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

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

Advisor: **Dr. Douglas C. Schmidt**

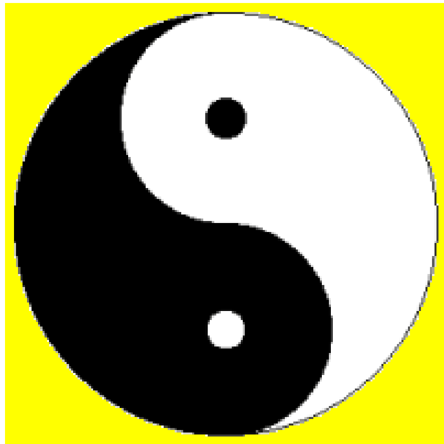
Washington University, St. Louis

May 13th, 1998

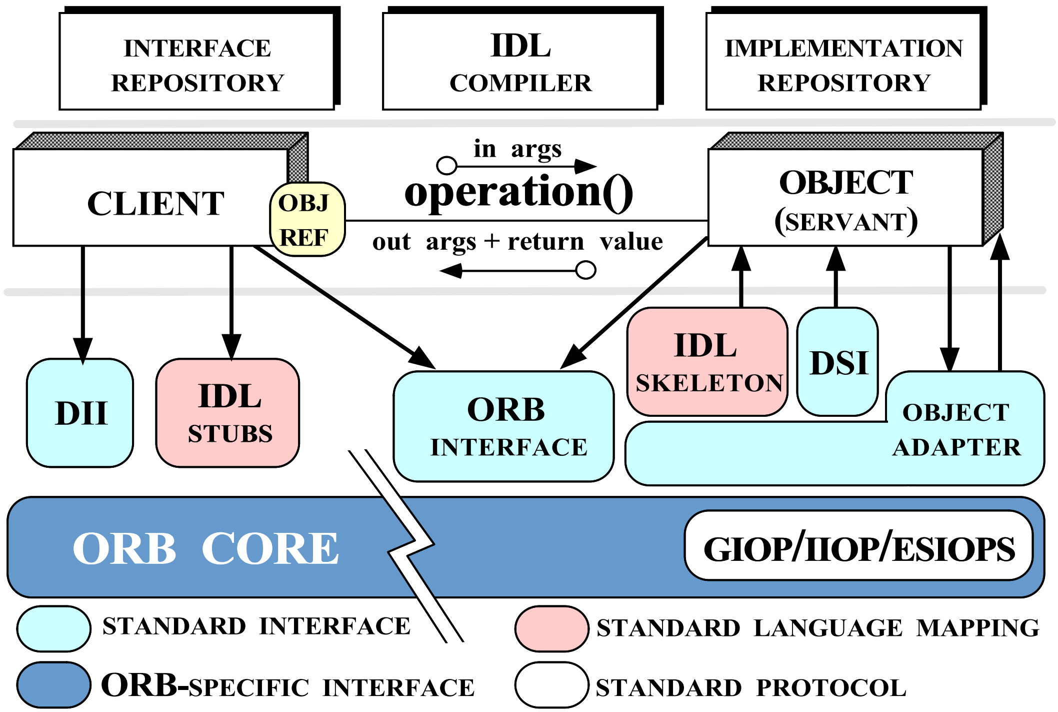
Doctoral Research (Funded by NSF grant NCR 9628218)

Presentation Outline

1. Motivation
   - Need for CORBA
   - Lack of Real-time support in CORBA
   - Research Contributions
2. Research Contributions – Optimizations in TAO ORB
   - Optimized IIOP Protocol engine
   - TAO IDL Compiler optimizations
   - Efficient Demultiplexing
3. Concluding Remarks
4. Future Work



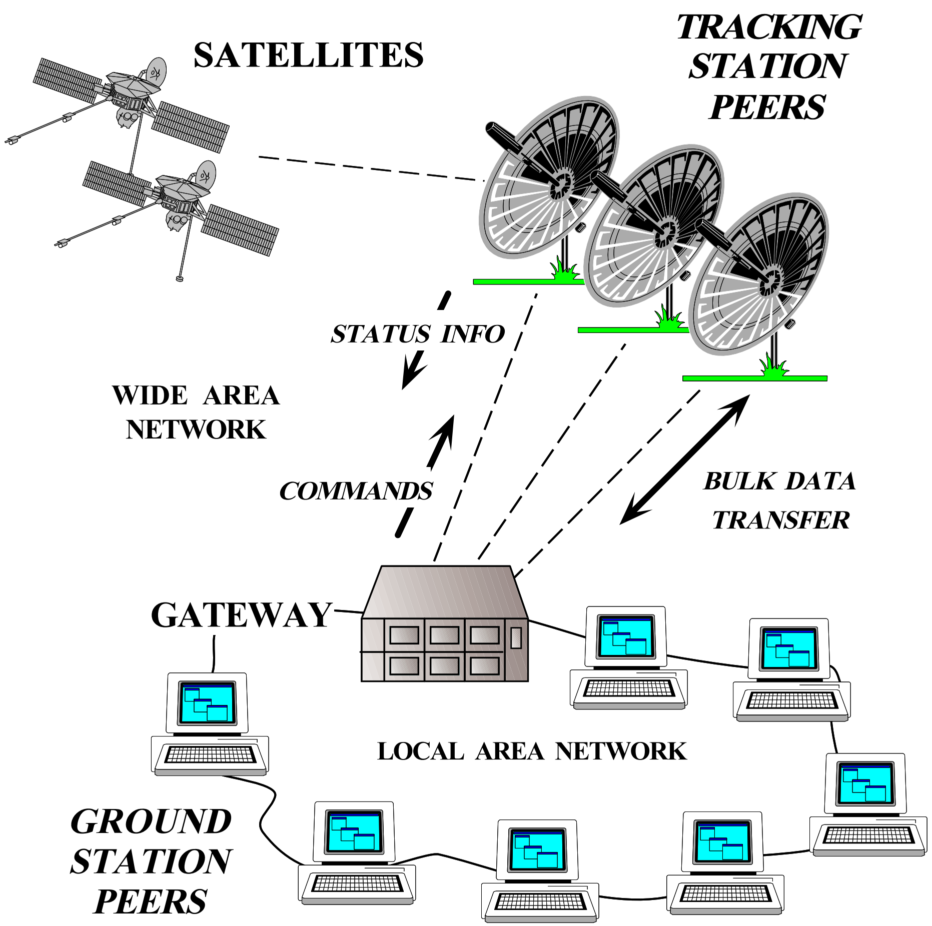
CORBA: Solution to the Distributed Software Crisis ?



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

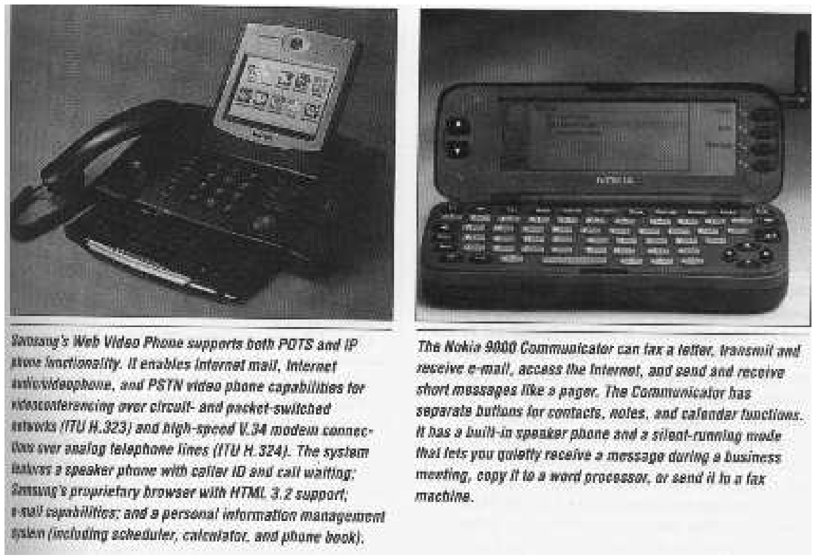
- **Goals of CORBA**
  - Simplify distribution by automating
    - Object location and activation
    - Parameter marshaling
    - Demultiplexing
    - Error handling
  - Provide foundation for higher-level services

Need for Real-time Features in CORBA



- Many applications require QoS guarantees
  - *e.g.*, telecom, avionics, WWW
- Building these applications manually is hard
- Existing middleware doesn't support QoS effectively
  - *e.g.*, CORBA, DCOM, DCE
- Solutions must be *integrated*

Rapid Growth in Hand-held Devices

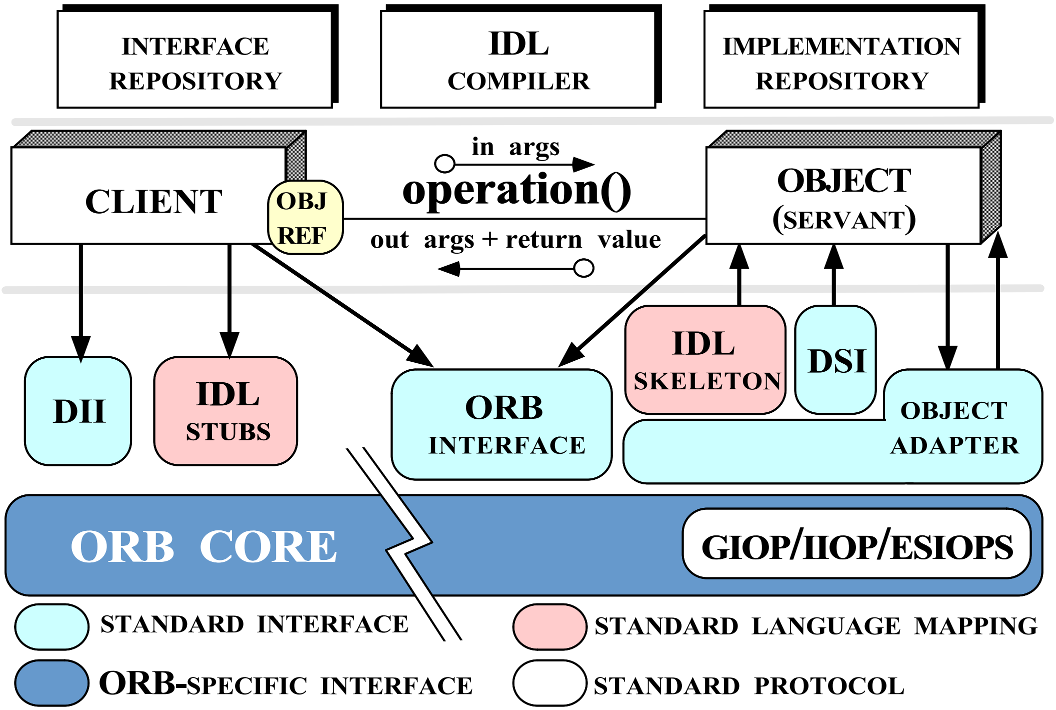


Lewis:98 (IEEE Computer Jan'98)

Comerford:98 (IEEE Spectrum May'98)

- **Devices:** WebPhones, WebTVs, PIMs, Palm PCs
- **RTOS:** Inferno, Windows CE2.0, PalmOS, VXWorks
- **Constraints:** Low power consumption => less storage => small footprint, but good and predictable performance

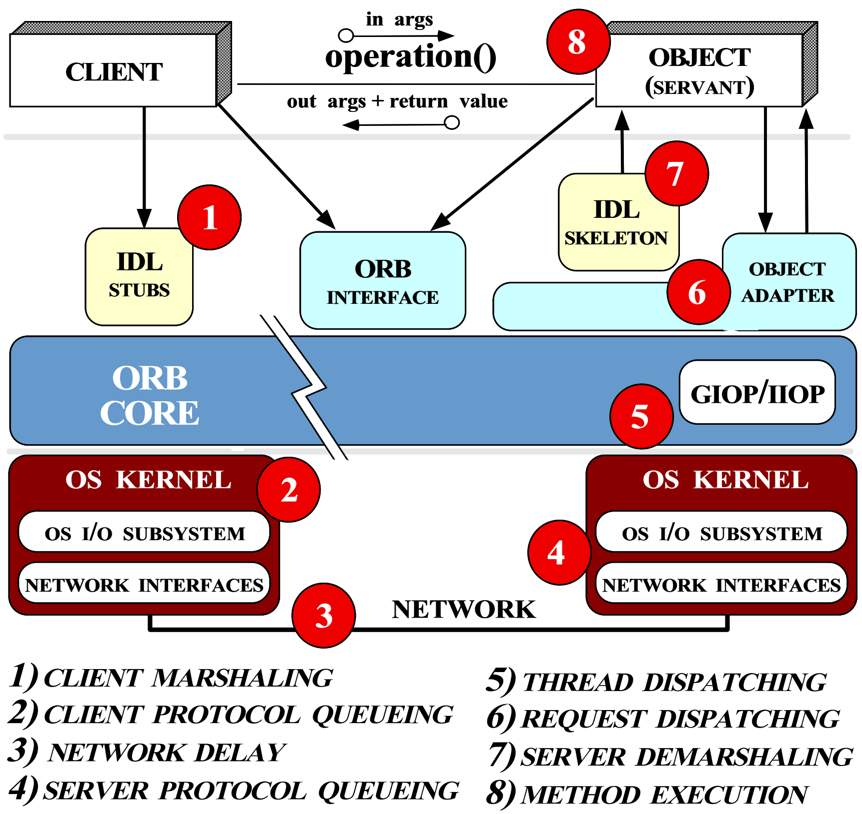
Caveat: Limitations of CORBA for Real-time Systems



Related Work: *Gokhale, Schmidt, Levine, Mungee, Flores, Parulkar, Harrison* – SIGCOMM'96, GLOBECOM'96, ICDCS'97, IEEE Communications Feb'97, IEEE RTAS'98, IEEE RTSS'97

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

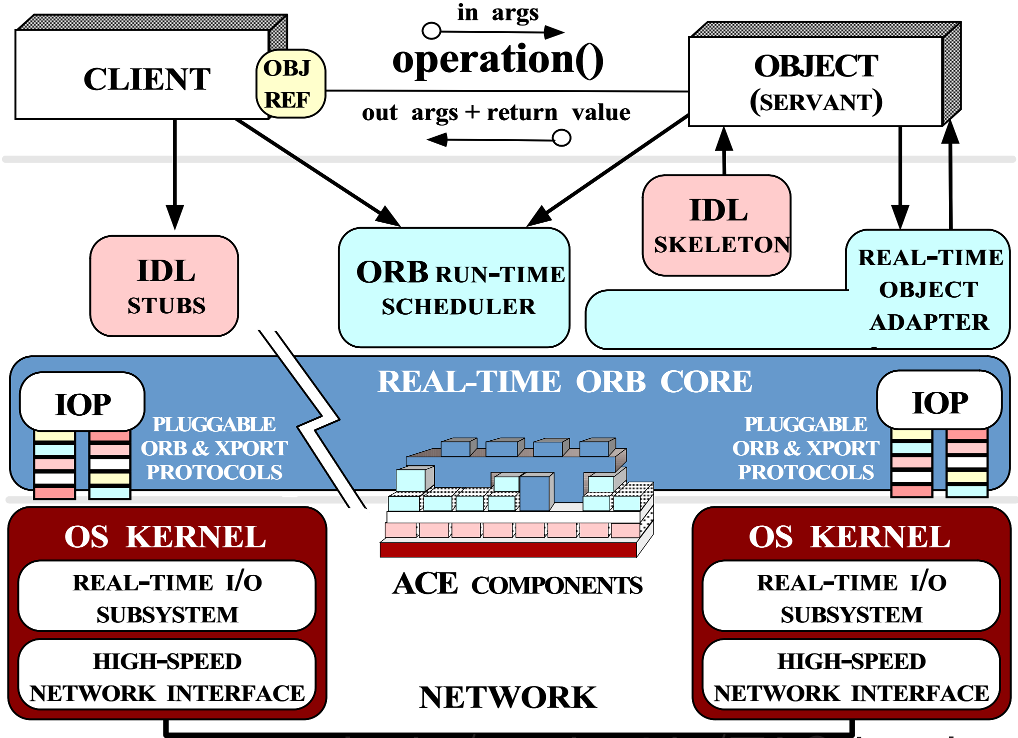
Problem: Meeting End-to-End QoS Requirements



- **Design Challenges**
  - Reducing presentation layer overhead
  - Reducing demultiplexing latency
  - Maintaining small footprint
  - Specifying QoS requirements
  - Meeting operation scheduling deadlines

Related OMG Documents: – ORBOS/97-09-31 (RT CORBA), ORBOS/97-11-08 (Minimum CORBA)

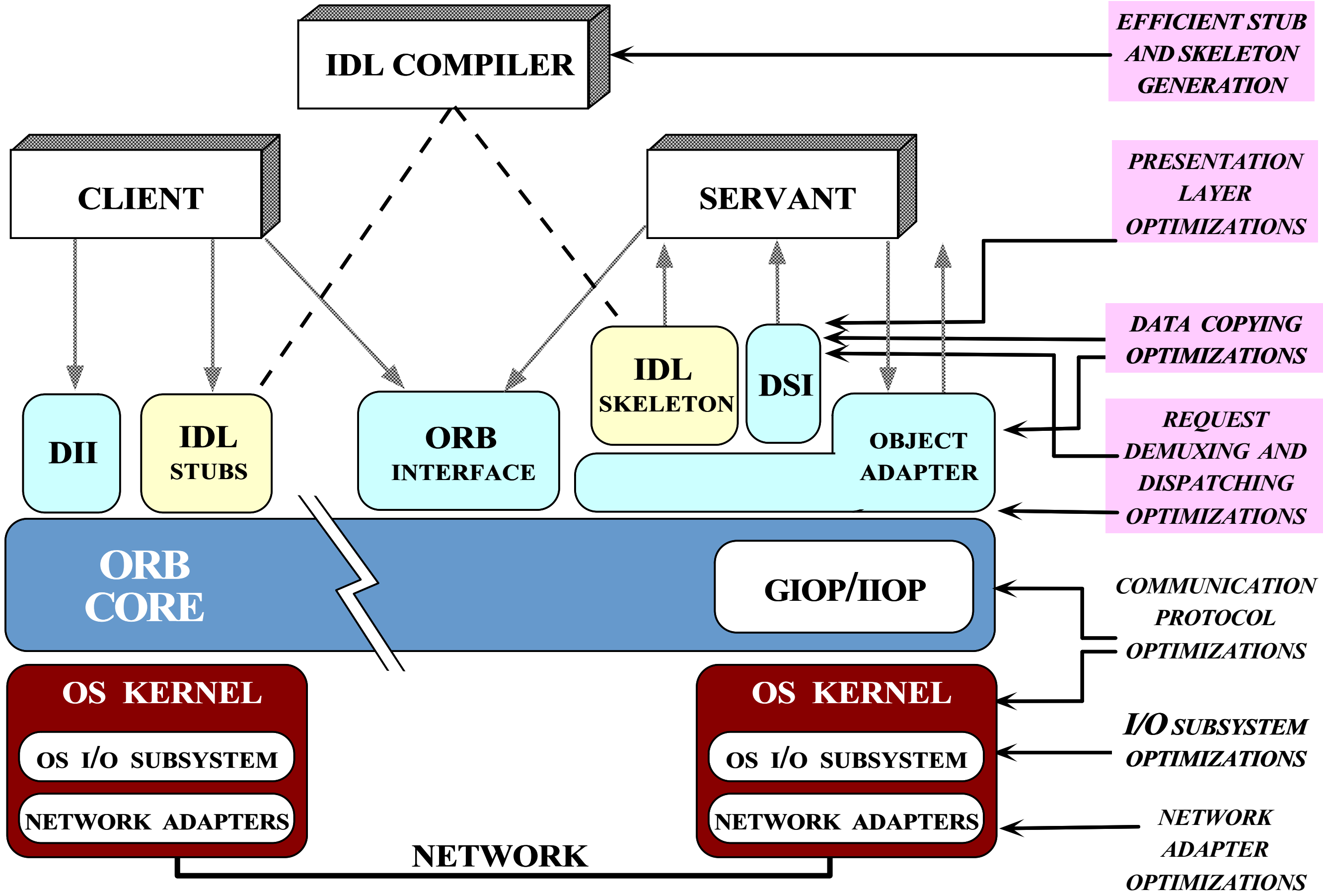
The ACE ORB (TAO)



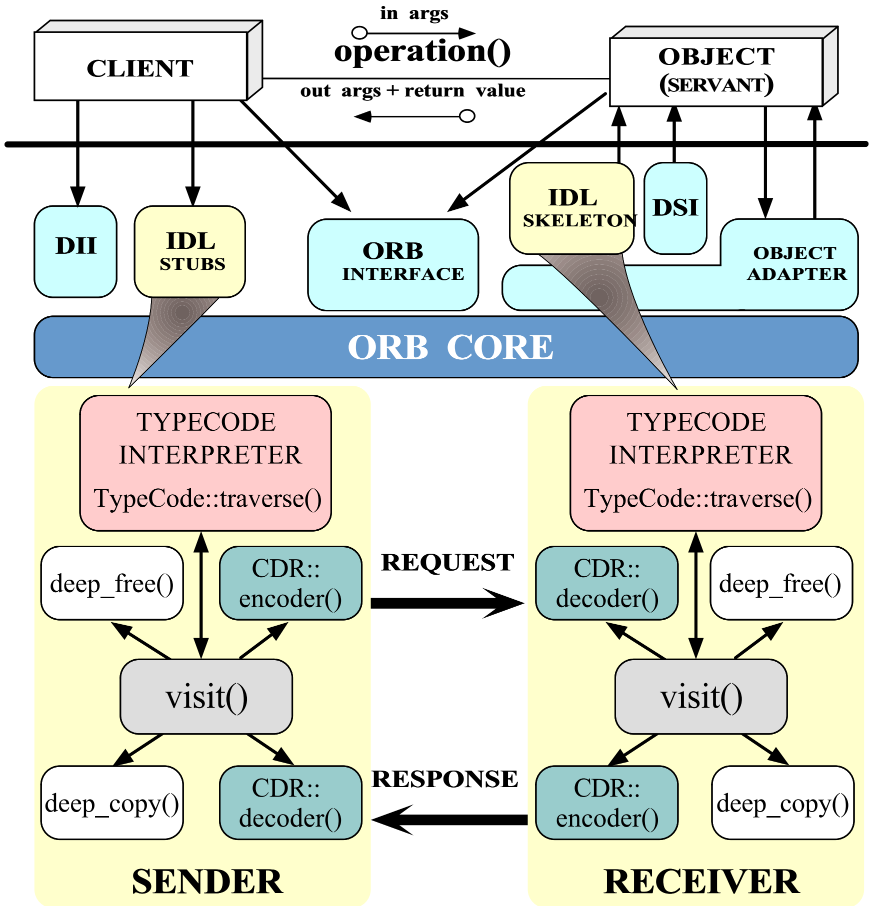
www.cs.wustl.edu/~schmidt/TAO.html

- **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

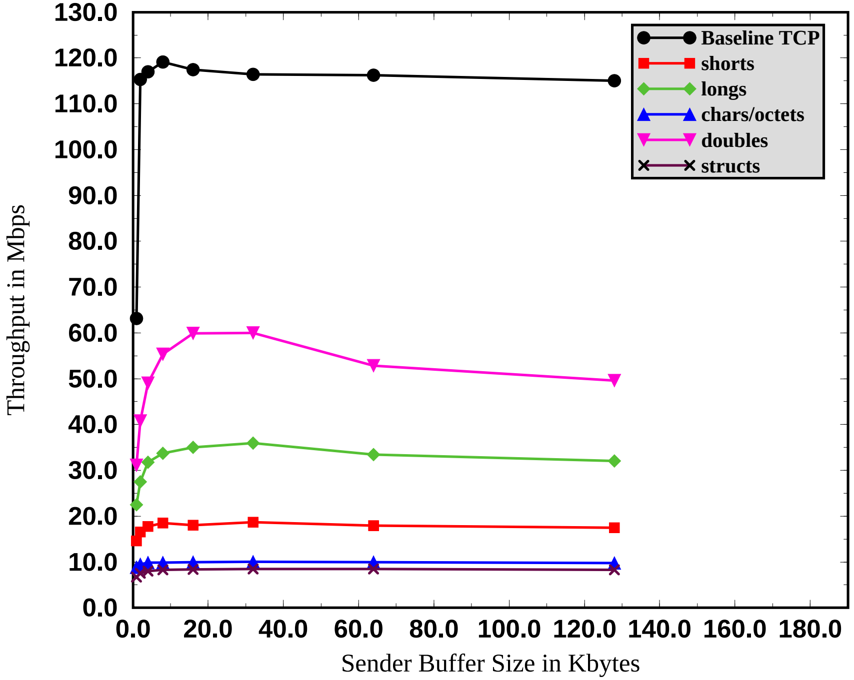


Problem (1): Reducing IIOP Protocol Engine Overhead



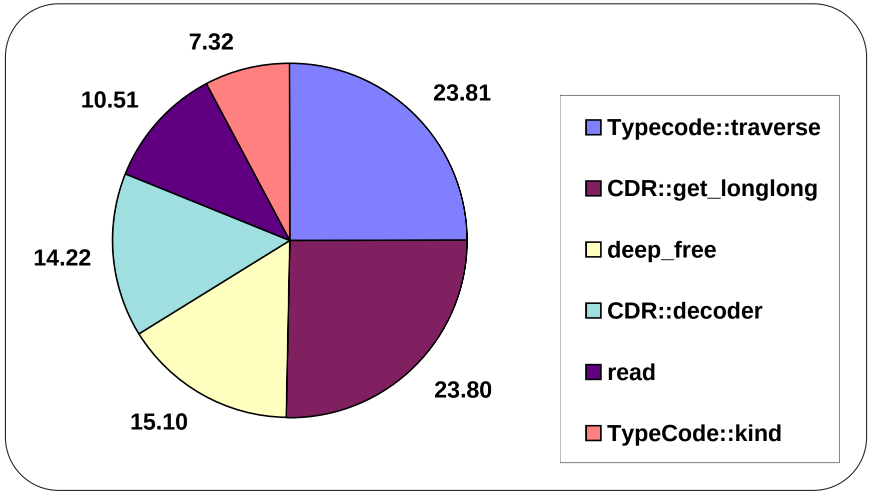
- **Design Challenges**
  - Small memory footprint
  - Predictable, efficient performance
  - Minimize the typecode interpreter overhead
  - IIOP compliant

Throughput of the SunSoft IIOP Implementation

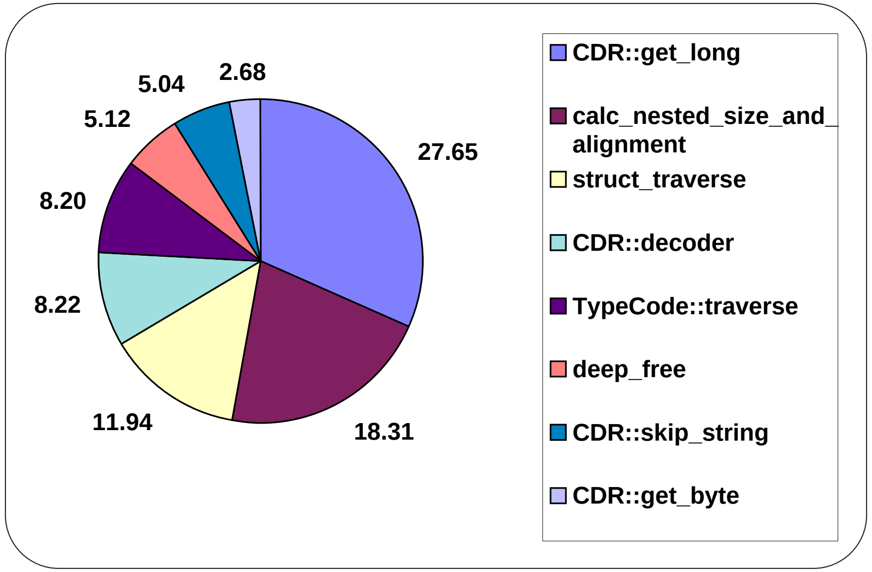


- Experimental design
  - Transfer 64 Mbytes of "oneway" data
  - Various types of data
- Note very poor initial performance

Receiver-side Analysis of SunSoft IIOP Implementation



double



struct

Percent Execution Time for doubles and structs

Solution: TypeCode Interpreter Optimizations

- **Problem**
  - Optimizing complex software is hard
  - Small "mistakes" are costly over high-speed networks
- **Solution Approach (Iterative)**
  - Pinpoint sources of overhead via *white-box* metrics
    - *e.g.*, `Quantify`, `TNF`, etc.
  - Apply optimization principles
  - Validate via white-box and black-box metrics
- **Related work**
  - Hoschka '97
  - O'Malley, Proebsting, and Montz '94 - USC Stub Compiler
  - Eide, Lepreau - Flick IDL compiler

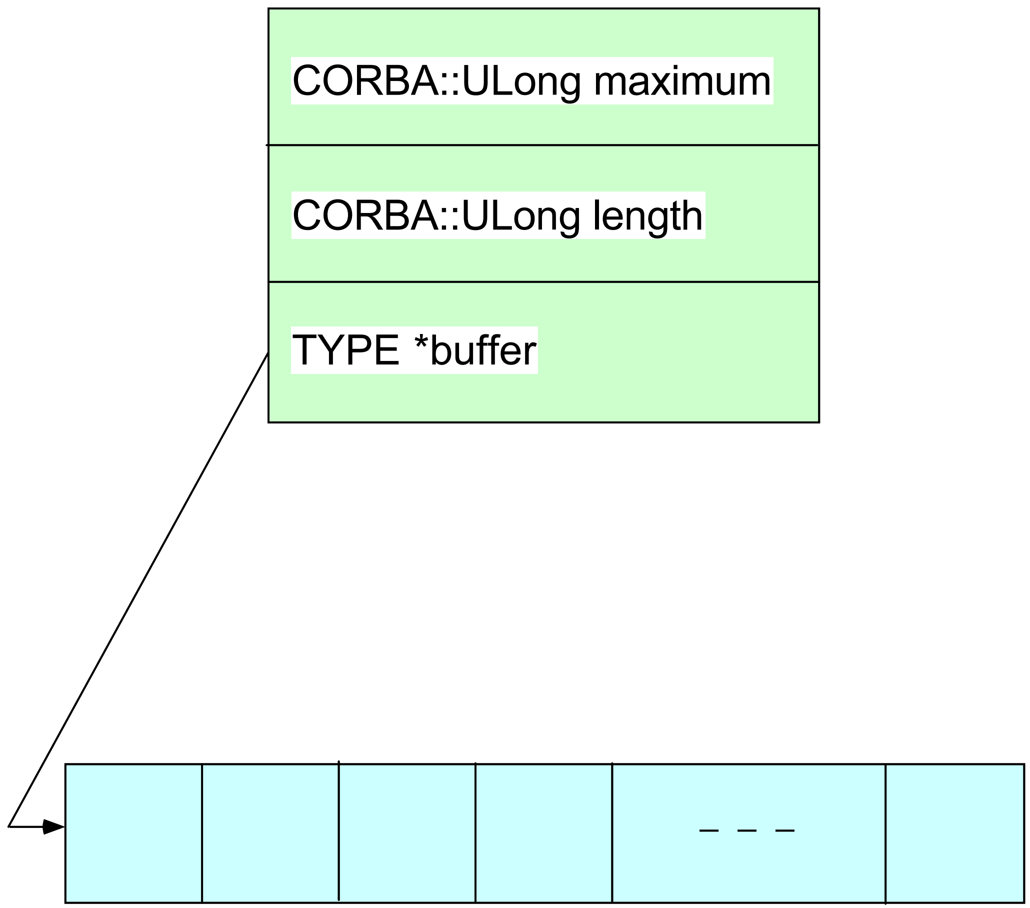
Optimization Principles

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

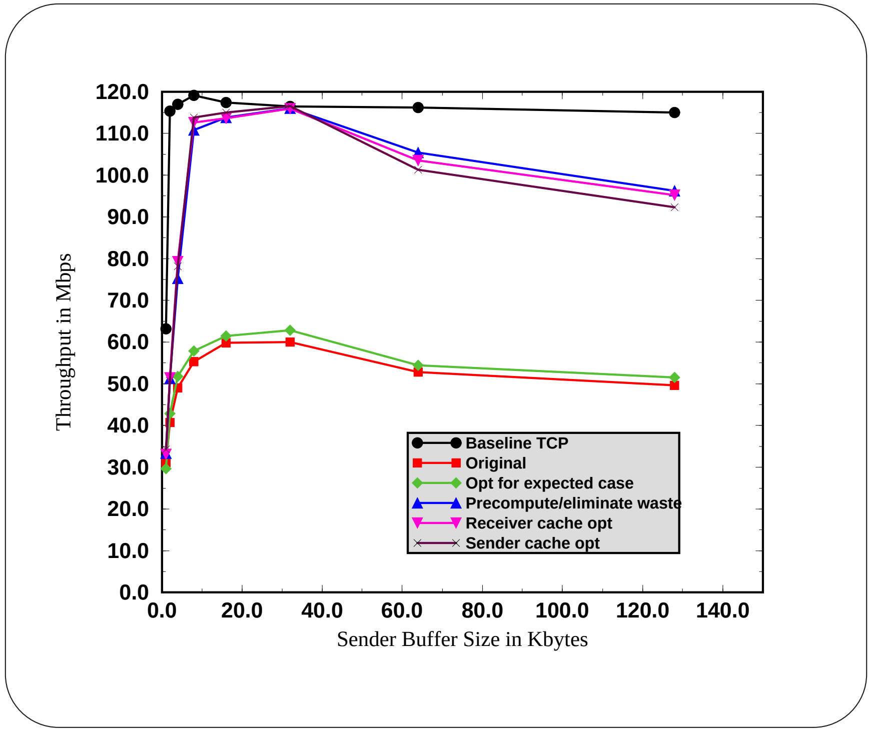
- G. Varghese, SIGCOMM'96
- Clark:90, Abott:93 – ILP
- Peterson:96 – Outlining
- Clark:89 – Header prediction
- Peterson:94 (PathFinder), Engler:96 (DPF), Mahesh:95 (packet filters)

Sequences Mapping

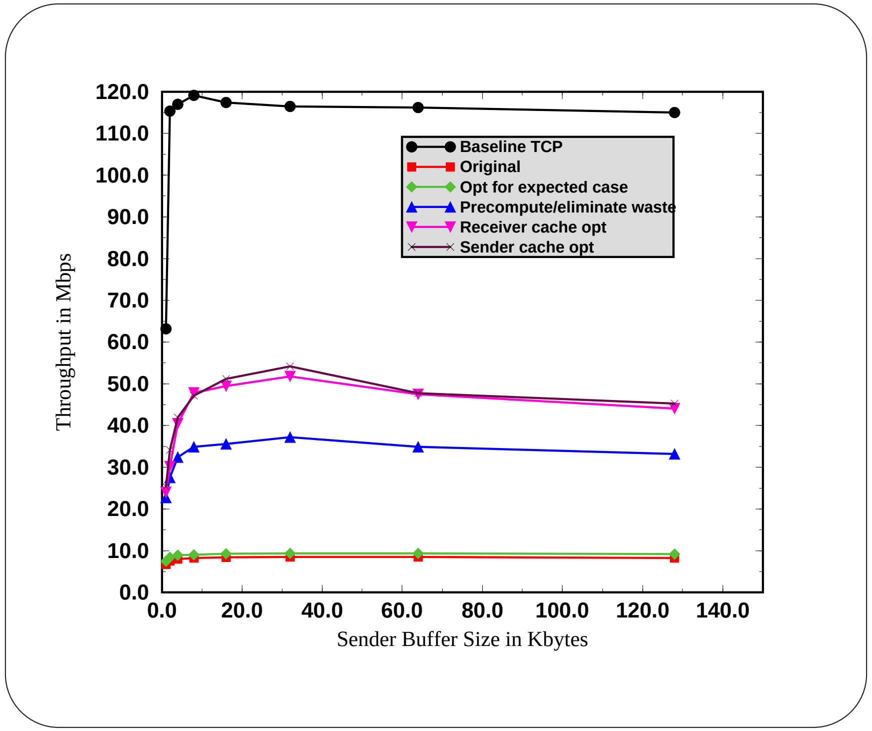


- **Key overheads**
  - Redundant computation of nested size and alignment
  - Wasteful deallocation for primitives
  - Excessive overhead of function calls

Throughput After Optimizations



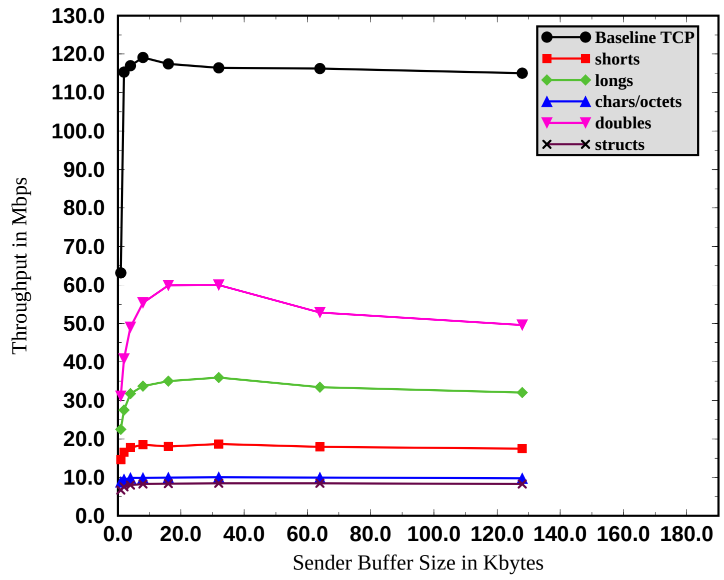
double



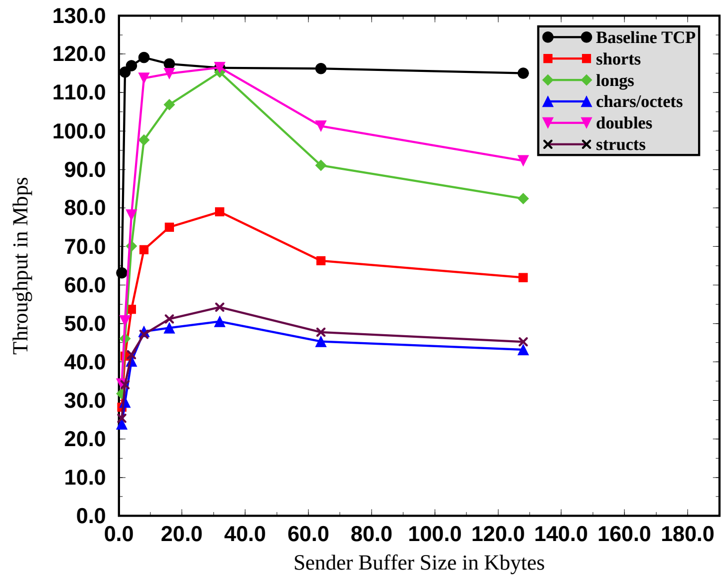
struct

Throughput for doubles and structs

Throughput Comparisons



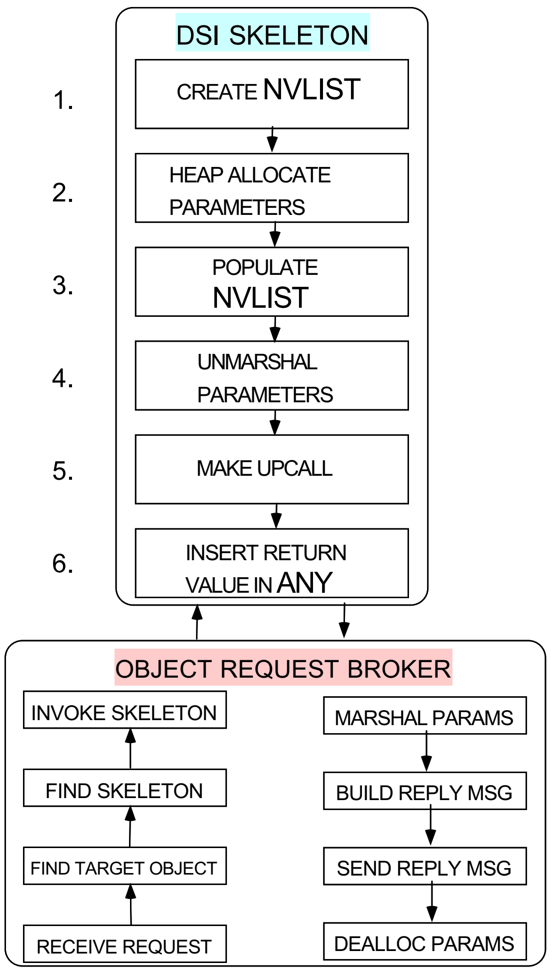
Original SunSoft



Optimized TAO

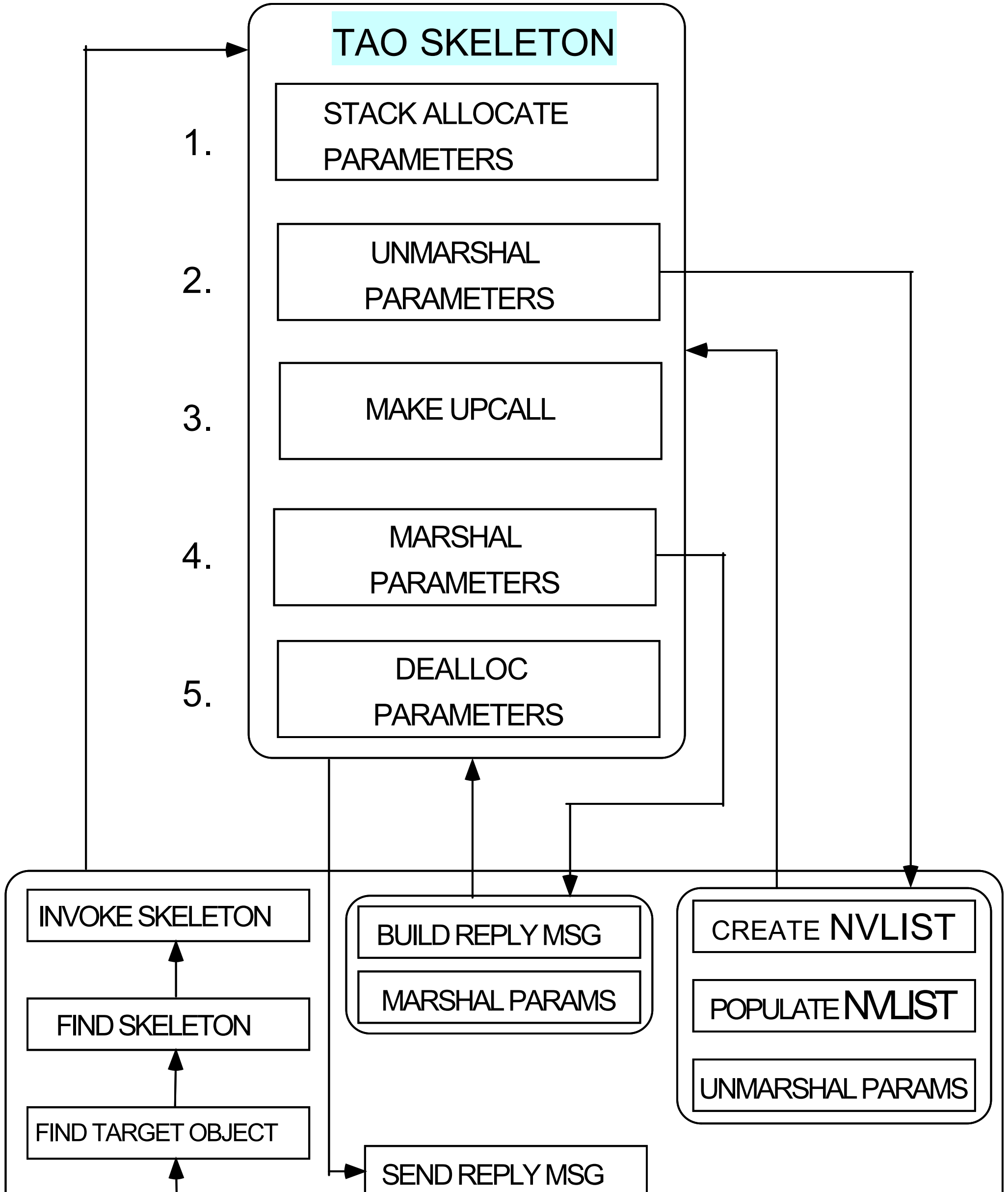
`www.cs.wustl.edu/~schmidt/HICSS-97.ps.gz` **(Best Paper Award)**

Problem (2): Inefficient stubs/skeletons

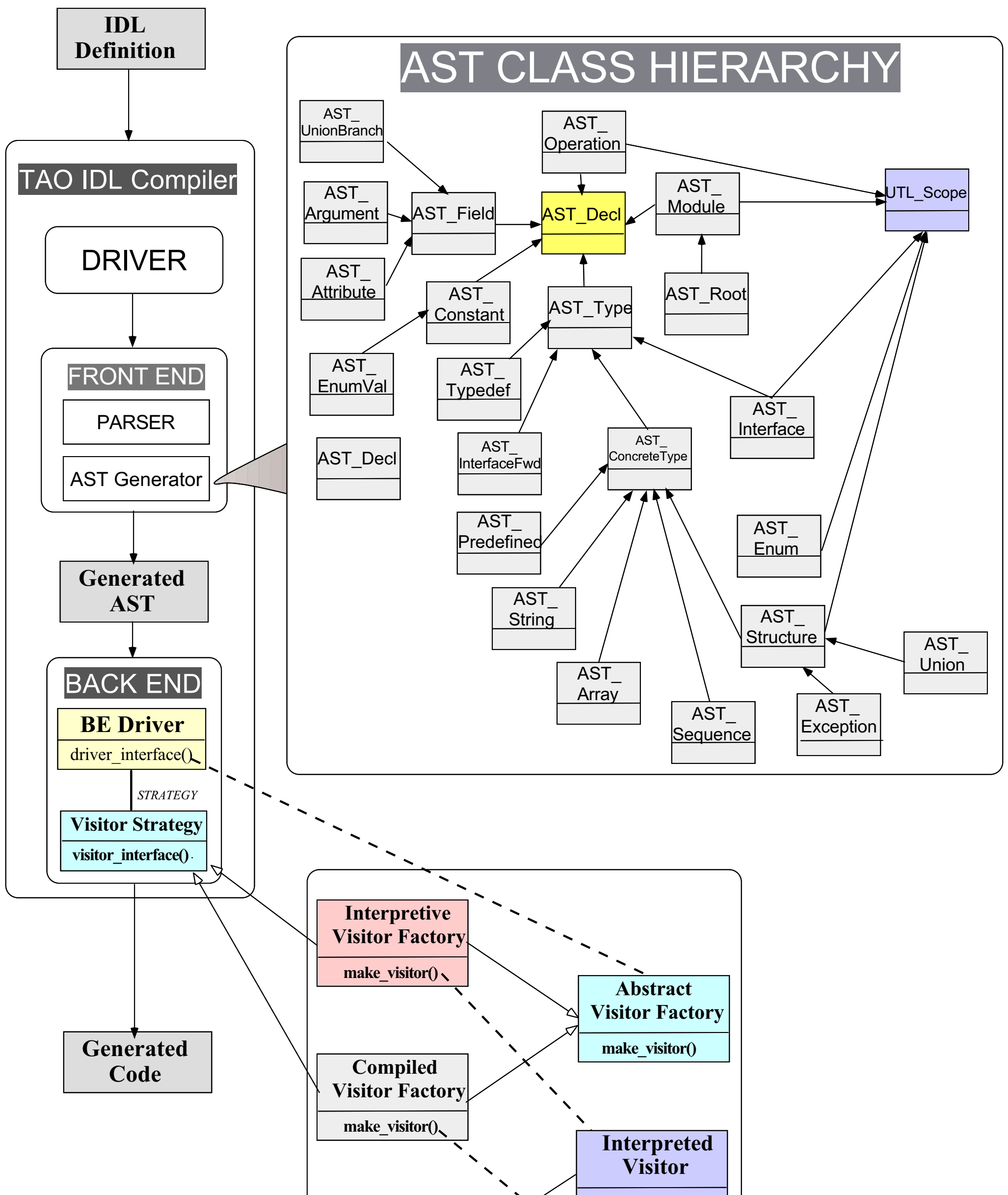


- **Key sources of overhead**
  - Excess heap allocation
  - Repetitive code in every skeleton => large footprint
  - Interpretive deallocation
- **Solution: Principle-based Optimizations**
  1. Eliminate gratuitous waste
  2. Factor out common tasks
  3. Optimize for the expected case
  4. Use compile-time knowledge as much as possible

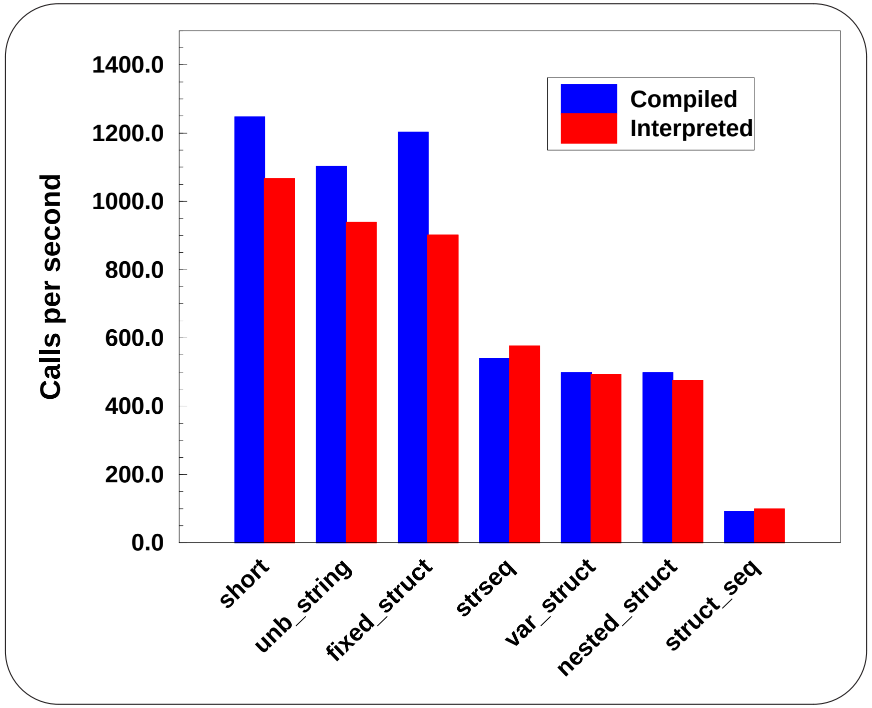
Optimized, Small footprint Skeletons



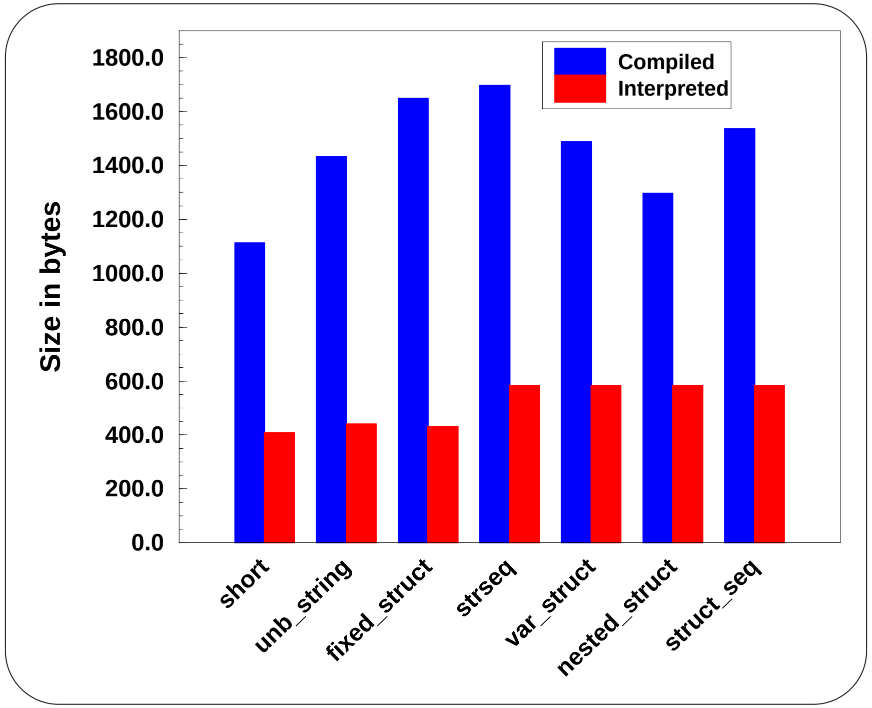
TAO IDL Compiler Design



Results: Compiled versus Interpretive Marshaling



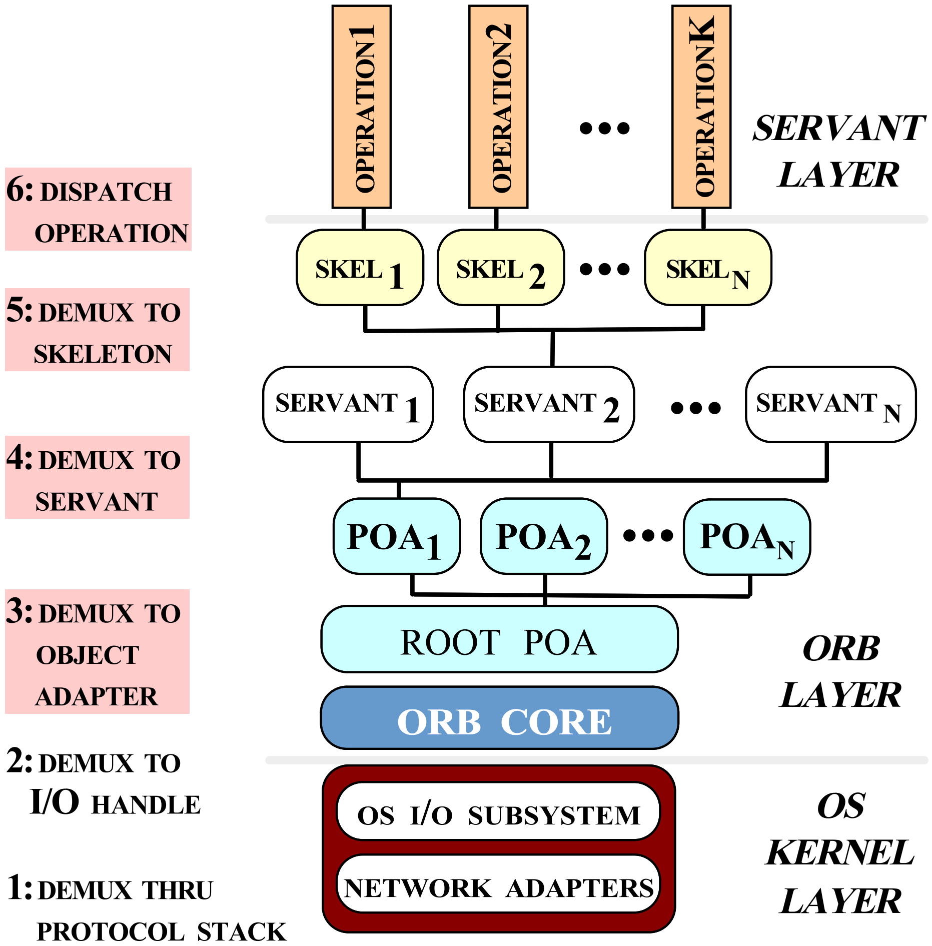
Performance



Footprint

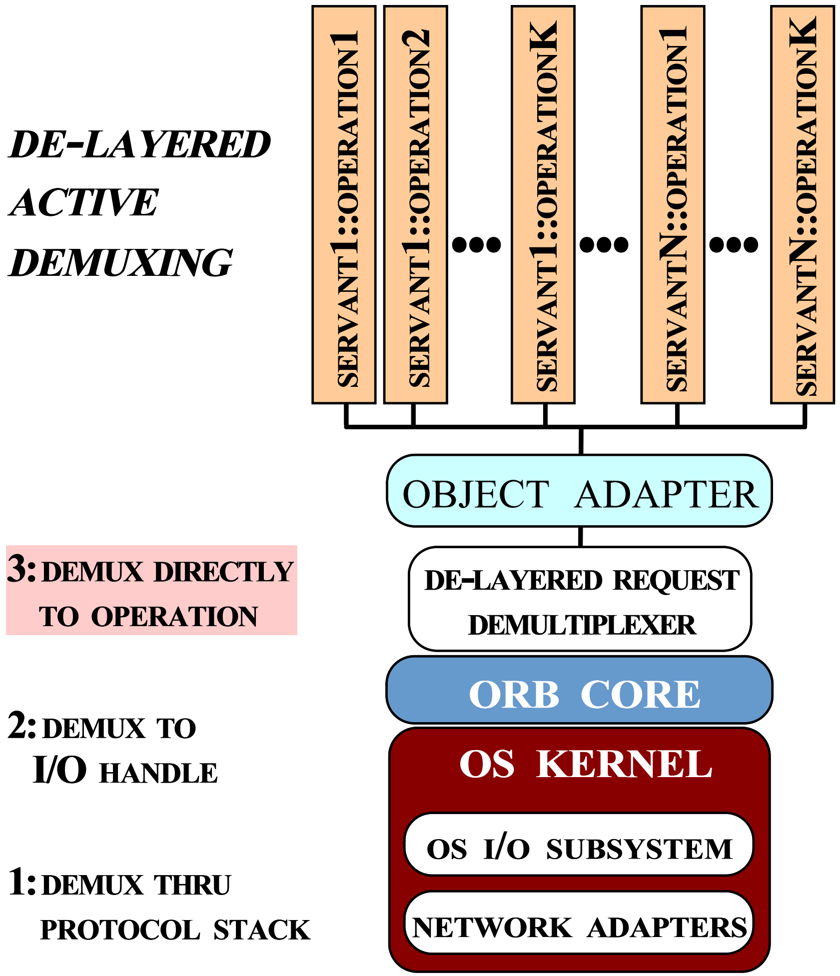
Related work – Hoschka '97, O'Malley'94 (USC Stub Compiler), Lepreau (Flick IDL compiler)

Problem (3): Reducing Demultiplexing Latency



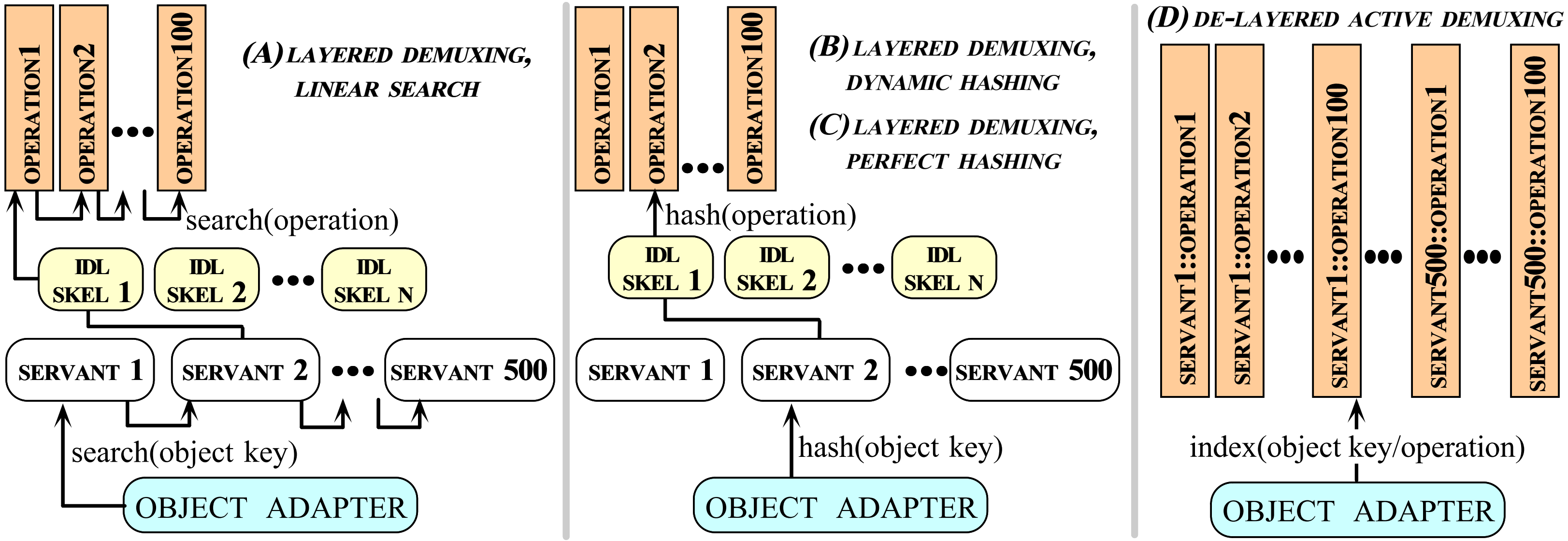
- **Design Challenges**
  - Minimize demuxing layers
  - Provide $O(1)$ operation demuxing
  - Avoid priority inversions
  - Remain CORBA-compliant

Solution: De-layered Active Demultiplexing



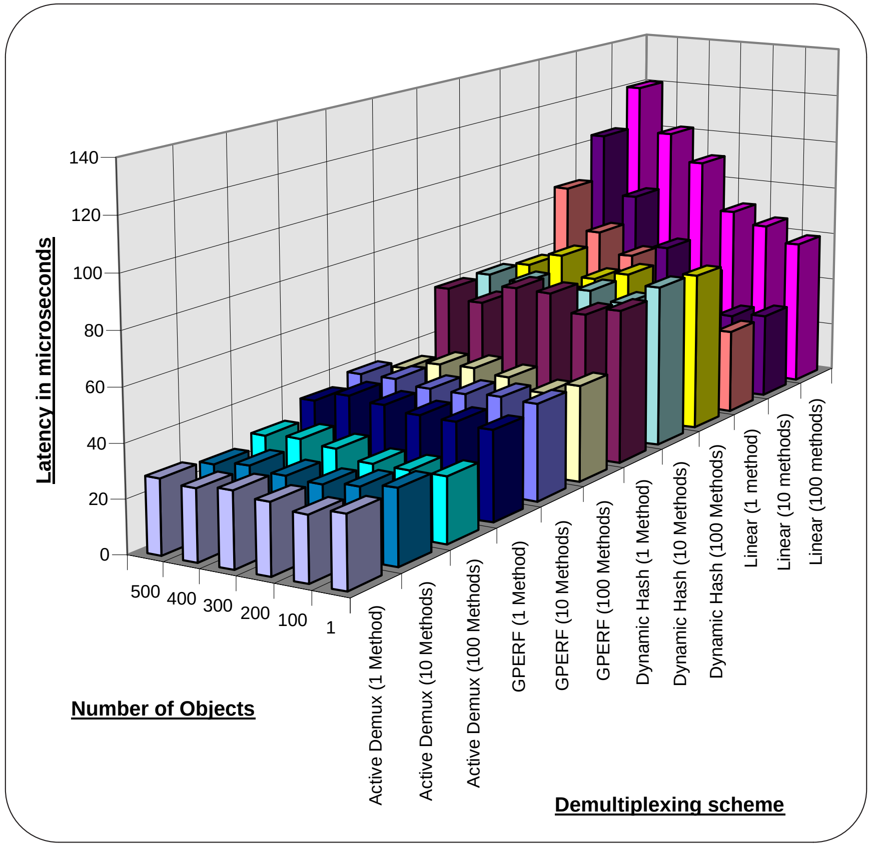
- **Solution Approach**
  - Pre-negotiate demuxing keys
  - Tunnel demuxing key with Object key
- **Related Work**
  - Yau and Lam '97
  - Dittia and Parulkar '97
  - Engler and Kaashoek '96

Demultiplexing Performance Experiments

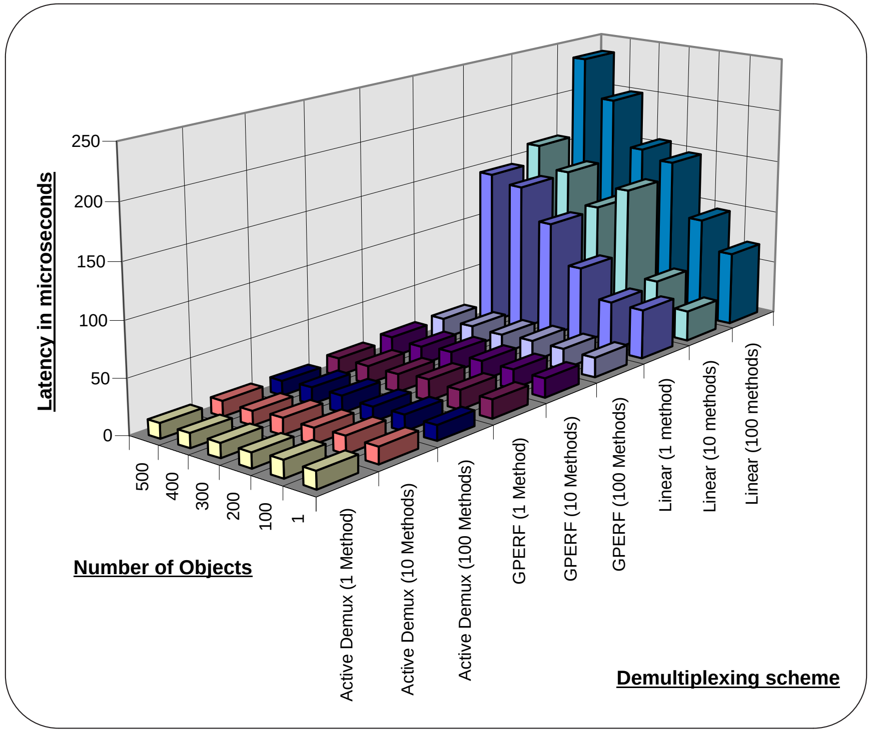


- Linear search based on `Orbix` demuxing strategy
- Perfect hashing based on GNU `gperf`
  - `http://www.cs.wustl.edu/~schmidt/gperf.ps.gz`
- `http://www.cs.wustl.edu/~schmidt/ieee_tc.ps.gz`

Demultiplexing Performance Results



Random



Worst case

Applicability of Demultiplexing Strategies

- `Linear Search` approach is 100% compatible, but very poor performance
- `Dynamic Hashing` approach is 100% compatible, much better than linear search, but not predictable
- `gperf` solution is 100% compatible, but static
- Optimal active demuxing isn't 100% compatible, but is dynamic

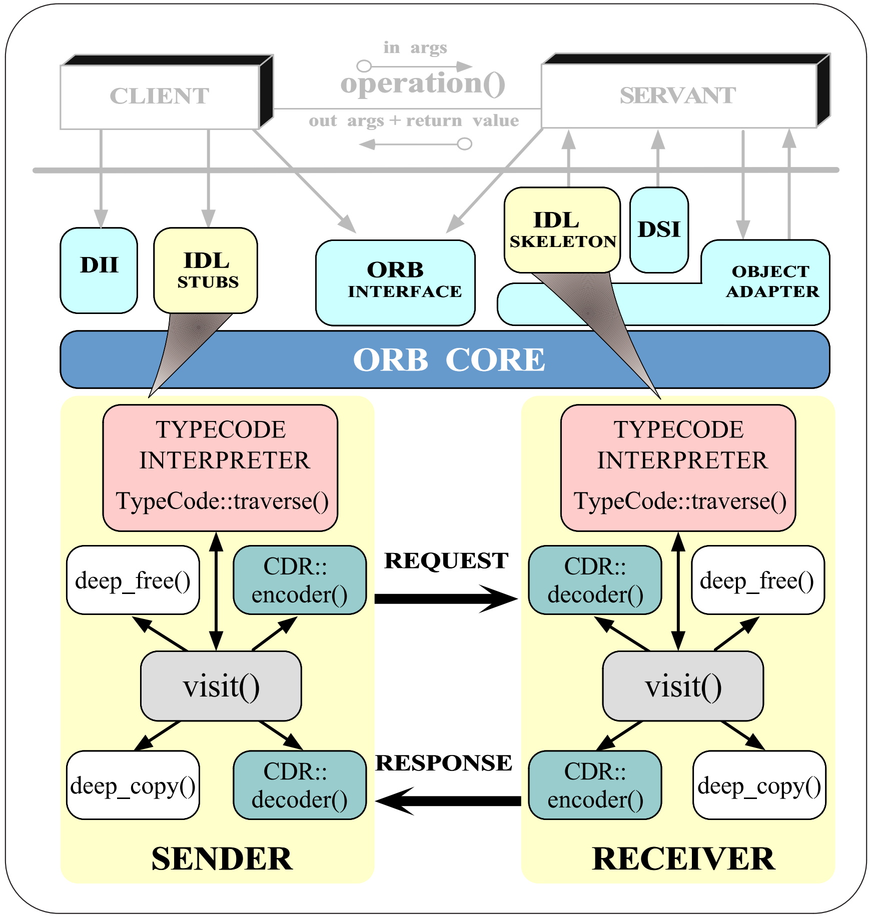
Publications

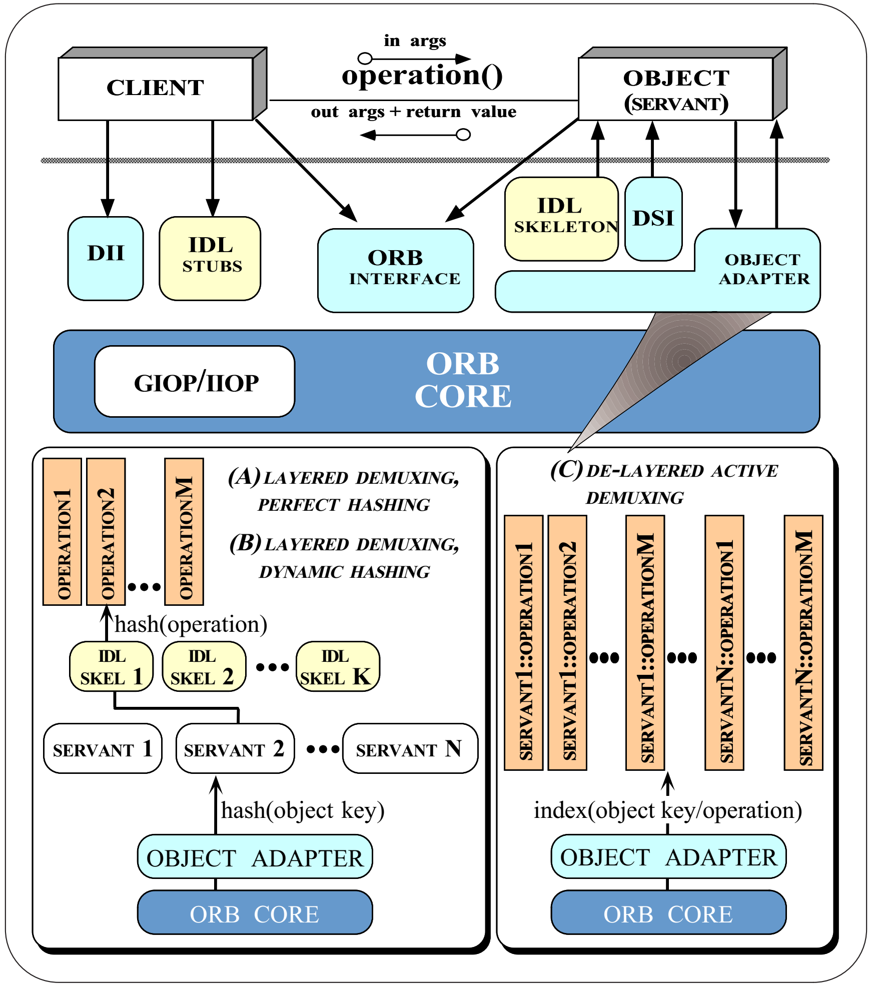
- Journal Papers
  1. *IEEE Transactions on Computers (April'98)* – with D. Schmidt
  2. *IEEE Communications Magazine, Feb'97* – with D. Schmidt, G. Parulkar, T. Harrison
- Conference Papers
  1. *IEEE RTAS'98* – with D. Schmidt, S. Mungee, S. Flores
  2. *HICSS-31, Jan'98 (Received the Best paper Award)* – with D. Schmidt
  3. *IEEE Globecom '97* – with D. Schmidt
  4. *ISS'97* – with D. Schmidt, S. Moyer
  5. *IEEE ICDCS'97* – with D. Schmidt
  6. *IEEE Globecom'96* – with D. Schmidt
  7. *ACM SIGCOMM'96* – with D. Schmidt
- Workshops and Poster
  1. *ACM OOPSLA'97 Poster Session*
  2. *ACM OOPSLA'96 Poster Session*
  3. *IWOOOS'96 Workshop* – with D. Schmidt, G. Parulkar, T. Harrison
- Currently under Review
  1. *IEEE JSAC'99* – with D. Schmidt

Concluding Remarks

- Distributed Software Crisis
  - CORBA - potential solution
  - Lack of real-time features in CORBA
  - The TAO RT-ORB project
- Research Contributions in TAO
  - Reducing latency via *de-layered active demuxing* and *perfect hashing*
  - Applying optimization principles to TypeCode interpreter and presentation layer
  - IDL compiler and optimized stub/skeleton generation
- Results – widely used ORB

Research Contributions Revisited: TAO Optimizations





Future Work

- Generate compiled stubs/skeletons
- Backend emitter framework for OMG IDL compiler
- Integrate IDL compiler with University of Utah's Flick compiler
- ORBs for PDAs and other embedded systems. Issues involved –
  - Small footprint
  - Real-time performance
  - Efficient tradeoff between compiled and interpreted forms of marshaling
- Reliability and Fault-tolerance (at Lucent Bell Labs)

Acknowledgments

1. Parents, Bharati, and relatives for their mental and financial support during extremely difficult times
2. Dr. Guru Parulkar who offered me admission and assistantship in the CS program
3. Dr. George Varghese and Dr. Ron Cytron who advised me for the first couple of years and provided the much needed encouragement and direction
4. Dr. G. James Blaine and Steve Moore of ERL for providing me the much needed employment for survival
5. The CS department for giving me a tuition waiver needed for survival during the first couple of years
6. Dr. Burkhard Stiller for being on my committee and coming all the way from Switzerland
7. DOC group members for all the healthy discussions and help
8. Friends like Girish for the support during some anxious moments
9. And finally - **Dr. Douglas Schmidt** without whom these achievements would have been just a dream

Does TAO mean anything?

Main Entry: Tao
Pronunciation: 'dau, 'tau
Function: noun
Etymology: Chinese (Beijing) dao, literally, way
Date: 1736

1. (a) : the unconditional and unknowable source and guiding principle of all reality as conceived by Taoists
   (b) the process of nature by which all things change and which is to be followed for a life of harmony
2. often not capitalized : the path of virtuous conduct as conceived by Confucians
3. often not capitalized : the art or skill of doing something in harmony with the essential nature of the thing <the Tao of archery>