Configurable Real-Time Middleware for Distributed Cyber-Physical Systems

Chenyang Lu
CSE 520S

Motivation

- Cyber-Physical Systems (CPS) require integrated design of computing & physical systems.
- Challenge: Diversity of CPS applications
  - Avionics, automobile, manufacturing, medical, power grid...
  - Different CPS applications need different middleware configurations.
  - Existing real-time middleware provides fixed sets of services.
    - Real-Time CORBA, Real-Time Java, CORBA Component Model.
- Goal: configurable middleware for diverse CPS applications.
  - Tailor middleware services to specific needs of a CPS application.
  - Facilitate integrated design of CPS.

Outline

- Middleware architecture
- Alternative service strategies
  - encapsulated in configurable middleware components
- Map CPS application characteristics to service strategies
  - supported by configuration tools
- Implementation and empirical evaluation

Application Model

- End-to-End Task $T_i = \text{chain of subtasks} (T_{i,1}, T_{i,2}, ..., T_{i,k})$
  - Aperiodic or periodic
  - Subject to end-to-end deadline
  - Example: Subtask triggered by event from predecessor
- Job: an instance of a task

Middleware Architecture

Task manager
  - Admission Control (AC)
  - Load Balancing (LB)
Application processors
  - Idle Releasing (IR)
  - Task Effector (TE)

Admission Control Strategies

- Admission test based on aperiodic utilization bound (Abedulahmet06)
  - Guarantee end-to-end deadlines of admitted tasks/jobs.
- AC per Task
  - Perform the admission test for an entire task when it arrives.
    - Example: Digital control.
    - Reserve capacity for all jobs of an admitted task $\rightarrow$ no job skipping.
    - More pessimistic admission test.
- AC per Job
  - Perform the admission test for each job of a task.
    - Example: Non-critical image acquisition.
    - No reservation for a task $\rightarrow$ may skip some jobs of a task.
    - Less pessimistic admission test.
Load Balancing Strategies

- Redirect events to replicas located on least loaded processors
- Light weight: No state synchronization among replicas.
- LB per Task
  - The path of a task is determined upon arrival → same path for all jobs.
  - Examples: Integral control, video.
  - Achieve state persistency between jobs.
  - Less performance benefit.
- LB per Job
  - Different jobs may be redirected to different paths.
  - Examples: Proportional control, image acquisition.
  - No state persistency between jobs.
  - More performance benefit.

CPS Applications ➔ Services

- Can tolerate job skipping?
  - Per Task AC (example: digital control)
  - Per Job AC (example: image acquisition)
- Component Replication?
  - No load balancing
  - Load balancing
- Require state persistency between jobs?
  - Per Job LB (example: Proportional control)
  - Per Task LB (example: Integral control)

Configuration Space

- 15 valid configurations ➔ difficult to configure manually!
- Some combinations are invalid: Per Task AC vs. Per Job LB

Configuration Tools

- Input: Application characteristics.
  - Does your application allow job skipping? [yes (Y), no (N)]
  - Does your application have replicated components? [yes (Y), no (N)]
  - Does your application require state persistence? [yes (Y), no (N)]
- Configuration Engine
  - Generate XML-based deployment plan
  - Avoid invalid combinations of strategies
- Deployment Engine (DAncE) [Deng07] executes deployment plan.

Component Middleware

- Based on OAO 0.6 [Wang04], open-source implementation of Light Weight CORBA Component Model (CCM) specification.
- Implemented real-time services as configurable components.
  - Supports real-time, aperiodic and periodic, end-to-end tasks.
**Experimental Platforms**

- **harry.cse**
  - Pentium4 2.53GHz
  - 1G RAM
  - 512KB cache
  - KURT-Linux 2.4.22

- **hermoine.cse**
  - Pentium4 2.80GHz
  - 1G RAM
  - 512KB cache
  - KURT-Linux 2.4.22

- **norbert.cse**
  - Pentium4 2.53GHz
  - 1G RAM
  - 512KB cache
  - KURT-Linux 2.4.22

- **ron.cse**
  - Pentium4 2.80GHz
  - 1G RAM
  - 512KB cache
  - KURT-Linux 2.4.22

- **neville.doc**
  - Pentium4 3.40GHz
  - 2G RAM
  - 2048KB cache
  - KURT-Linux 2.4.22

- **angelina.doc**
  - Pentium4 3.40GHz
  - 2G RAM
  - 2048KB cache
  - KURT-Linux 2.4.22

**Imbalanced Workloads**

- **AC_IR_LB**
  - **N**: None
  - **T**: Per Task
  - **J**: Per Job

- Easy to generate different configurations.
- Middleware configurations have significant impact on real-time performance.

**Conclusions**

- Configurable real-time middleware
  - Configuration tool maps application characteristics to middleware configurations
  - Components middleware implement configurable services
  - Facilitate integrated design of diverse CPS applications

**Reference**