Multicore Migration Study in Automotive Powertrain Domain

21/Apr/2017
Hitachi, Ltd.,
Takeshi Fukuda, Tasuku Ishigooka, Fumio Narisawa
Hitachi Automotive Systems, Ltd.,
Toru Irie, Takafumi Suzuki

2nd Workshop on Certifiable Multicore Avionics and Automotive Systems
Contents

1. Outline of Hitachi
2. Background
3. Control Software and Issue
4. Our Method
5. Evaluation with Engine Control Software
6. Summary and Discussion
## Hitachi corporate data

### Hitachi, Ltd.

**Incorporated**  1st February 1920 [Founded 1910]

**Head Office**  Marunouchi 1-6-6, Chiyoda-ku, Tokyo, JAPAN

### Hitachi Group (Consolidated)  FY2015

<table>
<thead>
<tr>
<th>Category</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revenues</td>
<td>JPY 10,343 billion</td>
</tr>
<tr>
<td>R&amp;D expenditure</td>
<td>JPY 333.7 billion [3.3% of revenue]</td>
</tr>
<tr>
<td>No. of subsidiaries</td>
<td>1,056 companies [262 domestic; 794 overseas]</td>
</tr>
<tr>
<td>No. of employees</td>
<td>335,244 [187,936 domestic; 147,308 overseas]</td>
</tr>
</tbody>
</table>
Revenues (FY2015)
10,343 billion (JPY)

- Smart Life & Ecofriendly Systems: 9%
- Information & Telecommunication Systems: 21%
- Social Infrastructure & Industrial Systems: 21%
- Electronic Systems & Equipment: 19%
- Financial Services: 3%
- High Functional Materials & Components: 7%
- Construction Machinery: 14%
- Others (Logistics and Other services): 6%
- Automotive Systems: 11%
System solutions that Hitachi Automotive Systems will provide for progress in “Environment”, ”Safety” and ”Information” field.

- **Environment field**
  - Engine management systems
    - Control system
    - Fuel system
    - Ignition system
    - Air intake system
    - Engine components and sub-systems
    - Exhaust system

- **Safety field**
  - Electric powertrain systems
    - Hybrid electric car/electric car system
    - Electrical equipment system
  - Drive control systems
    - Outside recognition drive system
    - Steering system
    - Brake system
    - Suspension system
    - Drive power transmission system
    - Other control systems

- **Information field**
  - Car information systems
    - Cloud information network services
    - On-board information equipment
Offering environment-friendly and high-efficiency engine management systems for reduction in CO2 and prevention of air pollution.

[Diagram of engine components with labels:
- ECU: Engine control unit
- High-pressure fuel pump
- Electronic throttle body
- Airflow sensor
- Ignition coil (Hitachi Automotive Systems Hanshin)
- Transmission control unit
- Injector
- Variable Valve Event and Lift Control System
- Valve Timing Control
- Oil pump
- Balancer
- Piston
- Water pump]
Contents

1. Outline of Hitachi
2. Background
3. Control Software and Issue
4. Our Method
5. Evaluation with Engine Control Software
6. Summary and Discussion
The performance requirements of automotive engine control are increasing, for instance to comply the exhaust emission regulations and reduce gasoline consumption.
Current electronic controller unit requests high performance microcontroller.

Trend of Microcontroller for Powertrain Systems

- High performance trend

Clock [MHz]

Performance

['10', '15', '20'] [Year]

Single Core  Core * 2  Core * 3 -
Our study: To migrate legacy source code to multicore platform.

Comparison with other products

<table>
<thead>
<tr>
<th>PC, Smart phone</th>
<th>Television</th>
<th>Automotive engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Dependency</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Looseley coupled</td>
<td>Looseley coupled</td>
<td>Complicated</td>
</tr>
<tr>
<td>Mailer</td>
<td>Text editor</td>
<td>Web browser</td>
</tr>
</tbody>
</table>
| Core1 | Core2 | Core3 | Core1 | Core2 | | Core1? | Core2?

1. Task + Execution order → Dataflow
Our study

Comparison

Data Dependency

1. Task + Execution order → Dataflow

A portion of dataflow of engine control software

[Cite]
Contents

1. Outline of Hitachi
2. Background
3. Control Software and Issue
4. Our Method
5. Evaluation with Engine Control Software
6. Summary and Discussion
Dataflow of control software roughly consists of sensing, calculating, and actuating.

E.g. Dataflow of fuel injection

(1) Sensing airflow
(2) Calculating air volume
(3) Fitting fuel volume
(4) Calculating timing of fuel injection

Notes
- Task in AD convertor
- Task in microcontroller
- Dataflow
Tasks are executed periodically with each cyclic time or executed sporadically based on event like an engine rotation speed.

Time chart (single core processor)

- **AD convertor (1)250μ**: Task 1
- **(2)1ms task**: Task 2
- **(3)2ms task**: Task 3
- **(4)Interrupt based on rpm**: Task 4

Wait time by fixed priority scheduling
Issue of parallelization

A large number of inter-core communication data causes high frequent inter-core synchronization between cores.
Issue: Long wait time is frequently happened when one core is interrupted by other high priority tasks due to frequent inter-core synchronization.
Issue: Long wait time is frequently happened when one core is interrupted by other high priority tasks due to frequent inter-core synchronization.

【Our Goal】
In order to achieve high parallelization, our goal is to reduce a number of inter-core synchronization.
Contents

1. Outline of Hitachi
2. Background
3. Control Software and Issue
4. Our Method
5. Evaluation with Engine Control Software
6. Summary and Discussion
We proposed parallelization method with performance requirements of system control. Feature of our method is to identify the inter-core data which doesn’t need inter-core synchronization according to the requirements of “Data Delay Time”.

(1) Program analysis
(2) Detecting inter-core data
(3) Categorizing data

Control SW for single core → Dataflow → Strategy of core allocation → Requests for data delay → Data

Data with synchronization → Data without synchronization
Data delay time is one kind of end-to-end path latency, especially, from start time of sensing task to start time of other task which has data dependence with sensing task.

**With synchronization between cores**

---

**Data delay time**

**Time chart 1**

Data delay is shown in the chart. The data delay is marked with an arrow and the timing is indicated with a box. The chart compares the data delay at two different time points, T+1 and T+2, with the synchronization between cores marked with a dotted line.

**AD convertor (1)250μ**

**Core 1** (2)1ms task

**Core 2** (3)2ms task

**Wait time**

The chart shows the timing differences and the synchronization points clearly.

---

CMAAS 2017
Data delay time (cont’d)

Without synchronization between cores

Time chart 2

AD convertor (1) 250μ

Core 1 (2) 1ms task

Core 2 (3) 2ms task

No wait time

[ms]
Our idea: we select the parallelization option which is without synchronization between cores when we can allow a WCRT (Worst Case Response Time) of data delay time.

<table>
<thead>
<tr>
<th></th>
<th>With synchronization</th>
<th>Without synchronization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input data</td>
<td>Same timing data</td>
<td>Previous task data</td>
</tr>
<tr>
<td>Wait time</td>
<td>Long 😞</td>
<td>Zero 😊</td>
</tr>
<tr>
<td>Parallelization</td>
<td>Low 😞</td>
<td>High 😊</td>
</tr>
<tr>
<td>Data delay time</td>
<td>Short 😊</td>
<td>Long 😞</td>
</tr>
<tr>
<td>WCRT of data delay time</td>
<td>Short 😊</td>
<td>Long 😞</td>
</tr>
</tbody>
</table>

\[
\sum_{i=1}^{N-1} (R(\tau_i) + \hat{n}T(\tau_{i+1}) - r(\tau_{i+1}) + S(\tau_N))
\]

[Cite]
Contents

1. Outline of Hitachi
2. Background
3. Control Software and Issue
4. Our Method
5. Evaluation with Engine Control Software
6. Summary and Discussion
Overview of evaluation

We migrated whole legacy engine control software to multicore ECU which has two cores. And evaluated it with HILS.

Core 1: Cyclic tasks
(E. g. 1ms, 2ms, 10ms tasks)

- Periodic Task
- API
- Periodic BSW
- Core 1

Core 2: Sporadic tasks
(regarding engine rotation speed)

- Crank Event Task
- API
- OS
- Inter-Core COM
- OS
- Event (rpm)
- Core 2

【Engine control SW】 included approx. 10000 control signals
【ECU】 two cores

ECU (Engine Controller Unit) (Hardware-in-the-loop Simulation)
Total amount of Inter-core communication data

We found that a total amount of inter-core communication data is approx. 600 in our process step two.
Our method indicated that more than 90% out of approx. 600 inter-core communication data don’t need synchronization.
Result of HILS evaluation indicated that our method is
(1) useful for parallelizing engine control software
(2)
(3)
Evaluation with HILS (Hardware-in-the-loop Simulator)

Result of HILS evaluation indicated that our method is

1) useful for parallelizing engine control software
2) able to distribute CPU load to cores
3)
Result of HILS evaluation indicated that our method is
(1) useful for parallelizing engine control software
(2) able to distribute CPU load to cores
(3) Keeping hard real-time deadline. e.g. fuel infection
Contents

1. Outline of Hitachi
2. Background
3. Control Software and Issue
4. Our Method
5. Evaluation with Engine Control Software
6. Summary and Discussion
Summary

【Proposed method】
・Parallelization method for control software
・The feature of our method is
  - identify the inter-core data which doesn’t need core synchronization according to the requirement of “Data Delay Time”

【Evaluation】
・We apply our method to whole legacy engine control software.

【Results】
・More than 90% of data out of approximately 600 inter-core communication data don’t need synchronization mechanism.

・The evaluation results with HILS indicated that the parallelized software satisfied requirements of real-time performance.
【Discussion】

- Process and Method to decide requirement time of data delay with control engineer is required.

It is tough issue to decide it which doesn’t cause a bad effect for the performance of systems.

It is a nonsense if multicore migration decrease engine performance like fuel efficiency.

- Precise WCRT analysis method is required.

The result of WCRT analysis for data delay time is too pessimistic.
Thank you!

HITACHI
Inspire the Next