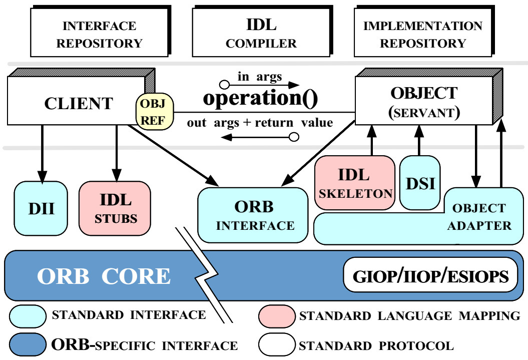
Design Principles and Optimizations for High-performance, Real-time CORBA

Aniruddha S. Gokhale
gokhale@cs.wustl.edu
http://www.cs.wustl.edu/∼gokhale/

Alexander B. Arulanthu
alex@cs.wustl.edu
http://www.cs.wustl.edu/∼alex/

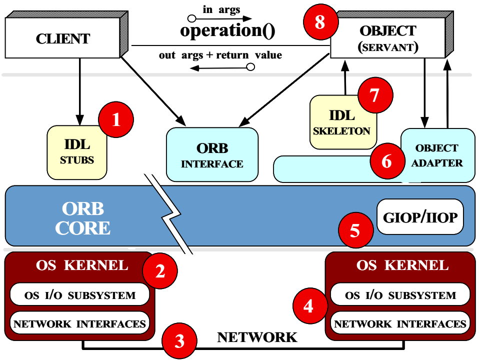
Washington University, St. Louis

Limitations of CORBA for Real-time Systems



- **Limitations**
  - Lack of QoS specifications
  - Lack of QoS enforcement
  - Lack of real-time programming features
  - Lack of performance optimizations

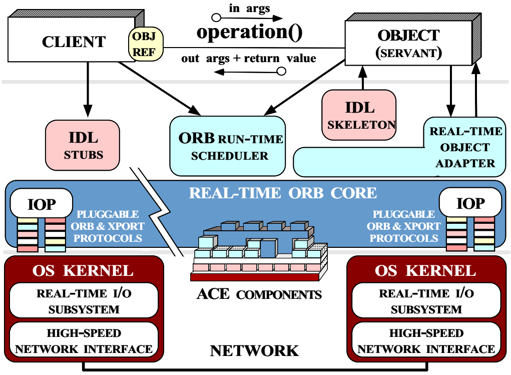
Research Problems: Meeting QoS Requirements



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

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

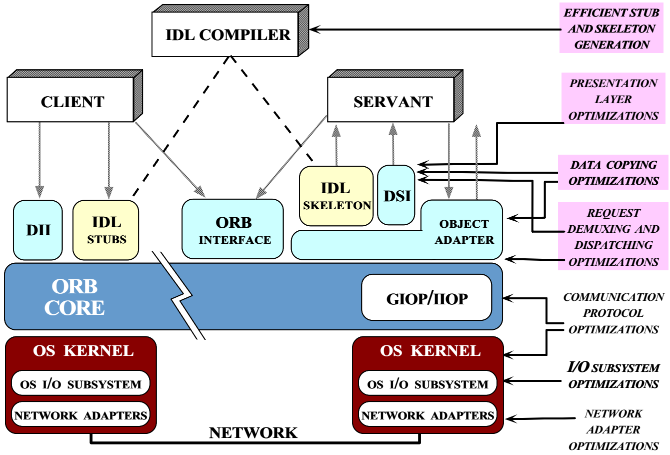
The ACE ORB (TAO)



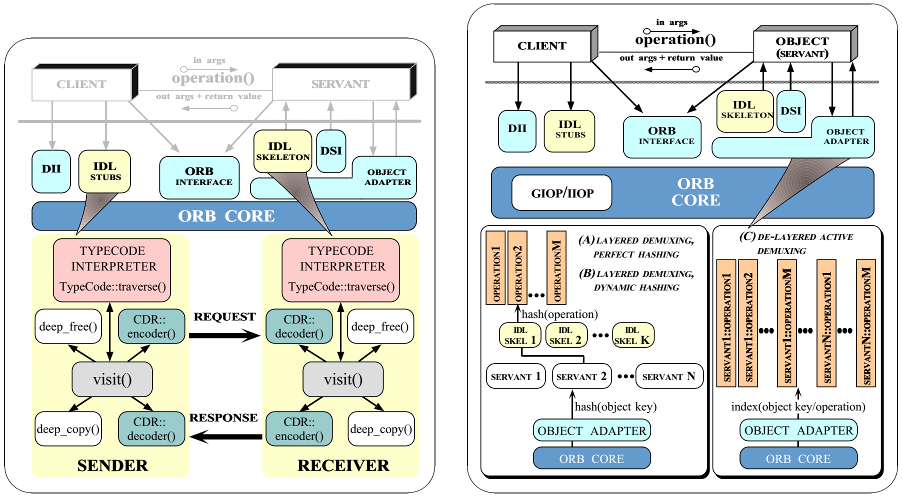
www.cs.wustl.edu/∼schmidt/TAO.html

- **TAO Overview**
  - A high-performance, real-time ORB
    - ∗ Telecom and avionics focus
  - Leverages the ACE framework
    - ∗ Runs on VxWorks, POSIX, and Win32
- **Related Work**
  - U. RI, MITRE
  - ARMADA (U. Mich.)
  - QuO (BBN)

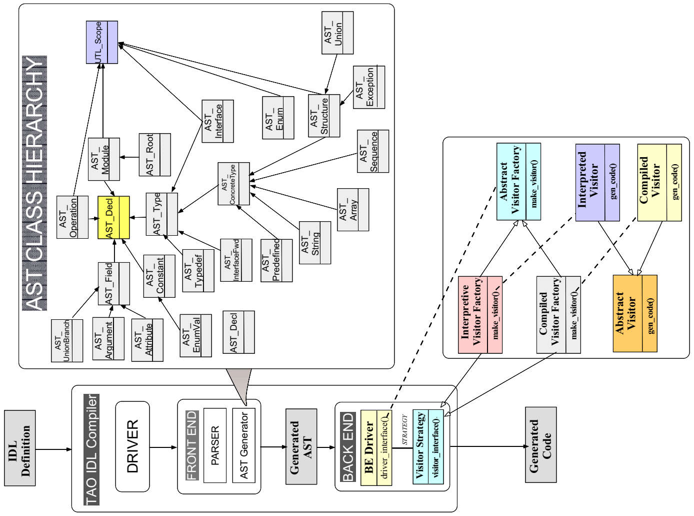
Research Contributions: TAO Optimizations


IDL COMPILER
CLIENT
SERVANT
DII
IDL STUBS
ORB INTERFACE
IDL SKELETON
DSI
OBJECT ADAPTER
ORB CORE
GIOP/IIOP
OS KERNEL
OS I/O SUBSYSTEM
NETWORK ADAPTERS
OS KERNEL
OS I/O SUBSYSTEM
NETWORK ADAPTERS
NETWORK
EFFICIENT STUB AND SKELETON GENERATION
PRESENTATION LAYER OPTIMIZATIONS
DATA COPYING OPTIMIZATIONS
REQUEST DEMUXING AND DISPATCHING OPTIMIZATIONS
COMMUNICATION PROTOCOL OPTIMIZATIONS
I/O SUBSYSTEM OPTIMIZATIONS
NETWORK ADAPTER OPTIMIZATIONS

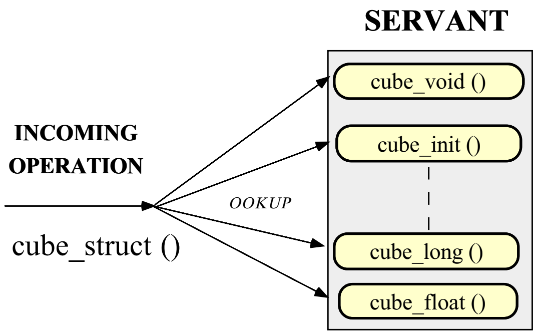
Research Contributions: IIOP and Demux Engines



TAO IDL Compiler Design

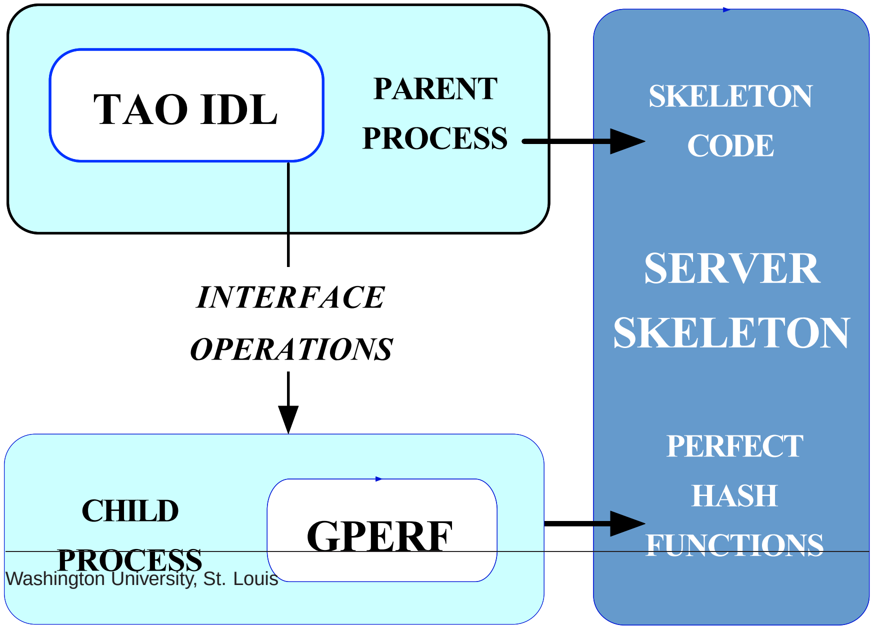


Optimizing Operation Dispatching

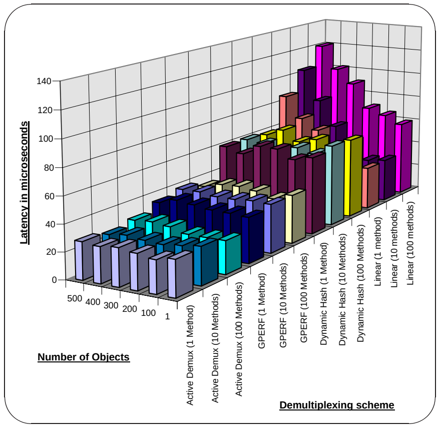


- Incoming CORBA requests dispatched to servant operation
- Goal is to minimize dispatching overhead

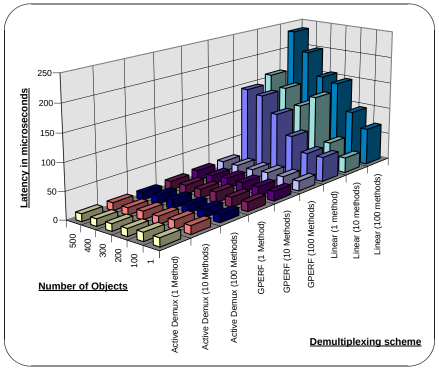
Using GPERF to Optimize Operation Dispatching



Demultiplexing Performance Results



Random



Worst case

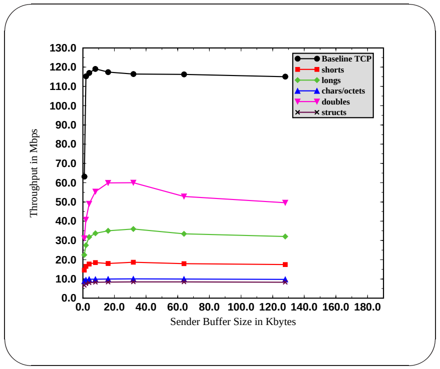
Optimization Principles

Number	Principle
1	Optimize for the common case
2	Eliminate gratuitous waste
3	Replace inefficient general-purpose methods with efficient special-purpose ones
4	Precompute values, when possible
5	Store redundant state to speed up expensive operations
6	Pass information between layers
7	Optimizations for cache

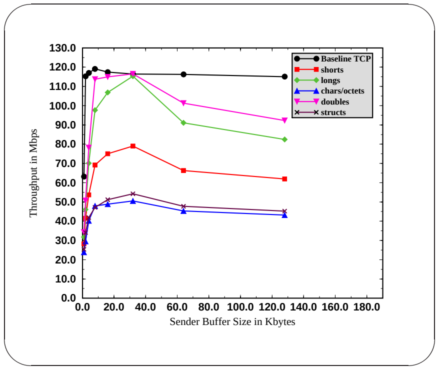
Related Work

- G. Varghese, SIGCOMM'96
- Clark:90, Abott:93 – ILP
- Peterson:96 – Outlining
- Clark:89 – Header prediction
- Peterson:94 (PathFinder), Engler:96 (DPF), Mahesh:95 (packet filters)

Throughput Comparisons



Original SunSoft



Optimized TAO

www.cs.wustl.edu/∼schmidt/HICSS-97.ps.gz (Best Paper Award)