Strategized, Coordinated Services for Real-Time Middleware

Lisa Cingiser DiPippo, Victor Fay-Wolfe, Matthew Murphy, Priyanka Gupta, Jiangyn Zhang, Jianming Ye, Kevin Bryan, Tim Henry

Sponsors:

University of Rhode Island

DARPA
EXO
Paces
Science & Technology
Distributed end-to-end middleware real-time enforcement using global scheduling information

- Coordinated middleware services
- Coordinated scheduling points
- Priority-based scheduling
- Classic real-time analysis and algorithms
URI implements coordinated distributed end-to-end real-time as middleware common object services.
Many Scheduling Points in End-To-End Chain

Client

1. BSS or Spawn
2. USS
3. ESS

Server (E2E-1)

0. RT Trader (Binding) Service
1. IDL Stubs
2. RT Trader Service Context
3. Notification Service
4. IDL Skeletons
5. Object Adapter
6. Object (Servant)
7. Mutex
8. Operation ()
9. NIC
10. NIC
11. RTC2 Scheduler
12. ORB Core
13. NIC
14. RTC2 Scheduler
15. NIC
16. Notification Service

Distributable Thread (E2E-N)

17. Object (Servant)
18. Mutex

Many Scheduling Points in End-To-End Chain

Client Server (E2E-1) Distributable Thread (E2E-N)
URI RT CORBA 1.0
Scheduling Service

Offline tools set parameters for scheduling service; service sets parameters in RT ORB

Scheduling Service

Sets CORBA Priority

Sets Priority Inheritance, Priority Lanes, Priority Bands

Sets OS Prio Mapping

Sets Network Prio Mapping

Priorities, ceilings, mappings, XML file
Dynamic Distributed Scheduling Service

Scheduling Service works with local schedulers to enforce global scheduling

Distributed Scheduling Service:
- Set end-to-end scheduling parameters for local schedulers using global criteria
- Detect overload
- Determine distributable thread cancellation points
Distributed Scheduling Problem
Space Taxonomy

System
- Local Only
- Static
- All Hard
- All Periodic
- E2E-1 (per request)
- Hard/Soft
- Periodic/Sporadic
- E2E-N
- Dynamic
- All Soft
- All Sporadic
- Remote Calls
- System Conditions
- Constraints
- Periodicity

Identify application and system characteristics
RT CORBA 1.0 Scheduling Service

- Handles local scheduling and one CORBA call, static systems, all forms of constraints depending on the offline capability

  *Example*: scheduling static UAV video stream
Dynamic Scheduling Service: First Problem Space Approached

- **System**
  - **Local Only**
  - **Static**
  - **All Hard**
  - **All Periodic**
  - **Periodic/Sporadic**
  - **All Soft**
  - **All Sporadic**
  - **Dynamic**
    - **E2E-1** (per request)
    - **E2E-N**

- **Example**: dynamically entering UAVs each with periodic video stream.

- First dynamic problem space - *dynamic periodic*

University of Rhode Island
Distributed Scheduling Service Strategization Points

Priority
• Rate Mono
• EDF
• MUF

Overload
• Load Shedding
• QoS adjustment
• Resource reallocation

Scheduling Service

Local Dispatch
• Priority Mapping
• RTC2 local ORB scheduler

Server Dispatch
• Priority Inheritance
• Release-Guard
• E2E intermediate deadlines

Network Routing
• Net Priority Mapping
• Network EDF with intermediate deadline

RT Trader (Binding) Service

Binding
• Best Fit Utilization
• Worst Fit Utilization
• Best fit laxity

System

E2E-1 (per request)
E2E-N
Local Only
Static
Dynamic
All Hard
Hard/Soft
All Soft
All Periodic
Periodic/Sporadic
All Sporadic

University of Rhode Island
Distributed Scheduling Service Strategization Points

Priority
- Rate Mono
- EDF
- MUF

Overload
- Load Shedding
- QoS adjustment
- Resource reallocation

Local Dispatch
- Priority Mapping
- RTC2 local ORB scheduler

Server Dispatch
- Priority Inheritance
- Release-Guard
- E2E intermediate deadlines

Network Routing
- Net Priority Mapping
- Network EDF with intermediate deadline

Binding
- Best Fit Utilization
- Worst Fit Utilization
- Best fit laxity

RT Trader (Binding) Service

Current Work – December 03 target
RT CORBA 2 Local Scheduler Interaction With Distributed Scheduling Service - Partial Class Diagram
Scheduling Point #1 - begin_scheduling_segment
Scheduling Point #2 - update_scheduling_segment

1: uss

1.1: uss

1.1.1: calculate_util()

1.1.2: update_task

1.1.3: schedulability_test

1.1.3.1: get_utilization

1.1.3.2: 

1.1.4: if overload: cancel_task

1.1.4.1: Thread_CancelException

1.1.5: get_sched_param

1.1.6: 

1.1.2.1: update_scheduling_segment

1.1.2: update_scheduling_segment

1.1.3: 

1.1.1: calculate_util()
Scheduling Point #3 - end_scheduling_segment
Scheduling Service

System Repository

Global Scheduling
- begin_global_scheduling_segment()
- calculate_util()

System Repository
- update_task()
- begin_task()
- get_utilization()

Overload Detector
- schedulability_test()

Overload Handler
- accept_task()
- deny_task()

DS_Adapter
- begin_new_scheduling_segment()

Scheduling Algorithm
- get_sched_param()
Real-Time Data Distribution

Supplier provides real-time data to consumers

Consumer specifies real-time requirements for data delivery

Scheduling service determines scheduling parameters for propagation of event data

RT NS Event Channel

Direction of Event Flow

EC - CosEvenComm
ECA - CosEvenChannelAdmin
(Pure Event Service)
NC-CosNotifyComm
NCA-CosNotifyChannelAdmin
*-multiple object instances supporting this interface
Model Integrated Scheduling - Approach

Modeled Problem Space Taxonomy in GME2000
Model-Integrated Scheduling - Current

1. Tri-Pacific model interpreter
2. XML file
3. Reads
4. RapidRMA Analysis
5. URI model interpreter
6. XML file
7. Modified BBN QuO model interpreter
8. Reads
9. Scheduling Service
10. RT Trader (Binding) Service
11. Scheduling
12. Aspect
13. Delegates
Model-Integrated Scheduling On-going Work

U Virginia

VEST/TimeWiz Integration

VEST (U Virginia)

Determine Scheduling Problem Space

System

Local Only

E2E-1 (per request)

E2E-N

Dynamic

All Soft

All Hard

Static

Hard/Soft

All Periodic

Periodic/Sporadic

All Sporadic

Scheduling Service

RT Trader (Binding) Service

Scheduling Aspect Delegates

Model-Integrated Scheduling On-going Work

U Virginia

VEST/TimeWiz Integration

VEST (U Virginia)

Determine Scheduling Problem Space

System

Local Only

E2E-1 (per request)

E2E-N

Dynamic

All Soft

All Hard

Static

Hard/Soft

All Periodic

Periodic/Sporadic

All Sporadic

Scheduling Service

RT Trader (Binding) Service

Scheduling Aspect Delegates
Summary

- Creating Real-Time Services
  - Scheduling, Binding, Data Distribution
- Embodying classic real-time algorithms in middleware
- Creating new algorithms as needed
- Using existing services for framework and enforcement
Strategized, Coordinated Services for Real-Time Middleware

Lisa Cingiser DiPippo, Victor Fay-Wolfe, Matthew Murphy, Priyanka Gupta, Jiangyin Zhang, Jianming Ye, Kevin Bryan, Tim Henry

Contact
Vic Fay-Wolfe (wolfe@cs.uri.edu)
Lisa DiPippo (dipippo@cs.uri.edu)
http://rtdoc.cs.uri.edu

Sponsors: