **MT-Orbix** version 2.2 (www.iona.com)
    - PowerBroker **CORBAplus** for C++ version 2.1.1 (www.expersoft.com)
  - ORBs on Sun ClassiX r3.1 RTOS on an MVME177
    - **TAO** for C++ version 0.0.42 (www.cs.wustl.edu/∼schmidt/TAO.html)
    - Chorus **miniCOOL** for C++ version r4.3 (www.chorus.com)

Motivation

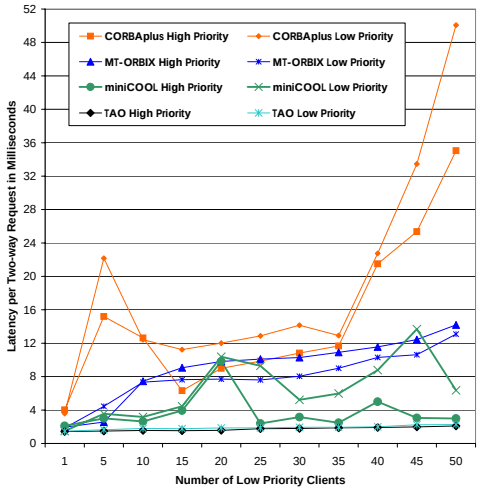
- Increasing demand to extend CORBA to support real-time distributed applications
- Conventional ORBs not yet suited for real-time application development
- General purpose operating systems do not have support for applications with QoS requirements

Black-box Test Configuration on Solaris



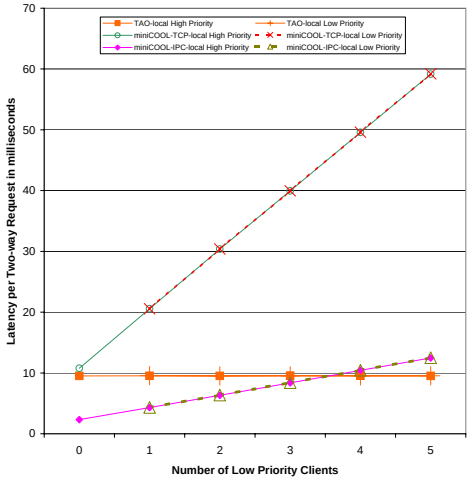
- Vary ORBs, hold operating system constant
- Solaris real-time threads
- High priority client C_0 connects to servant S_0 with matching priorities
- Clients $C_1 \ldots C_n$ have same lower priority
- Clients $C_1 \ldots C_n$ connect to servant S_1
- Clients invoke twoway CORBA calls that cube a number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

Black-box Real-time Results on Solaris



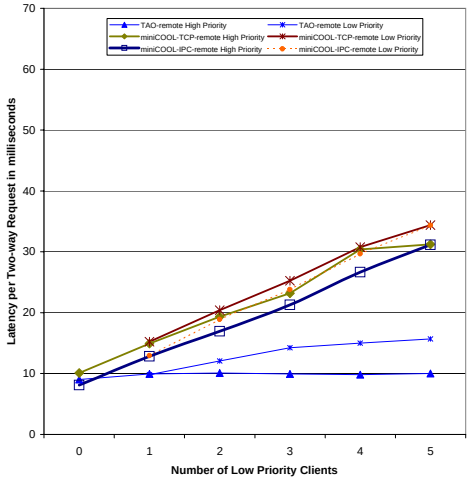
- Excessive use of user-level locks in **CORBAplus** makes its performance worst of all ORBs tested
- As the number of low priority clients increase, the latency for **miniCOOL** increases
- Latency in high priority clients increase in a non-linear fashion
- Non-linear behavior of **miniCOOL**, makes it unsuitable for real-time applications

Black-box Real-time Results on ClassiX (local)



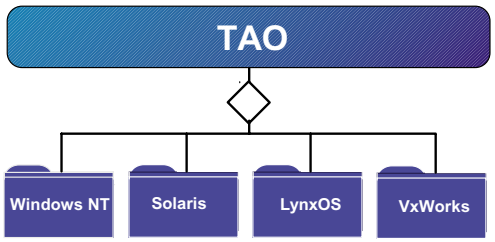
- As the number of low priority clients increase, the latency for **COOL** increases
- Latency in TAOs high priority client remains relatively stable as compared to COOL which increases in a linear fashion
- Unpredictable latency behavior of **COOL**, makes it unsuitable for real-time applications

Black-box Real-time Results on ClassiX (remote)



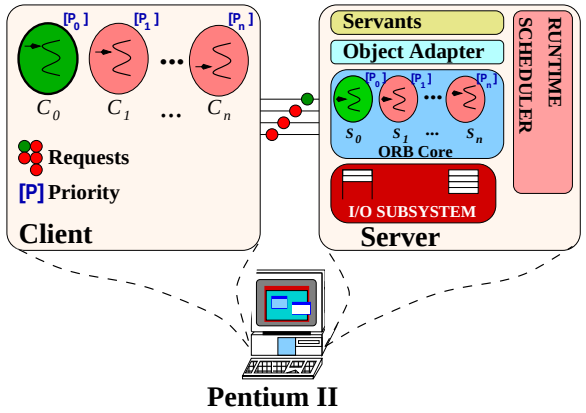
- As the number of low priority clients increase, the latency for **COOL** increases
- Latency in TAOs high priority client remains relatively stable as compared to COOL which increases in a linear fashion
- Unpredictable latency behavior of **COOL**, makes it unsuitable for real-time applications

Black-box Test Configuration for TAO



- ORB remains constant: **TAO**
  - Displayed most deterministic behavior compared with other ORBs
- General Purpose Operating System Tested
  - Sun **Solaris** 2.6
  - Microsoft **Windows NT** 4.0 SP3
- Real-time Operating System Tested
  - Lynx Real-Time Systems **LynxOS** 2.5.0
  - WindRiver **VxWorks** 5.3.1

Black-box Test Configuration for TAO (cont)

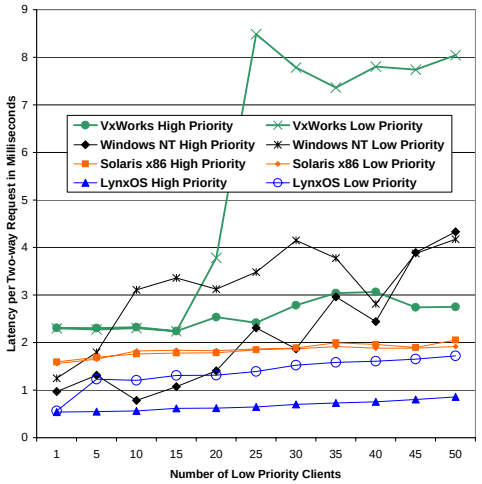


- Single-processor Intel Pentium Pro 200 Mhz
- 128 Mbytes of RAM
- Client and servant run on the same machine
- Client C_i connects to servant S_i with priority P_i
- i ranges from $1 \ldots 50$
- Clients invoke twoway CORBA calls that cube a number on the servant and returns result

Benchmark Environment

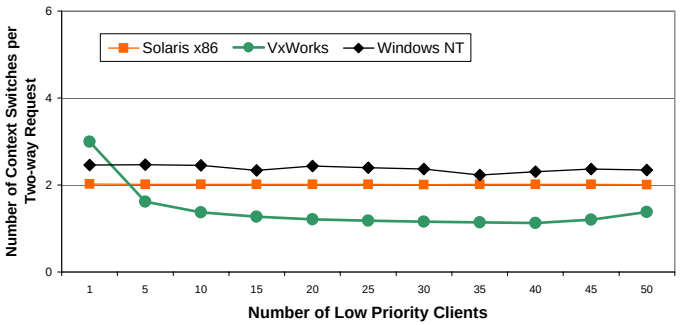
- Operating System Benchmarks
  - Number of Context Switches
  - Context Switch overhead
- ORB Benchmarks
  - CPU processing overhead
  - Priority Inversion
  - Latency

Black-box Real-time Latency Results for TAO



- 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 erratic
- Solaris' latency is high but predictable

Black-box Context Switch Results for TAO



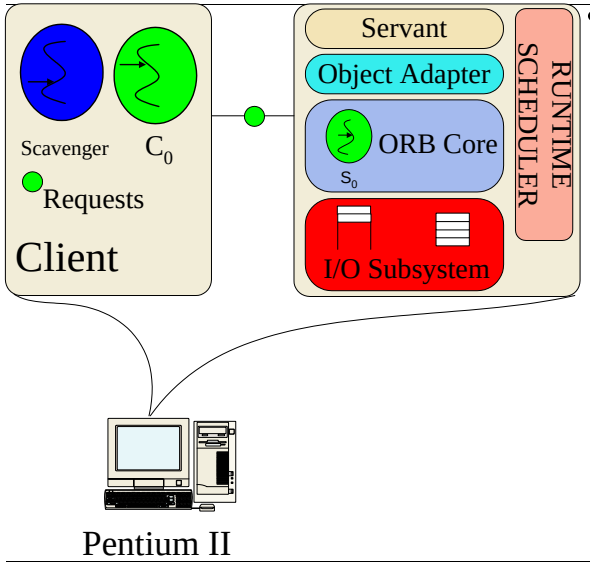
- Solaris showed low variability
- No results for LynxOS due to lack of instrumentation
- Windows NT incurs highest context switches per request
- VxWorks rapidly stabilizes to one context switch per request

Black-box Context Switch Time Results for TAO

- Solaris showed low variability
- No results for LynxOS due to lack of tools
- Windows NT incurs highest context switches per request
- VxWorks rapidly stabilizes to one context switch per request

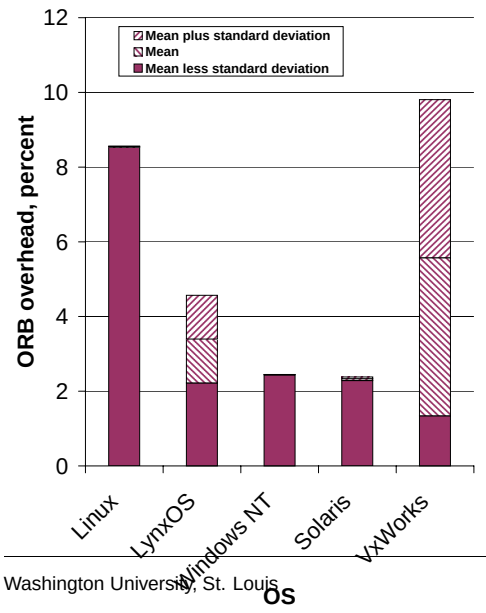
OS	Context Switch Time, usec mean (std dev)		
	Suspend-Resume Test	Yield Test	Synch Test
Solaris	46 (N/A)	17.8 (0.837)	128 (N/A)
VxWorks	15.4 (0.294)	40 (N/A)	N/A
LynxOS	N/A	3.72 (0.043)	5.45 (0.077)
Linux	N/A	6.52 (0.117)	18.2 (0.264)
NT	3.45 (0.059)	5.76 (0.449)	7.02 (0.113)

Black-box Client CPU Processing Test Configuration



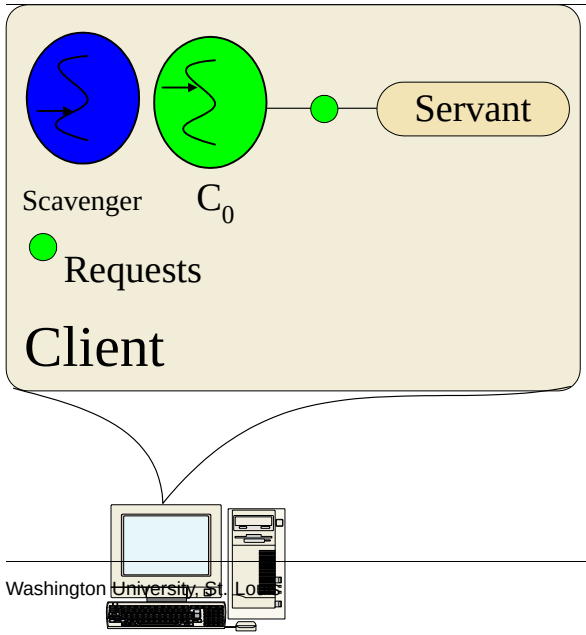
- Vary operating system, hold ORB constant
- Client runs on: Single-processor Intel Pentium Pro 200 Mhz
- Servant runs on: Single-processor UltraSPARC
- Client C_0 connects to servant S_0
- Clients invoke twoway CORBA calls that cube a number on the servant and returns result

Black-box Real-time Client CPU Processing Overhead Results for TAO



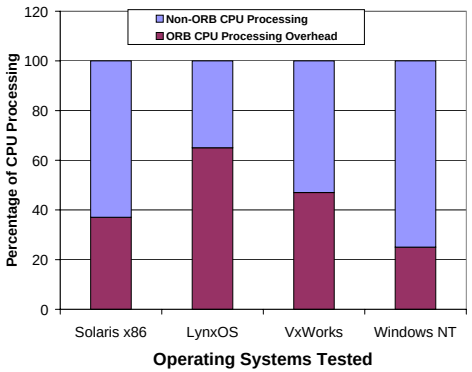
- VxWorks and LynxOS minimize their system call CPU processing overhead
- Solaris and Windows NT have high processing overhead, yielding unpredictable latency behavior

Black-box Server CPU Processing Test Configuration



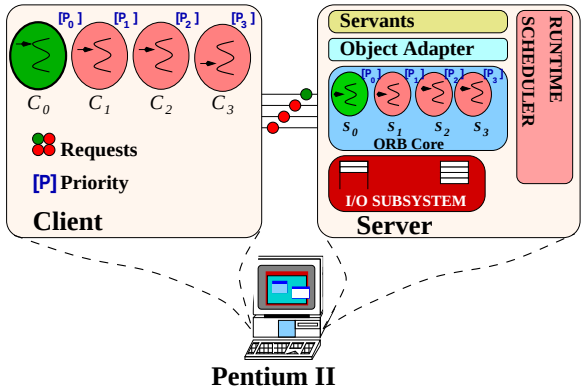
- Client runs on: Single-processor UltraSPARC
- Servant runs on: Single-processor Intel Pentium Pro 200 Mhz
- Client C_0 connects to servant S_0
- Clients invoke twoway CORBA calls that cube a number on the servant and returns result

Black-box Real-time Server CPU Processing Overhead Results for TAO



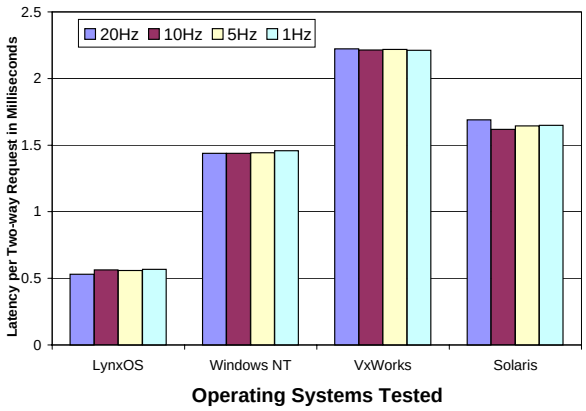
- VxWorks and LynxOS minimize their system call CPU processing overhead
- Solaris and Windows NT have high processing overhead, yielding unpredictable latency behavior

Black-box Priority Inversion Test Configuration



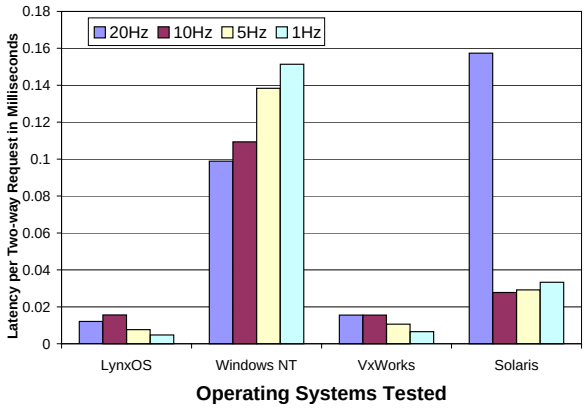
- Client and servant run on the same machine
- Client C_i connects to servant S_i with priority P_i
- i ranges from $1 \ldots 4$
- Clients invoke twoway CORBA calls that cube a number on the servant and returns result
- Clients invoke calls at 20, 10, 5 and 1 Hz

Black-box Priority Inversion Results for TAO



- VxWorks and Solaris show average priority inversion

Black-box Priority Inversion Jitter Results for TAO



- VxWorks and LynxOS have predictable latency
- Solaris and Windows NT have high jitter, yielding unpredictable latency behavior

Concluding Remarks

- Recommendations for operating systems that support applications with QoS requirements
  - Integrate I/O subsystem with CORBA middleware architecture
  - Provide deterministic context switch overhead
  - Optimize system calls and protocol processing
  - Provide QoS specification and enforcement
  - Provide better tool support to precisely determine sources of overhead
  - Adopt standard measurement-driven methodologies
  - Provide deterministic latency behavior
- Benchmarking is REALLY hard on platforms that do not have enough tool support, *e.g.*, VxWorks

Acknowledgments

- David Levine for developing the context switch time test
- Irfan Pyarali for his continuous support
- Sumedh Mungee and Dr. Gokhale for their comments on the paper
- Entire DOC group
- Dr. Schmidt