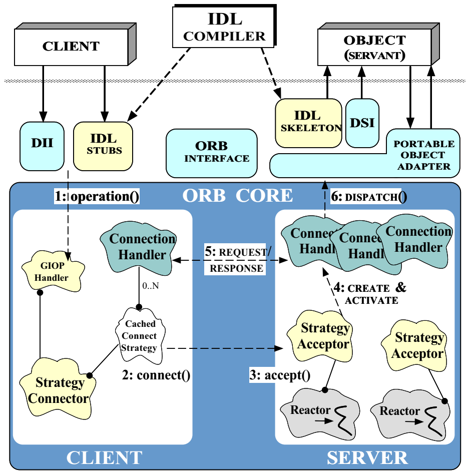
Overview of TAO's ORB Core

Nanbor Wang
Research Assistant
nanbor@cs.wustl.edu
http://www.cs.wustl.edu/∼nanbor/

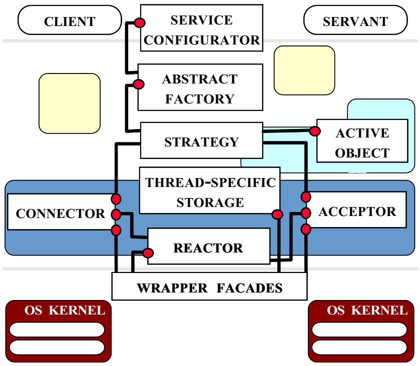
October 13, 1999

Architecture of TAO ORB Core



- Responsibilities of TAO ORB
  - Connection management
  - Request receipt/ forwarding
  - Request demuxing/ scheduling/dispatching
  - Concurrency control

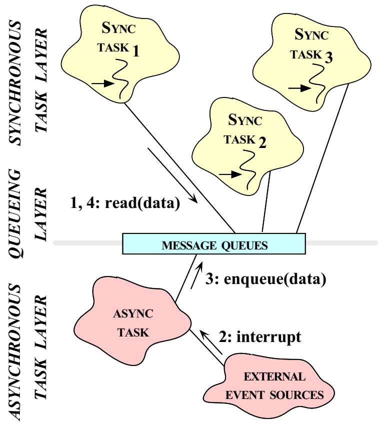
Patterns Used in TAO



http://www.cs.wustl.edu/
∼schmidt/ORB-patterns.ps.gz

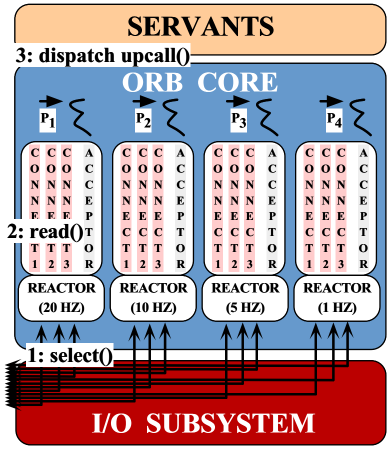
- Factories produce strategies
- Strategies implement interchangable policies
- Service Configurator permits dynamic configuration
- Concurrency strategies implemented using Reactor, Active Object, etc.
- Connector/Acceptor decouple transport type from GIOP operations

Sources of Priority Inversion in Existing ORBs



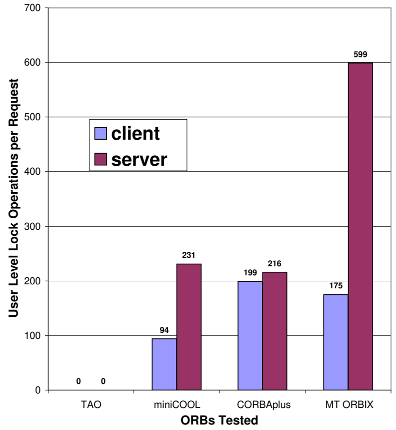
- Half-Sync/Half-Async pattern is ubiquitous
- Used to bridge asynchronous event sources and synchronous threads
- Most existing ORBs add a queueing layer in OA
- Queueing layer causes priority inversion, synchronization overhead, extra data copying and context switching

Configuring TAO's ORB Core for Real-time Systems



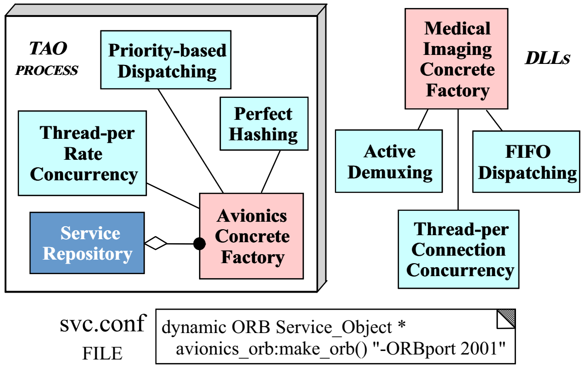
- Integrate endpoint demultiplexing and dispatching
- Minimize priority inversion and non-determinism
- Reduce context switching and synchronization overhead
- Works with other TAO resource management schemes

Minimizing Locking Overhead in TAO



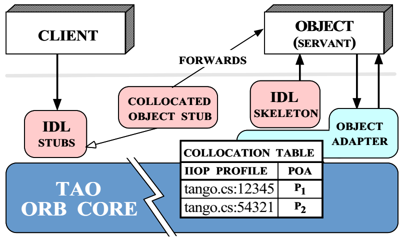
- Locking overhead significantly affects latency and jitter
  - Memory management commonly involves locking
- RT ORBs should minimize or eliminate all locking operations
- TAO is carefully designed to minimize locking and memory allocation

Other Configurable Components in TAO



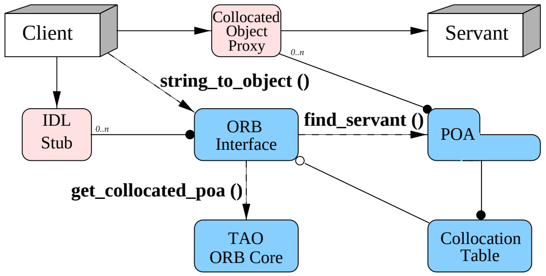
- Configurable resource management tradesoff resource sharing and contention
- Adaptive locking mechanisms enhances flexibility
- ORB strategies can be "scripted"

Implementing Collocation in TAO's ORB Core



- **Motivation**
  - Eliminate overhead if an object resides in the same address space as client
- **Solution**
  - Use collocation tables to check if we are requesting the same object using its IIOP profile, *i.e.*, hostname/port number

Collocation Optimizations in TAO's ORB Core



- **POA** – must provide "local servants"
- **IDL compiler** – must generate both remote stubs and collocated stubs

Future Work

- Support for Thread Pool Reactor
- Support for native C++ exceptions
- Support for IIOP 1.2 spec
- Support for GIOP 1.1 spec
- Support for `CORBA_ORB::perform_work` AND `CORBA_ORB::work_pending` methods