

Joint RT QoS Task Plan

Dock Allen

The MITRE Corporation

Convener of the Joint Steering
Team

Vision

- End-to-end top-to-bottom timeliness based on an open integrated QoS framework that is:
 - supported by metrics and measurements
 - adaptive
 - language and infrastructure neutral
 - conformance tested
- QoS specifications that are defined in application terminology

Vision

- End-to-end top-to-bottom timeliness based on an integrated QoS framework that is open, measured, and supported by conformance test
- QoS specifications that are defined in application terminology

Team

- Dock Allen, The MITRE Corporation
- Dave Emery, The MITRE Corporation
- Sally Long, The Open Group
- Dave Loundsbury, The Open Group
- Joe Bergman, The Open Group
- Jean Hammond, Quarry Tech.
- Joe Lipasec, Raytheon
- Norm Eagleston, Sun
- Robert Allen, Boeing
- Art Robinson, STDC

The State of Affairs

- Some vendors say that they have solved the problem
- AND major users say that we do NOT know how to build these systems such that we know that they will work
- Note: (Boeing et al paraphrased) as you web enable an application, you uncover its inherent deadlines (due to orders of magnitude increases in temporal jitter)
 - In other words, it's gonna get worse

Background: Software Enclaves

- There seem to be 3+ non-overlapping software enclaves
 - Procedural programming
 - Database programming
 - Parallel programming
 - (Safety Critical systems are becoming a separate enclave)
- These enclaves have their own methodologies, tools, languages, infrastructures, programming communities, vendors, standards, etc
 - There is remarkable little overlap and crossover between them
 - And we need to serve more one of them

Common Ground

- Many concerns apply to all enclaves
 - timeliness
 - performance
 - availability
 - fault tolerance
 - software reliability
 - etc
- Common solutions may be applied to multiple enclaves

Background: Res Temporalis

- Traditional (meaning MY definition) “real-time”
 - **time critical** data and operations
- Streaming data in “real-time”
 - **temporal latency** and **temporal pacing**
- “Real-time” interactive systems
 - acceptable **temporal responsiveness** between participants
- Other concerns
 - Correct **temporal ordering** of requests/data
 - **Temporal coherence** of data/requests from different sources
 - **Temporal conditions** in work flow
 - Accuracy of **global temporal views**
 - Impact of fault tolerance on temporality

Background: QoS

- QoS originated to handle network bandwidth and line quality
- Current standards (RT CORBA and extensions to the Java Language) and research (e.g. DARPA QORUM) are using QoS for timeliness
- We want an integrated approach that spans infrastructure layers

Background: Application QoS

- Application view of QoS: Tracking
 - QoS = track quality and the number and type of dropped tracks
 - The QoS policy is situational - as you go between surveillance and engagement
 - Other applications (such as stock monitoring and trading) follow a similar pattern
- Application view of collaborative decision making
 - QoS = all participants see identical data
- There are a set of QoS patterns and QoS policies from an application perspective, which can be mapped onto infrastructure QoS

Proposed Tasks and Objectives

- Objectives
 - Standards and guidelines for integrated QoS including dependable timeliness
 - Application QoS patterns
 - Metrics for integrated QoS
- Tasks
 - An Open Group Challenge
 - Open Group guidelines
 - Standards as needed

Plan

- Distinguish the unmet needs through open community discussion
- Work on application QoS and QoS policy patterns and match to enclaves
- Define goals and objectives for an Open Group challenge
- A challenge on end-to-end top-to-bottom
- Document the resulting guidelines and initiate any needed standardization

Things to Consider

- Temporal policy languages
- Adaptive resource management
- Multi-resource management
- Cross enclave support (especially procedural and database enclaves)
- Utility functions and other advanced scheduling approaches
- Situational policy management
- Emerging Service Based Architectures (e.g. web services)

Next Steps

- Host opening shots in Boston Joint Session
 - Sun
 - Boeing
 - other vendors and users
 - possibly split sessions per enclaves
- Start preparatory work
 - Application QoS Study by small group
 - Build challenge team

Concerns

- Are BIG users ready to cooperate (has the problem been unsolved long enough to make Mutually Assured Survival attractive)
- Can we find a leader for the challenge
- It's a LOT of work
- ---
- ---