HYDRA: High Speed Simulation Architecture for Precision Spacecraft Formation Flying

AIAA Modeling and Simulation Technologies Conference

B. Martin and G. Sohl

Aug 11, 2003

Motivation

- Upcoming mission concepts place higher demands on simulation environments
  - Starlight: two spacecraft precision formation flying with optics
  - TPF: five spacecraft precision formation flying with optics
  - Lisa concept: hundreds of small spacecraft
- Simulation environments need to evolve with mission requirements
- Architectural areas of concern for simulation environments:
  - Usable and re-usable: allowing one simulation environment to be used for project life cycle (from analysis to hardware testing)
  - Configurable: simulation components are modular
  - Fast: a distributed, multi-CPU environment allows simulation to meet higher fidelity requirements without sacrificing speed

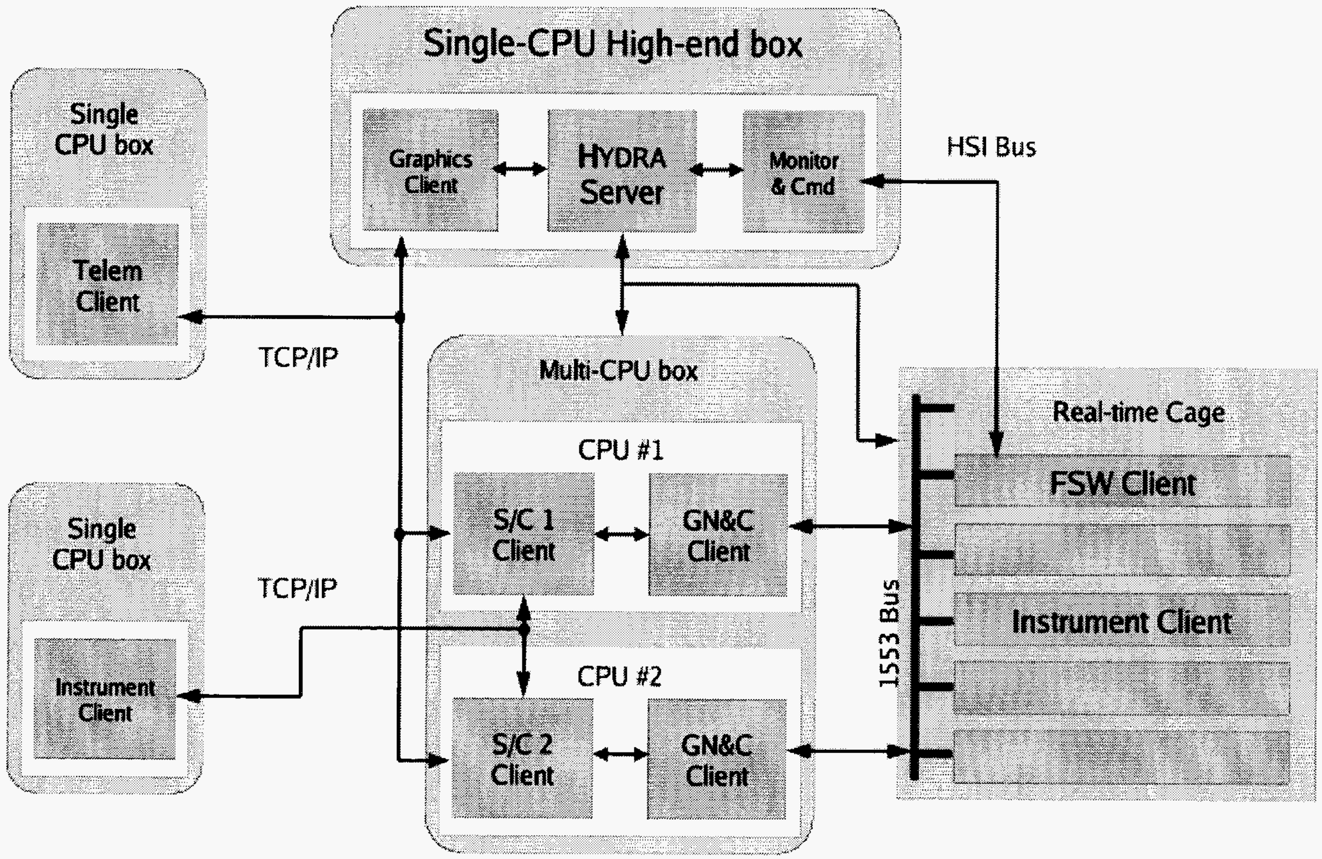
Objective

- Develop an architecture (HYDRA) to facilitate high-fidelity and real-time simulation of formation flying missions.
- Infuse this technology into missions for development and testing of flight control algorithms, flight software, and mission concepts.
- Distributed architectures ("middle-ware") are not a new concept
  - CORBA
  - HLA
  - Wide range of other solutions
- High speed simulation of multi-spacecraft have requirements which were not fully met by existing solutions

HYDRA – Description and Overview

HYDRA: *Hierarchical Distributed Reconfigurable Architecture*

An ongoing architecture for building heterogeneous, distributed real-time applications, with primary focus on high-speed simulation.



HYDRA Overview

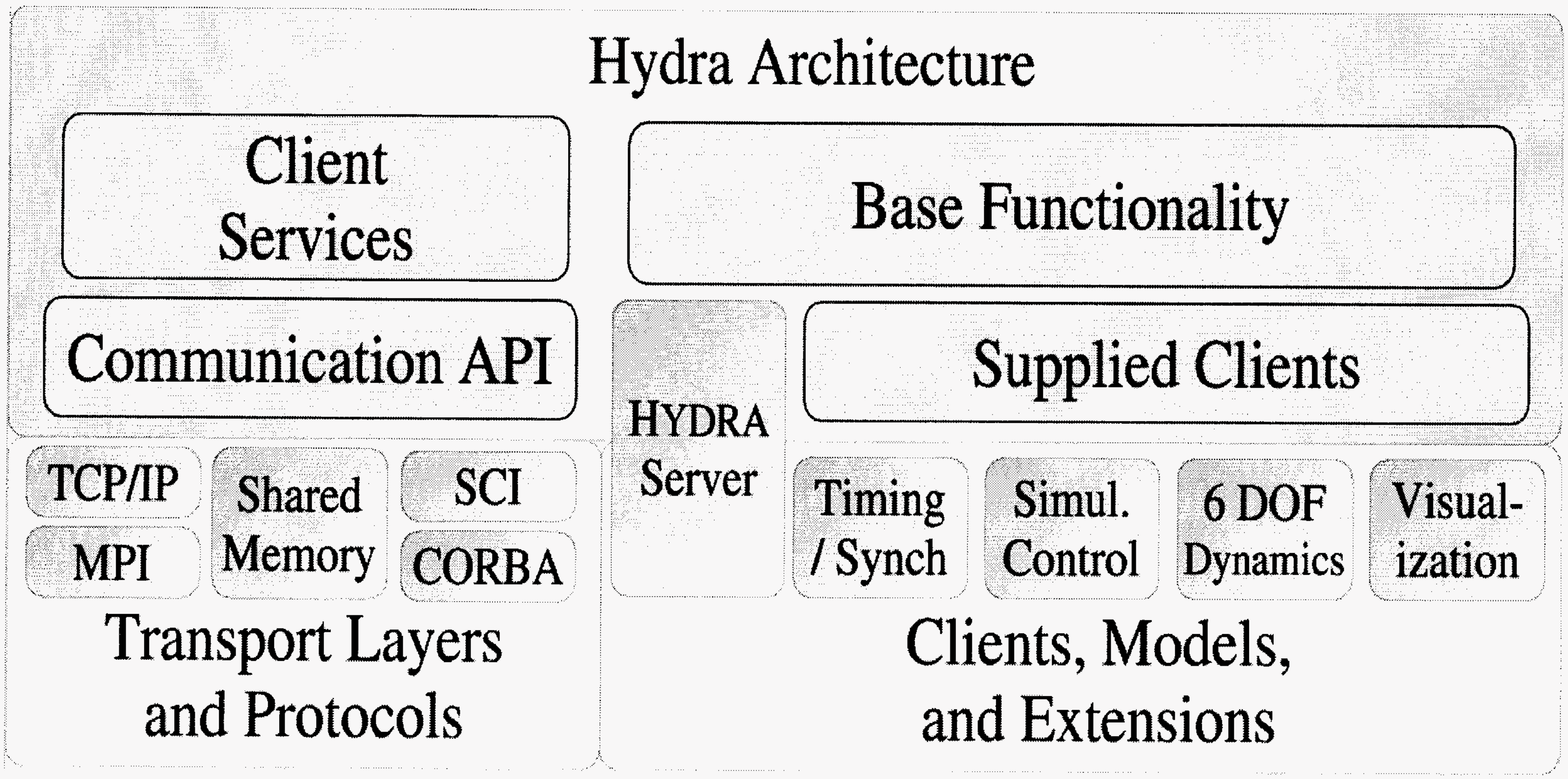
Formation Flying Simulation Research – Inception of HYDRA from desire to solve upcoming challenges in formation simulation.

- Multi-OS client-server architecture with synchronous and asynchronous clients, automates connectivity and inter-client communication, eliminates overhead
- Architecture runs in single-CPU or distributed environment
- Object-oriented design principles for efficient coding & operation, without sacrificing run-time efficiency
- Layered design approach permits overriding behaviors or capabilities to gain efficiency locally,
- Prioritized messages reduce key latencies, automated message and timing tracking improve visibility for testbed control

HYDRA – Hierarchy

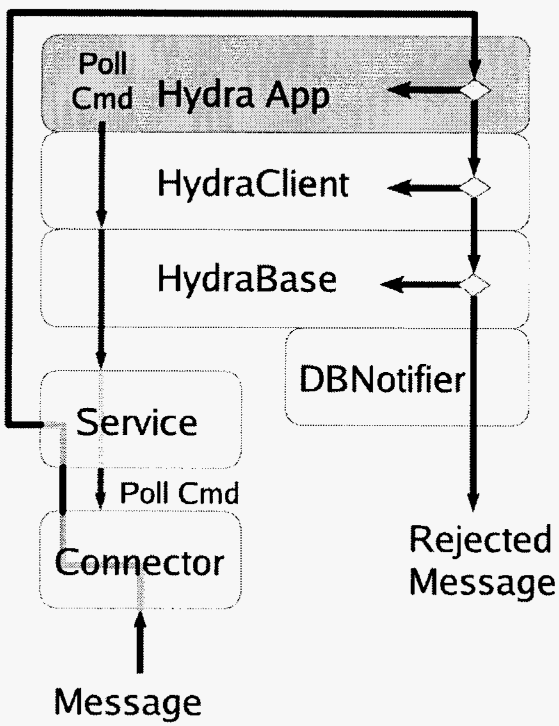
Experience | Maintainable Software
Simulation Testbeds
Reusable Code

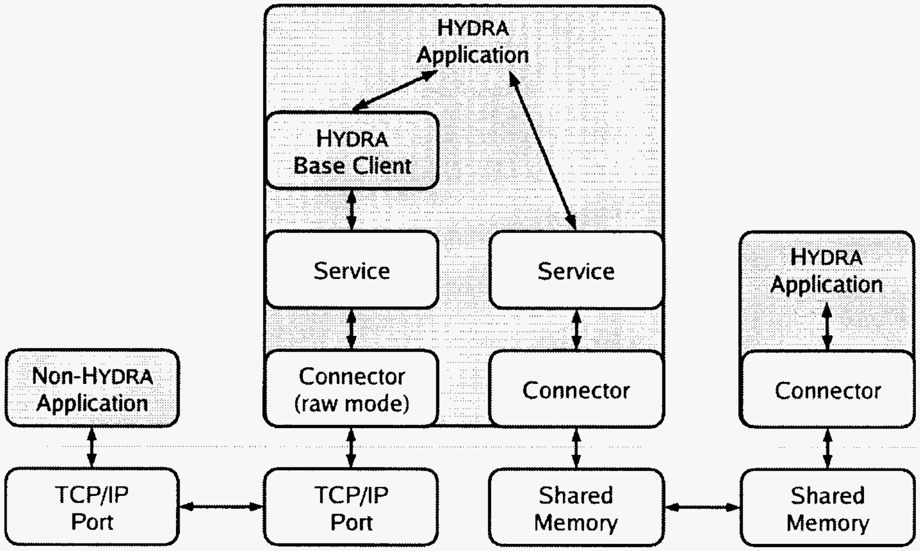
HLA
OOP | Design Philosophies



Communication in HYDRA

- HYDRA facilitates communication between various applications using a variety of protocols (TCP/IP, shared memory, 1553 bus, etc.)



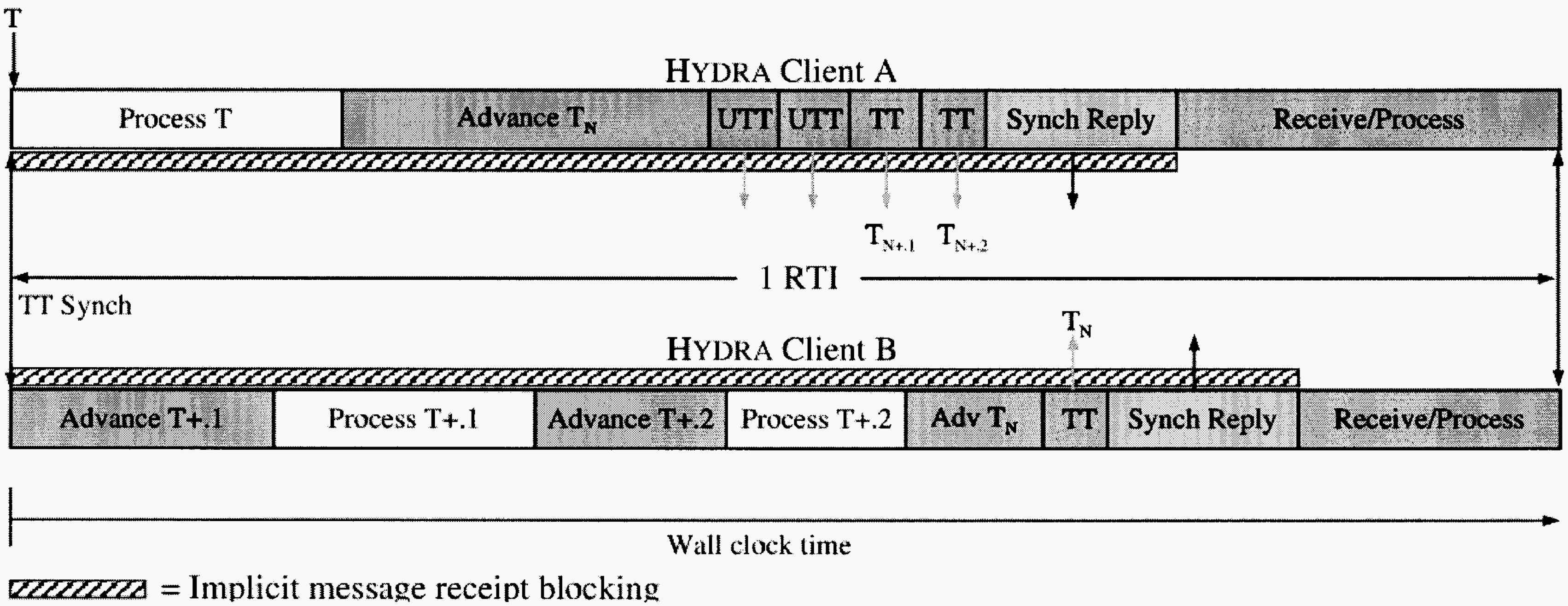


- Each message can be processed by several different hierarchical layers
- User developed application need only process messages of interest (other messages are passed to next layer)

Simulation Specific Concerns in HYDRA

- Distributed simulation applications must interact with a numerical integrator for dynamic state propagation
- This places additional timing constraints on communication between clients – they must occur at the correct simulation time
  - Simulation messages are time tagged based on simulation time
  - Received messages are placed in time ordered queue
  - As the numeric integrator is used to propagate state forward in time, queued message are processed
- Synchronization points (at which all client application integrators are at the same simulation time) are used to ensure that a given client's integrator will not advance past a time when a message should be received

Simulation Specific Concerns in HYDRA (cont.)

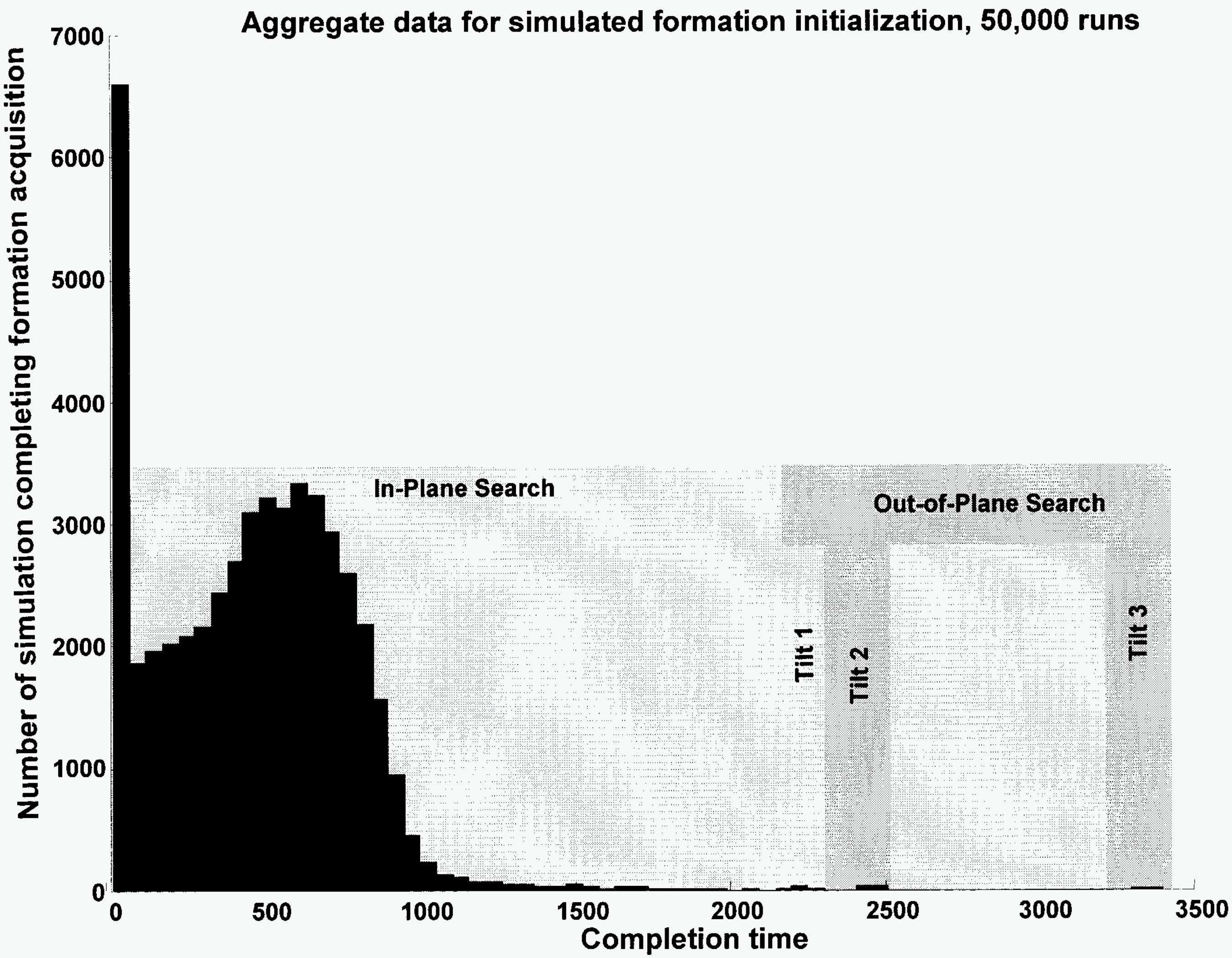


Example Application (Formation Acquisition)

- Testing of formation acquisition algorithm for 5 spacecraft formation
- Precision formation sensors have limited field of view and require face-to-face lock. After spacecraft reset, relative information is lost and sensor lock must be re-established.
- Search algorithms were developed at JPL (S. Ploen, D. Scharf, J. Keim)
- Simulations were used to test the completeness of these algorithms using Monte Carlo testing.
- A Matlab based simulation was developed and executed 25 test cases / hour

HYDRA distributed implementation executed 1800 formation initialization test cases per hour with greater fidelity than Matlab testbed

Sample Problem Results



Conclusion, Current and Future Work

- HYDRA provides a high-speed, lightweight, reconfigurable architecture for connecting disparate simulation elements in either single CPU or distributed environments
- Web based tools for monitoring the status of HYDRA based clients are currently under development
- HYDRA will be used in upcoming simulation projects at JPL:
  - SIM: Space Interferometer Mission
  - FAST: Formation Algorithm Simulation Testbed