A Software Approach for Managing Shared Resources in Multicore IMA Systems

Xavier Jean, Marc Gatti, David Faura. Thales Avionics, France
Laurent Pautet, Thomas Robert. Telecom ParisTech, France

2013/10/10
DASC 2013 - Syracuse
Context

Problem Statement

Approach for Shared Resources Management

Prototyping

Conclusion
Digital Avionic Systems Evolution

- Processing Requirements
  - Home designed processors
  - COTS processors
  - Federated architecture
  - Integrated architecture
  - Single-core processors
  - Multicore processors

- Concorde
  - ATR-42
  - A320
  - ATR-72
  - SSJ-100
  - A380
  - B787
  - A350

- Time:
  - 1960 - 1980
  - 1980 - 2000
  - 2000 - 2020
  - 2020 - 2040
Integrated Modular Avionics concept

- **Set of Hardware and Software components**
  - Modular development
  - Incremental certification

- **Dependability constraints**
  - Worst Case Execution Time computability
    - Safety of WCET computation
  - Failure Isolation: Robust Partitioning
    - Modularity of WCET computation

- **Platform efficiency**
  - Raw Performances for each Partition
  - Number of Hosted Partitions
- **ARINC 653 Partitions Deployment**
  - Asymmetrical Multi-Processing
  - Backward compatibility on legacy
  - No global constraint on schedule

- **Hardware Resources Allocation**
  - Private vs Shared resources
  - Interleaving of concurrent transactions in the interconnect

- **Inter-Core Conflicts Occurrences**
  - Resources sharing policy driven by the hardware
Context

Problem Statement

Approach for Shared Resources Management

Prototyping

Conclusion
Problem Statement

◆ How to compute a WCET?
  ○ Simulate the core’s worst case behavior executing the application
  ○ Consider any access to a shared resource as taking its Worst Case Access Time (WCAT)

◆ Problem: How to determine WCAT to shared resources?
  ○ No constraint on embedded partitions
  ○ No guarantees on a minimal bandwidth granted to each core
  ○ In practice we observe pathological situations

Hardware management of shared resources accesses seems not safe
Approach for Shared Resources Management

- **Software approach implemented in a hypervisor**
  - One virtual machine per core hosting an Operating System
  - Shared resources management policy hidden to guest software

- **Resource sharing policy setup and configuration**
  - Interconnect bandwidth quota allocated to each core
  - On the fly control of accesses to cope with the allocated quota
  - Detection of pathological situations

Which impact on application’s performances?
Context

Problem Statement

Approach for Shared Resources Management

Prototyping

Conclusion
◆ Objectives
  ▪ Determine a usage domain for applications
  ▪ Determine configurations for which shared resources access times are independent

◆ Experimentation
  ▪ Quad-core processor from the QorIQ series
  ▪ Focus on DDR accesses regulation

◆ Virtual machines scenarios
  ▪ One benchmark VM
  ▪ Up to three stress VM
Phase 1: Estimate raw performances of applications: AES and FFT

- Simulation of Average Memory Access Time
- Comparison of execution time with a reference execution in a bare metal configuration

There is a usage domain in which our solution seems efficient.
Phase 2: Estimate performances degradation of bandwidth sharing

- Allocation of a portion of interconnect bandwidth
- Comparison of execution time with a reference execution inside a virtual machine that is granted all interconnect bandwidth

Measurements for AES

Measurements for FFT

Bandwidth sharing entails a limited degradation of performances
Phase 3: Assess bandwidth management impact on worst case access times to DDR

- The benchmark VM is granted a portion of interconnect bandwidth
- Stress VM share another portion of the interconnect bandwidth
- Access times from benchmark VM to DDR are collected

Access profile for the reference execution
Phase 3: Reference execution, no traffic regulation

- No stress Cores
- 1 stress Cores
- 2 stress Cores
- 3 stress Cores
Phase 3: 50% of interconnect bandwidth is allocated to each group.
Phase 3: 30% of interconnect bandwidth is allocated to each group.
Context

Problem Statement

Approach for Shared Resources Management

Prototyping

Conclusion
Conclusion

- Computing modularly WCET of several applications on a COTS multicore under AMP deployment is an open problem
  - Gathering WCAT to shared resources is the root problem
  - This problem was partially closed by removing some constraints
    - Deterministic home-maid processors
    - Non modular WCET analysis

- We introduced a purely software solution
  - Configurable resources sharing policy, not visible by guest software
  - Limited impact on application performances within a usage domain

- Future prototype improvements
  - Main effort on shared resources management capabilities
  - Deeper experiment with several avionic applications
Thanks for your attention!