Innovation at the speed of NOW.
Measuring Priority Inversions in Real-Time ORB’s

Objective Interface Systems
Overview

◆ **Background Information**

◆ **Priority Inversion Defined**
  – Unbounded
  – Priority Inheritance
  – Bounded

◆ **Real Time ORB Priority Inversions**

◆ **Measuring Priority Inversions**
  – Work done to date
  – Future research

◆ **Conclusion**
Background: Real-Time Scheduling

◆ Tasks
  – Units of schedulable work
  – Assigned priorities

◆ Real-Time Scheduling
  – Preemptive – Lower priority tasks suspended when higher priority task enabled
  – Assignment of priority related to the tightness of deadlines, e.g., higher frequency implies higher priority
  – Fails if tasks miss deadlines
Overview

◆ Background Information

◆ Priority Inversion Defined
  – Unbounded
  – Priority Inheritance
  – Bounded

◆ Real Time ORB Priority Inversions

◆ Measuring Priority Inversions
  – Work done to date
  – Future research

◆ Conclusion
Whenever the highest priority task that is otherwise enabled is not executing:

- Usual cause – contention for limited quantity resource with a lower priority task
- For example, a lower priority task has locked a resource while in a critical section
Unbounded Priority Inversion

Priority

Block

Priority Inversion

Resource

Locked

Time
Priority Inheritance

Priority

Time

Blocke

Unbloc

Priority Inherited

Priority Inversion

locke

unlock
Bounded Priority Inversion

- Priority inheritance helps to bound priority inversion
- But reduces schedulability of application

\[ R_i = C_i + B_i + \sum_{\forall j \in hp(i)} \left\lfloor \frac{R_i}{T_j} \right\rfloor C_j \]

- \( R_i \) – maximum response time of task \( i \)
- \( C_i \) – maximum computation time of task \( i \)
- \( B_i \) – maximum blocking time of task \( i \)
- \( T_i \) – period of task \( i \)
- \( hp(i) \) – higher priority tasks than task \( i \)
Bounded Priority Inversions

\[ B_i = \max_{k,s | k \in lp(i) \land s \in used\_by(k) \land \text{ceil}(s) \geq \text{pri}(i)} CS_{k,s} \]

- \( lp(i) \) – lower priority tasks
- \( used\_by(i) \) – critical section is used by task
- \( \text{ceil}(s) \) – priority ceiling of critical section
- \( \text{pri}(i) \) – priority of task \( I \)
- \( CS_{k,s} \) – execution time of task \( k \) in critical section

◆ Priority inversions may be counted multiple times
Distributed Priority Inversions

◆ Occur when request processing on a remote node must contend for resources with other requests
  – Other clients
  – Other threads

◆ May require distributed priority inheritance to resolve
Overview

◆ Background Information

◆ Priority Inversion Defined
  – Unbounded
  – Priority Inheritance
  – Bounded

◆ Real Time ORB Priority Inversions

◆ Measuring Priority Inversions
  – Work done to date
  – Future research

◆ Conclusion
Sources of Priority Inversions in ORBs

Client

Object Implementation (Server)

Memory Buffers

Connection Tables

Communication Connections

Threads

Active Object Table

Stub

Object Request Broker (ORB)

Object Adapter

Skeleton

Memory Buffers

Object Request Broker (ORB)
Avoiding Priority Inversions in ORBs

- ORBs avoid priority inversion by avoiding sharing resources from multiple threads

- Where contention remains, resource management must be priority aware
  - Priority ordered
  - Priority inheriting
  - Priority ceiling
Overview

◆ Background Information

◆ Priority Inversion Defined
  – Unbounded
  – Priority Inheritance
  – Bounded

◆ Real Time ORB Priority Inversions

◆ Measuring Priority Inversions
  – Work done to date
  – Future research

◆ Conclusion
Measuring Priority Inversion

◆ Possible to build “constructive model”

◆ Need
  – Extensive benchmark or estimation of ORB internal operation
  – Detailed model of ORB behavior

◆ However
  – Requires extensive knowledge of ORB internals
  – Result would be obsolete by product revision

◆ Measurement seems to be required
Washington U St. Louis
- Measured Round Trip Latency of High Priority Task
- Added lower priority tasks
- Yields - Figure of Merit
  - Useful for comparing ORB implementations
  - Slope may be useful as indicator of blocking factor
- Doesn’t separate user local from blocking factor
Measuring Priority Inversion

Previous Work

- **Rttaskdemo (Objective Interface)**
  - Shipped as part of ORB*express* RT
  - Instrumented multi-priority benchmark
  - Yields
    - Throughput and variability at each priority level
    - Demonstrates lack of unbounded inversions
    - Unclear applicability to schedulability analysis
Measuring Priority Inversion

Previous Work

- **HARTStone Benchmark (SEI)** —
  - Series of
    - Sets of Periodically Scheduled Tasks
    - Performing Synthetic Loads
  - Varied
    - Phasing
    - Loads
  - Tracked deadlines
Measuring Priority Inversion

- **ORBstone (Objective Interface)**
  - HARTstone adapted to RT CORBA
  - Compare remote and local operations
  - Estimate blocking factor from regression analysis
  - To be added as a product demo