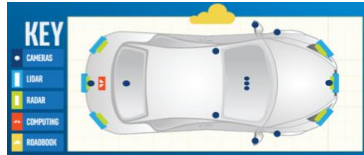
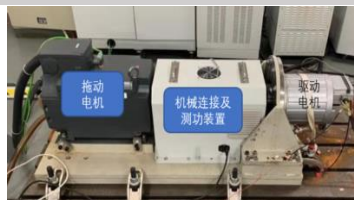
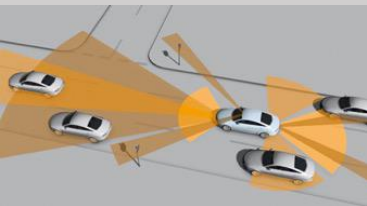


AUTOSAR多核操作系统监控与优化

同济大学—维克多汽车技术联合实验室



1

实验室简介

2

AUTOSAR多核OS发展与挑战

3

AUTOSAR多核OS监控与优化

3.1 基于T1的AUTOSAR多核OS监控

3.2 基于TA的AUTOSAR多核OS建模与优化

3.3 自主开发的AUTOSAR多核OS监控与优化

4

总结

1.1 合作伙伴



联合实验室签约仪式

其他合作伙伴



英飞凌培训

- ✓ AURIX Level1和Level2培训
- ✓ AURIX 2G培训
- ✓ AUTOSAR MCAL配置培训



同济大学
英飞凌日

联合研究

- ✓ 基于Vector VT system联合开发HIL台架
- ✓ 自动驾驶中传感器融合与虚拟化测试
- ✓ 共同撰写并出版AUTOSAR领域的书籍

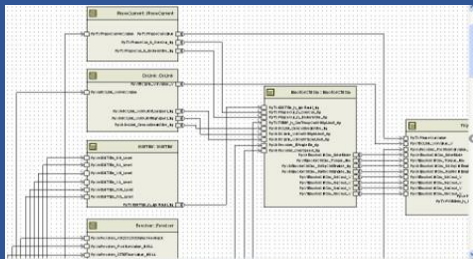


同济大学
维克多
联合研究



1.2 国家重点研发项目

符合ASIL-C等级的高功率密度电机控制器关键技术研究



开发AUTOSAR架构下符合ASIL-C功能安全等级的软件

VECTOR **Press Release**
July 29, 2016
1 / 5

The First Prototype of E-Motor based on AUTOSAR in China has been Successfully Developed

- Collaboration among Tongji University, Dajun, Infineon and Vector

Shanghai, CHINA, 2016-07-29 – Sino-German school for postgraduate studies of Tongji University (CDHK), Shanghai Dajun Technologies, Inc., Infineon Technologies and Vector Automotive Technology announced today that CDHK have developed the prototype of motor controller on the basis of Infineon Aurix TC275 microcontroller and Vector AUTOSAR 4.0.3 with multicore OS. This is the first time in China that Aurix MCU has been used to develop motor system based on AUTOSAR.

首个基于AUTOSAR的E-Motor原型机开发

发明专利

Technical Report

No. 18-00031-CER-SHA-00

On the Software module "Torque estimation"

Customer: CHINESECH-DEUTSCHES HOCHSCHULKOLLEG of Tongji University

Revision 1.0
Dated 07th Sep 2018

Test Body: TÜV SÜD China

软件代码被TÜV南德评定为符合ASIL-C功能安全等级

陆科博士被Exida认证为功能安全专家

Certificate / Zertifikat / 认证
CFSE 200609 001

exida Certification hereby confirms that:

Ke Lu

Has met the program requirements as defined by the CFSE Advisory Board in the field of functional safety applied to:

Automotive Application

And is authorized to use the title:

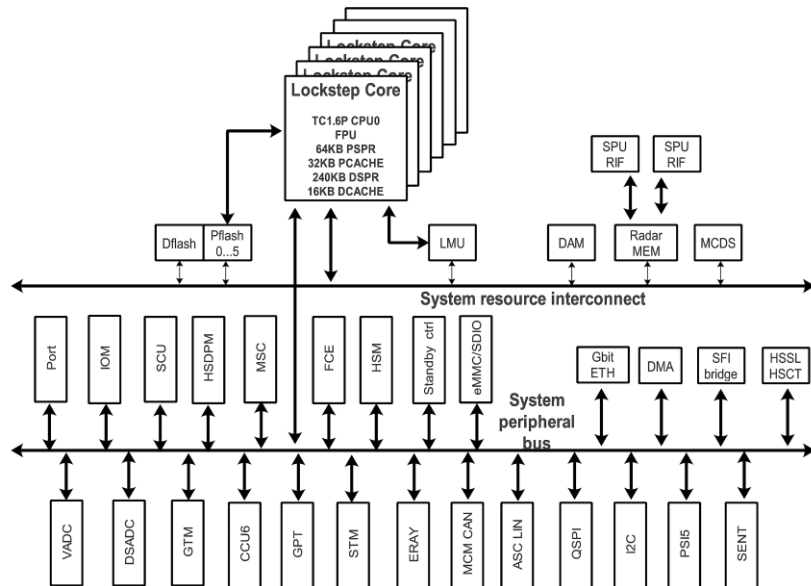
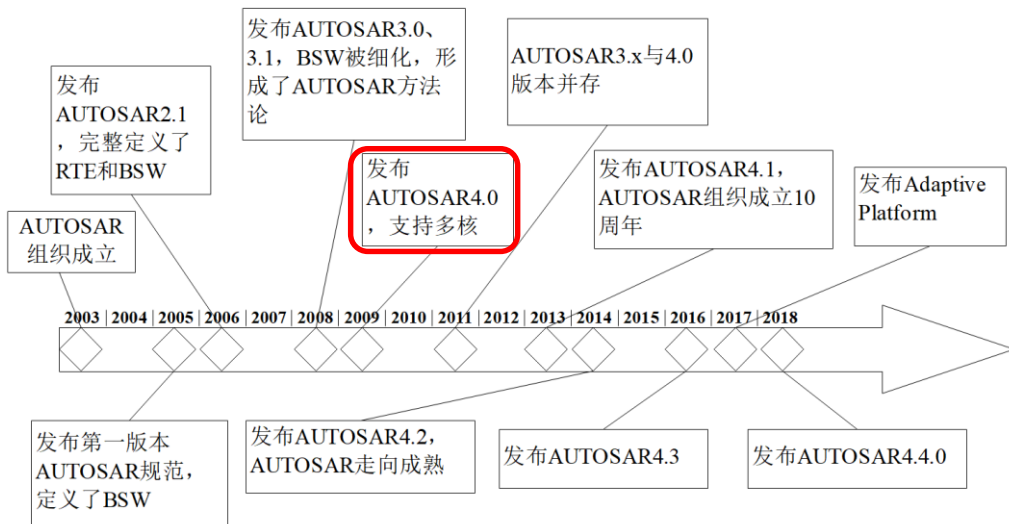
Certified Functional Safety Expert

And passed the exam for the standard: **ISO 26262**

A/CFSE has clear obligations and duties to exercise his or her skills toward the creation, promotion and maintenance of functional safety.

Exida Certification, Columbia, TN, USA
Director: *William M. Wolfe*

2.1 AUTOSAR多核OS发展



正确性

■ 避免

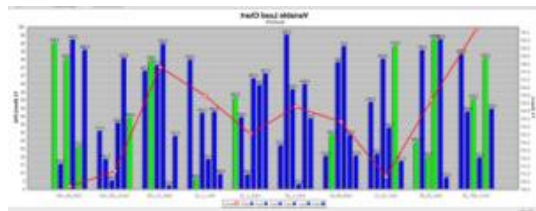
- 死锁 (deadlocks)
- 竞争条件 (race conditions)
- 饥饿 (starvation)
- 优先级反转 (priority inversion)



■ 保证数据一致性

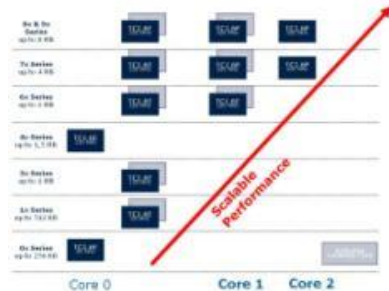
性能要求

- 每个核都要满足可调度性约束
- 降低核间数据交互损耗
- 均衡多核负载率



可扩展性

- 多核操作系统不仅能支持一个多核处理器，而且支持一系列不同核数的处理器，具有可扩展性从而控制软件开发成本。



2.2.1 数据一致性

两个任务异常读取数据

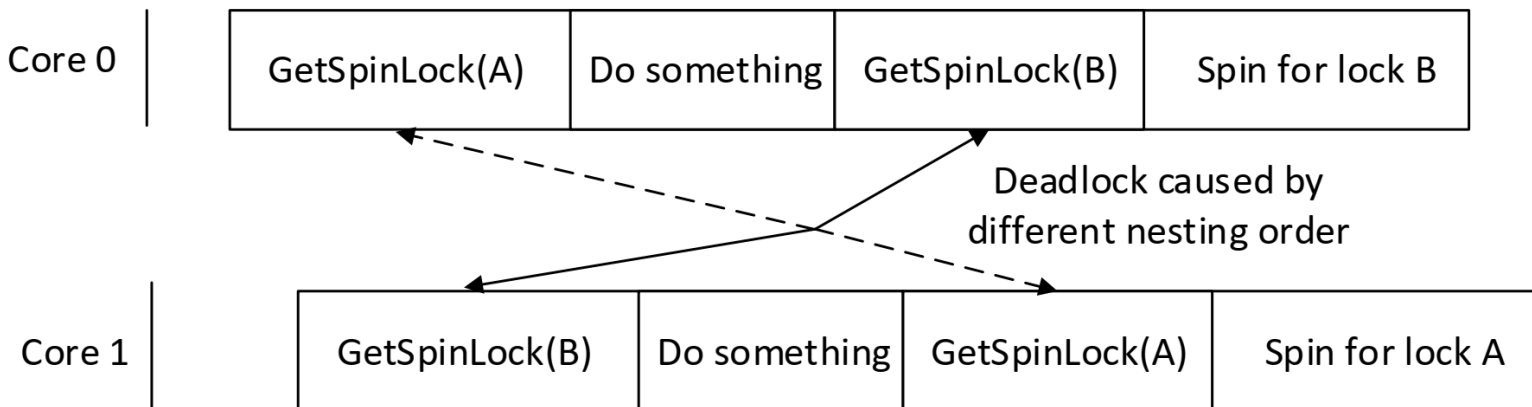
| Thread 1 | Thread 2 | | Integer value |
|----------------|----------------|---|---------------|
| | | | 0 |
| Read value | | ← | 0 |
| | Read value | ← | 0 |
| Increase value | | | 0 |
| | Increase value | | 0 |
| Write back | | → | 1 |
| | Write back | → | 1 |

两个任务正常读取数据

| Thread 1 | Thread 2 | | Integer value |
|----------------|----------------|---|---------------|
| | | | 0 |
| Read value | | ← | 0 |
| Increase value | | | 0 |
| Write back | | → | 1 |
| | Read value | ← | 1 |
| | Increase value | | 1 |
| | Write back | → | 2 |

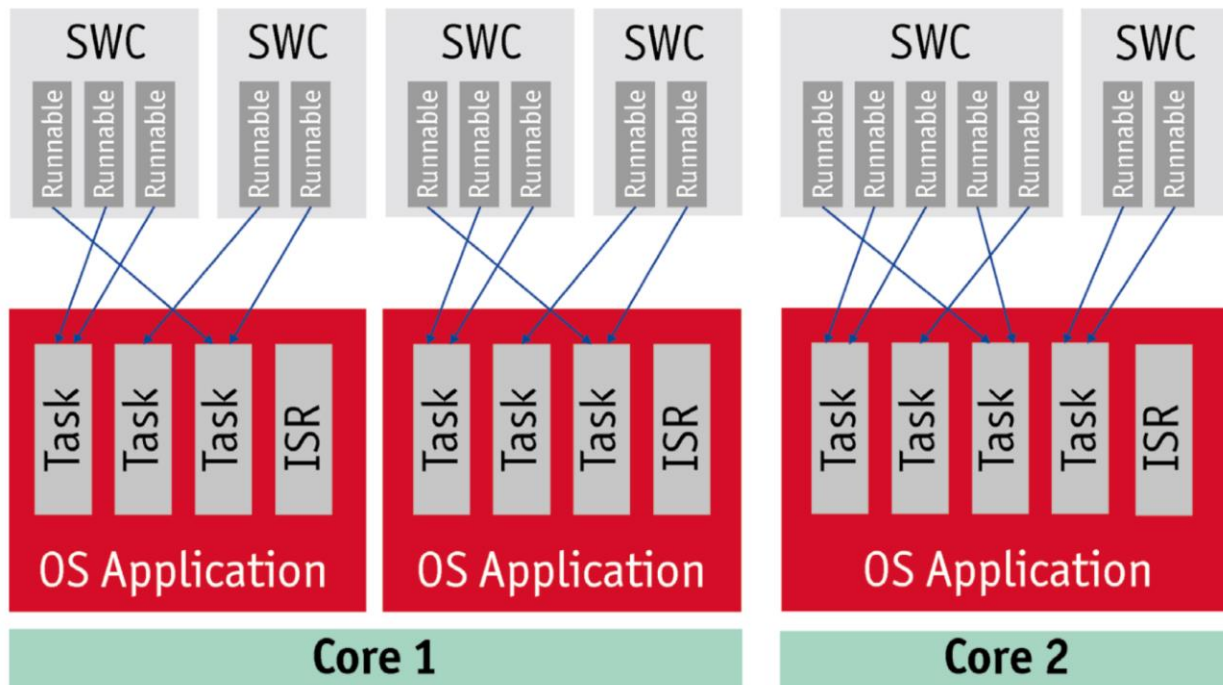
2.2.1 数据一致性

自旋锁Spinlock



死锁是由两个不同内核上的TASK以不同顺序获取的两个自旋锁引起的。

2.2.2 任务分配部署



正确性问题

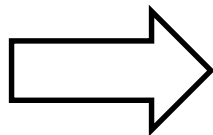
死锁

系统运行异常

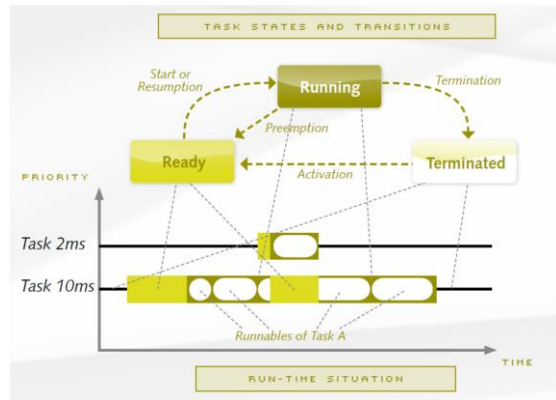
数据一致性

数据交互损耗

...



多核OS实时监控

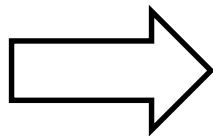


性能要求

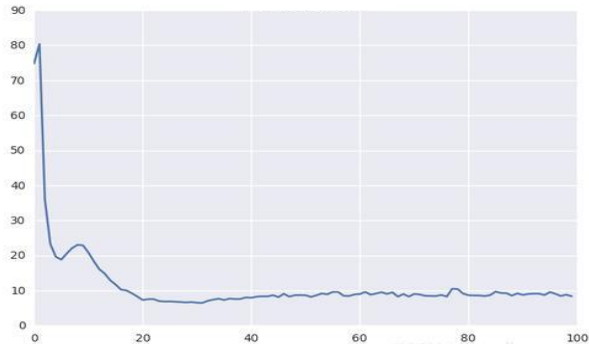
负载均衡

实时性要求

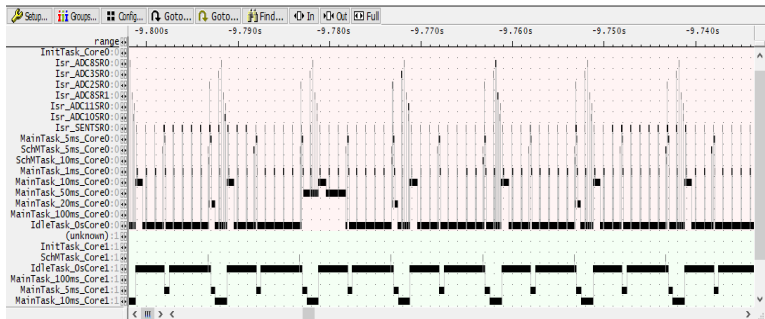
...



多核OS建模优化

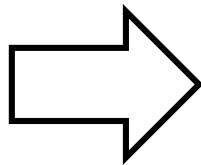


硬件调试器方案

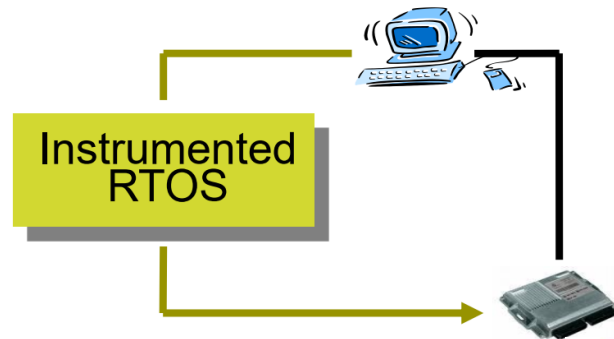


缺点:

- ✗ 应用成本高
- ✗ 监控对象需要特殊芯片 (例如ED芯片)
- ✗ 使用场景局限



软件监控方案

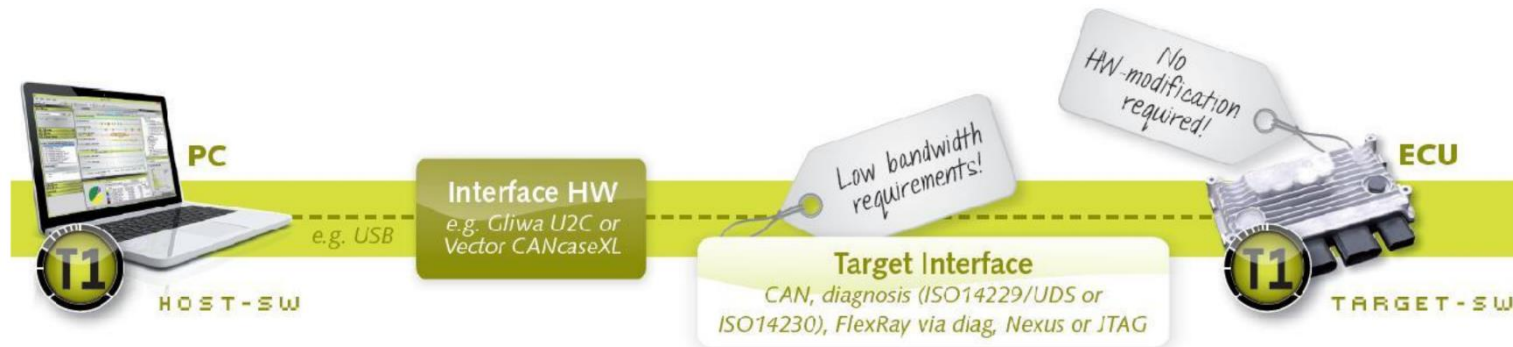


优点:

- ✓ 可移植性强, 成本低
- ✓ 无硬件限制
- ✓ 使用场景广泛
- ✓ 可分析多种时序参数



3.1 基于T1的AUTOSAR多核OS监控

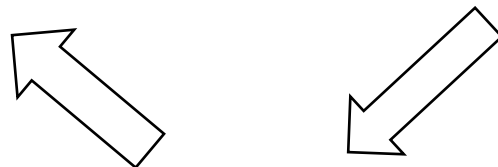


T1-HOST-SW上位机 (软件)

- 运行于PC之上
- 显示测量结果
- 分析测量结果
- 控制/配置T1

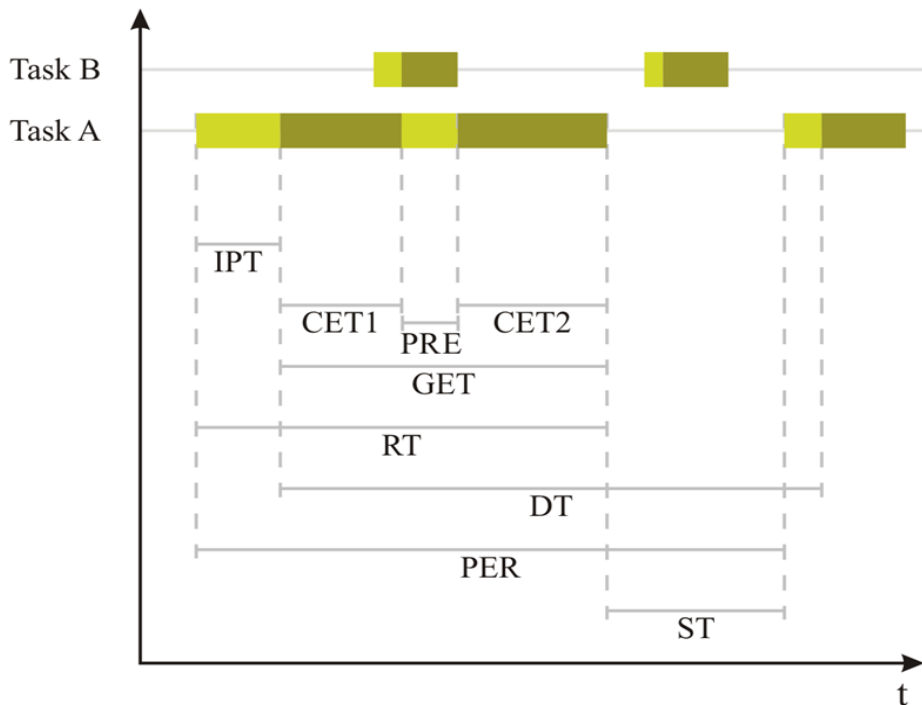
T1-TARGET-SW目标 (软件)

- 运行于 (嵌入式) 目标系统内
- 测量/追踪
- 监控
- 分析结果 (实时)



使用已有总线系统通讯
如CAN/ETH等

3.1 基于T1的AUTOSAR多核OS监控

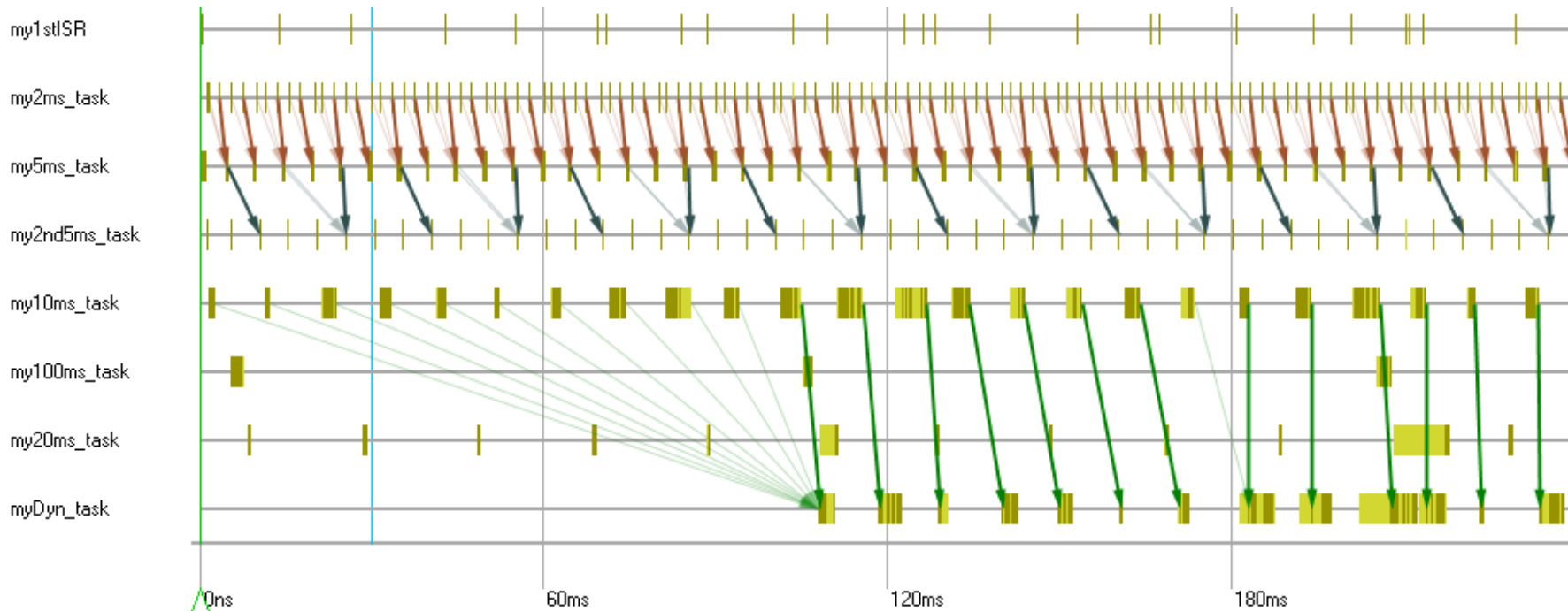


$$\text{CET} = \text{CET1} + \text{CET2}$$

| Abr. | Explanation (EN) |
|------|----------------------|
| IPT | initial pending time |
| CET | core execution time |
| GET | gross execution time |
| RT | response time |
| DT | delta time |
| PER | period |
| ST | slack time |
| PRE | preemption |
| JIT | jitter |
| CPU | cpu load |

3.1 基于T1的AUTOSAR多核OS监控

监控数据流



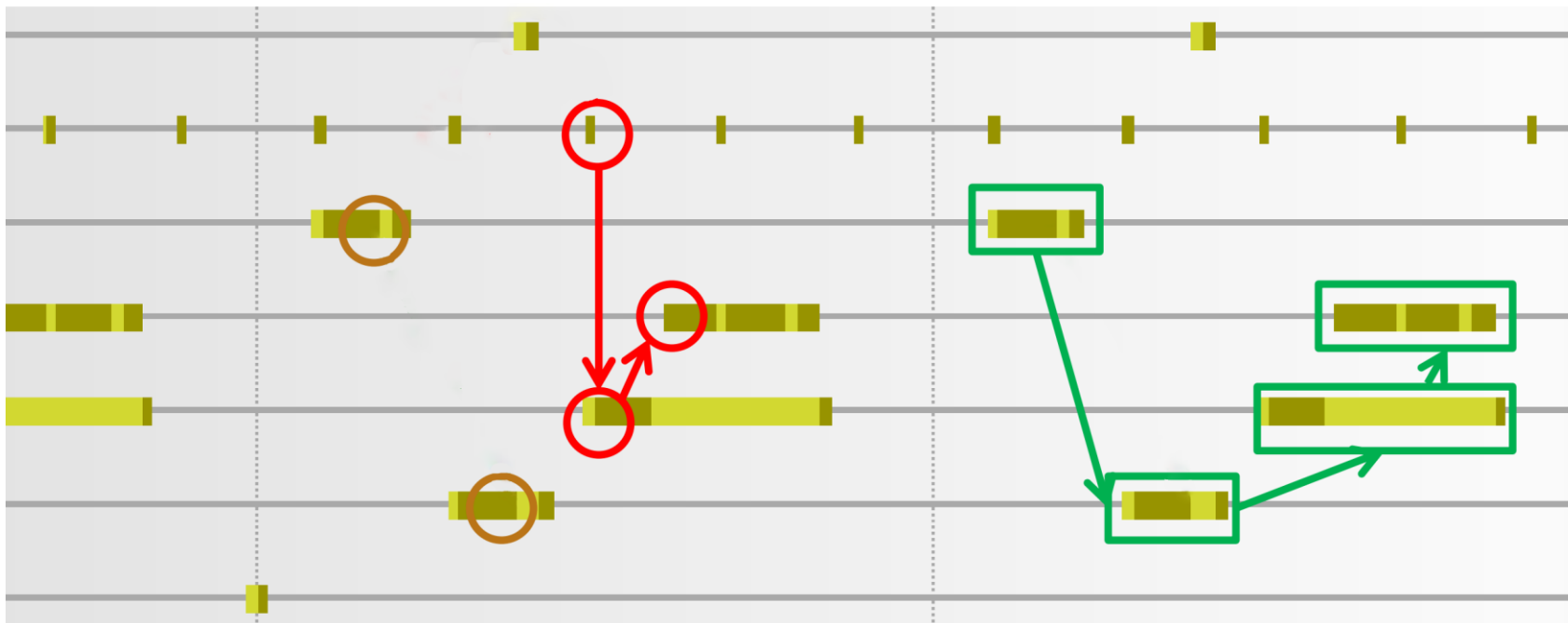
3.1 基于T1的AUTOSAR多核OS监控

监控运行的时序

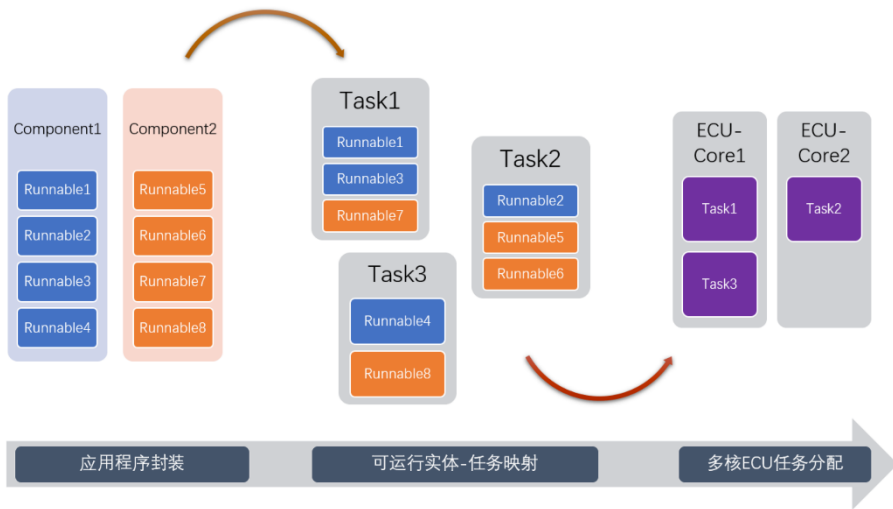
○ 事件 (Event) 发生

○ 事件链 (Event Chain) 发生

□ 系统中执行的任務



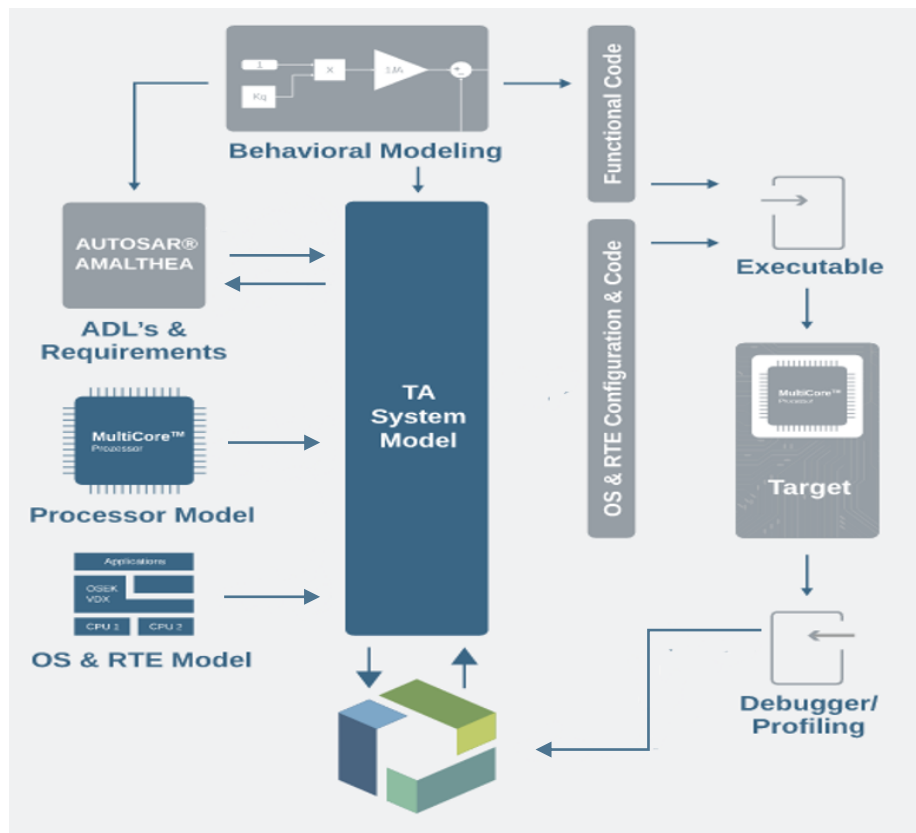
任务分配部署：手动分配



任务分配建议

| | |
|-----|--|
| 建议1 | 对于关键任务应分配较高的优先级，关键性相同的任务周期越短优先级越高。 |
| 建议2 | 耦合性较强的任务集应被分配至同一核中，以减少核间任务阻塞时间。 |
| 建议3 | 任务对自旋锁（Spinlock）的访问应尽量避免嵌套和回环。 |
| 建议4 | 无顺序约束的任务可采用定时或事件的触发方式，有顺序约束的任务最好按顺序采用事件（Set event）触发下一个任务。 |
| 建议5 | 在重要代码执行前应采用资源保护机制，以免被高优先级任务打断。 |
| 建议6 | 多核多任务的静态分配策略应使CPU负载率更加均衡。 |

3.2 基于TA的AUTOSAR多核OS建模与优化



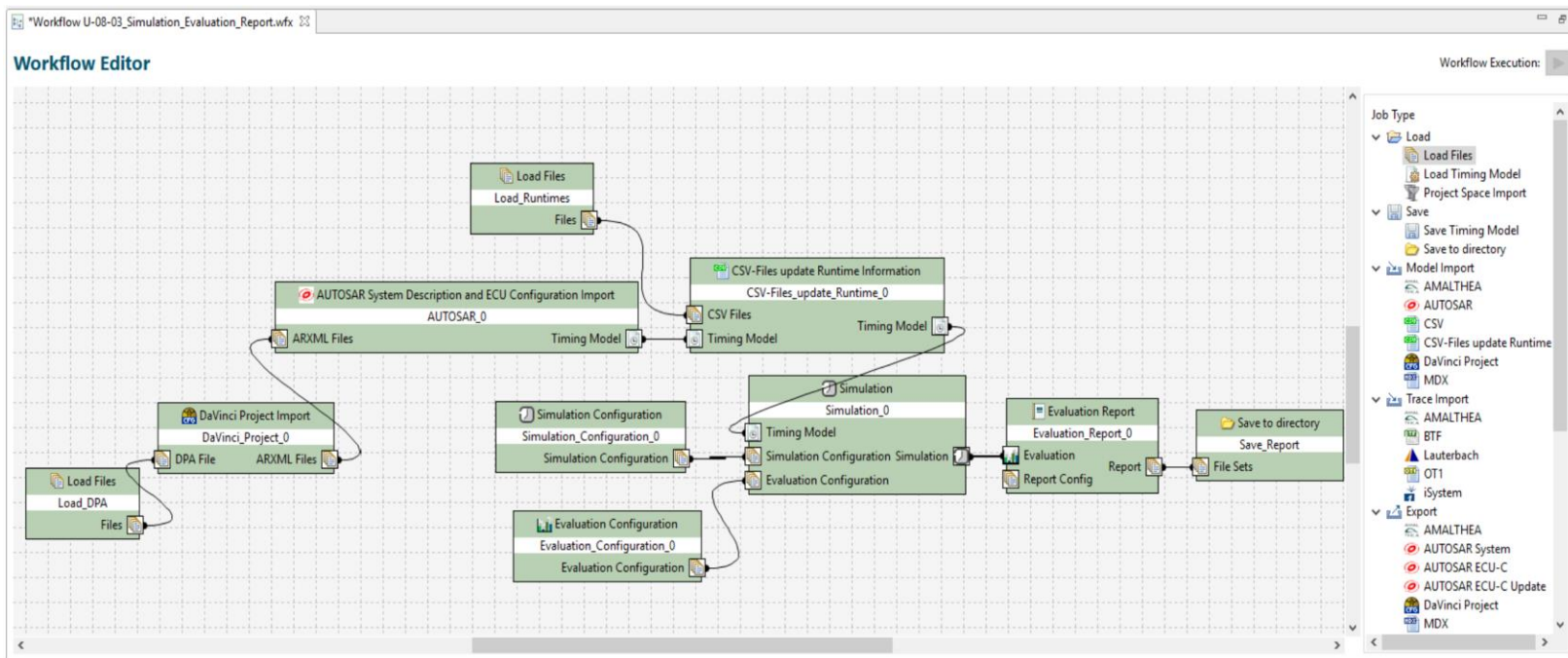
优化过程:

1. 硬件处理器建模
2. 输入多核OS和RTE配置文件
3. 输入单片机上运行的监控跟踪结果, 建立时序模型
4. TA建模仿真优化 (遗传算法)
5. 产生新的OS和RTE配置文件; 更新代码, 下载到单片机中运行

重复3-5过程, 直至满足性能要求

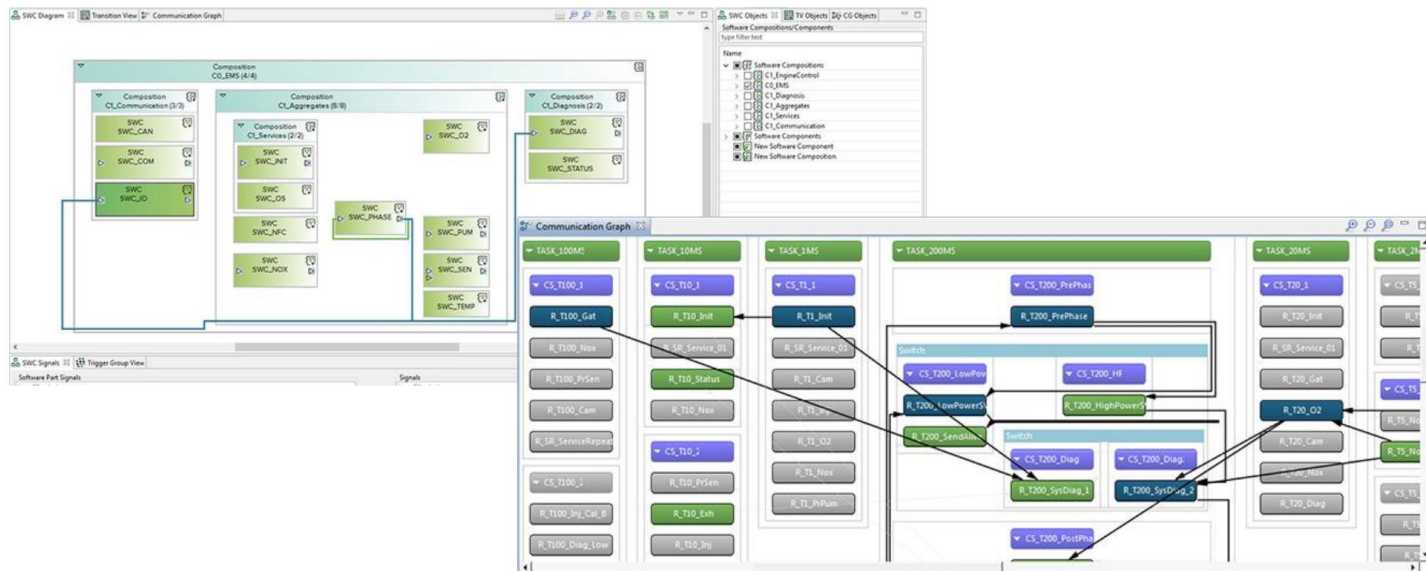
3.2.1 TA——Workflow

- 将建模过程模块化
- 提供与DaVinci工具的接口
- 支持命令行操作



3.2.2 TA——Design

- 应用层软件设计与静态架构分析
- 可视化功能组件和通信架构
- 定义和分析软件需求、事件链和软件约束



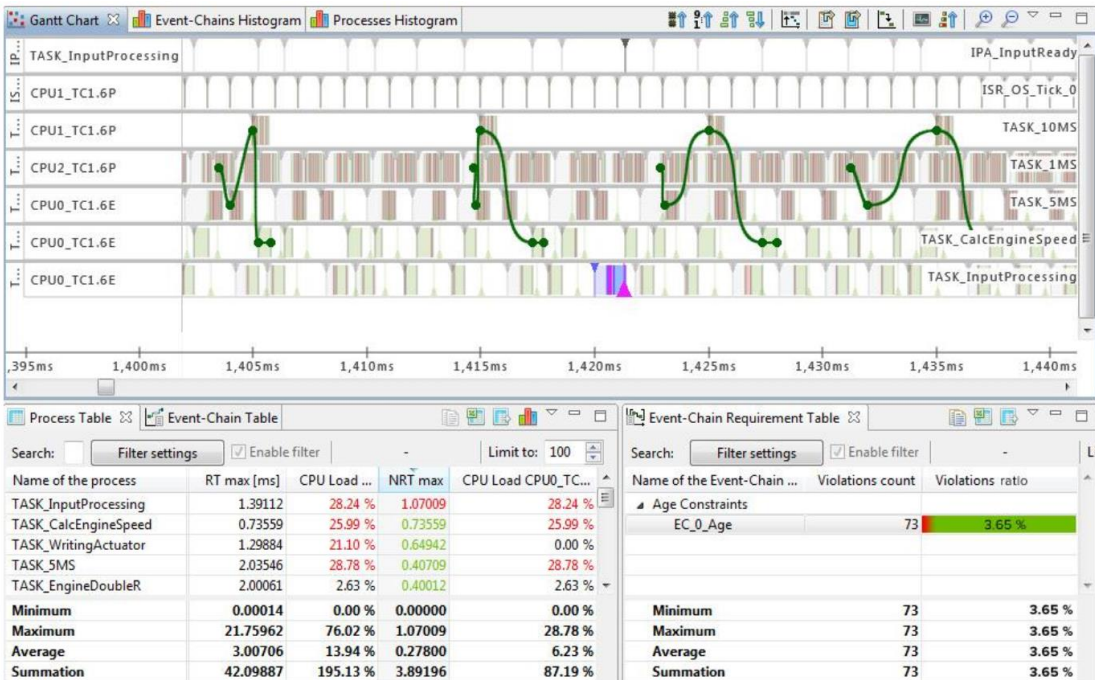
3.2.3 TA——Simulation

- 基于硬件仿真的可调度性分析
- 仿真验证软件系统是否满足实时性要求
- 深度仿真评估系统软硬件资源消耗情况



3.2.4 TA——Inspection

- 导入监控跟踪文件进行分析
- 直观对比监控文件与仿真结果
- 为时间模型添加时序参数



Import of traces format

- ▶ Best Trace Format (BFT)
- ▶ iSYSTEM Traces
- ▶ Lauterbach Traces
- ▶ PLS Traces
- ▶ GLIWA OT1

3.2.5 TA——Optimization

1、优化任务触发偏置:

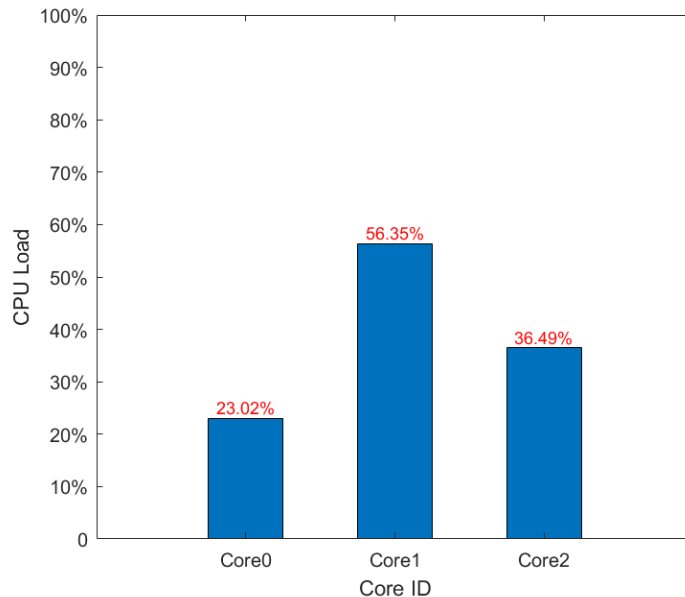
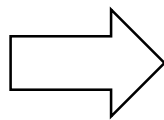
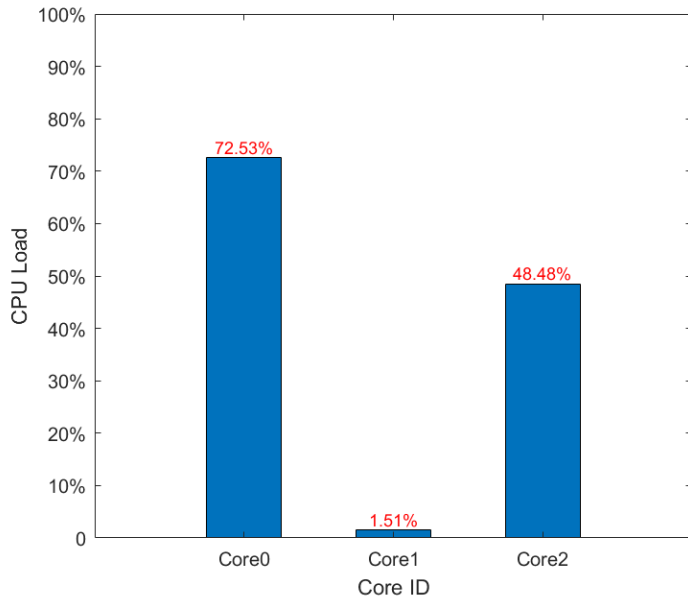
红色框为SchM_Task_C0，是一个5ms周期性Task，但其响应时间超出了5ms，不满足实时性要求。



3.2.5 TA——Optimization

2、优化任务分配:

更改Task到内核的映射关系，平衡CPU中各核的负载率，可以有效提高系统的可靠性和实时性。



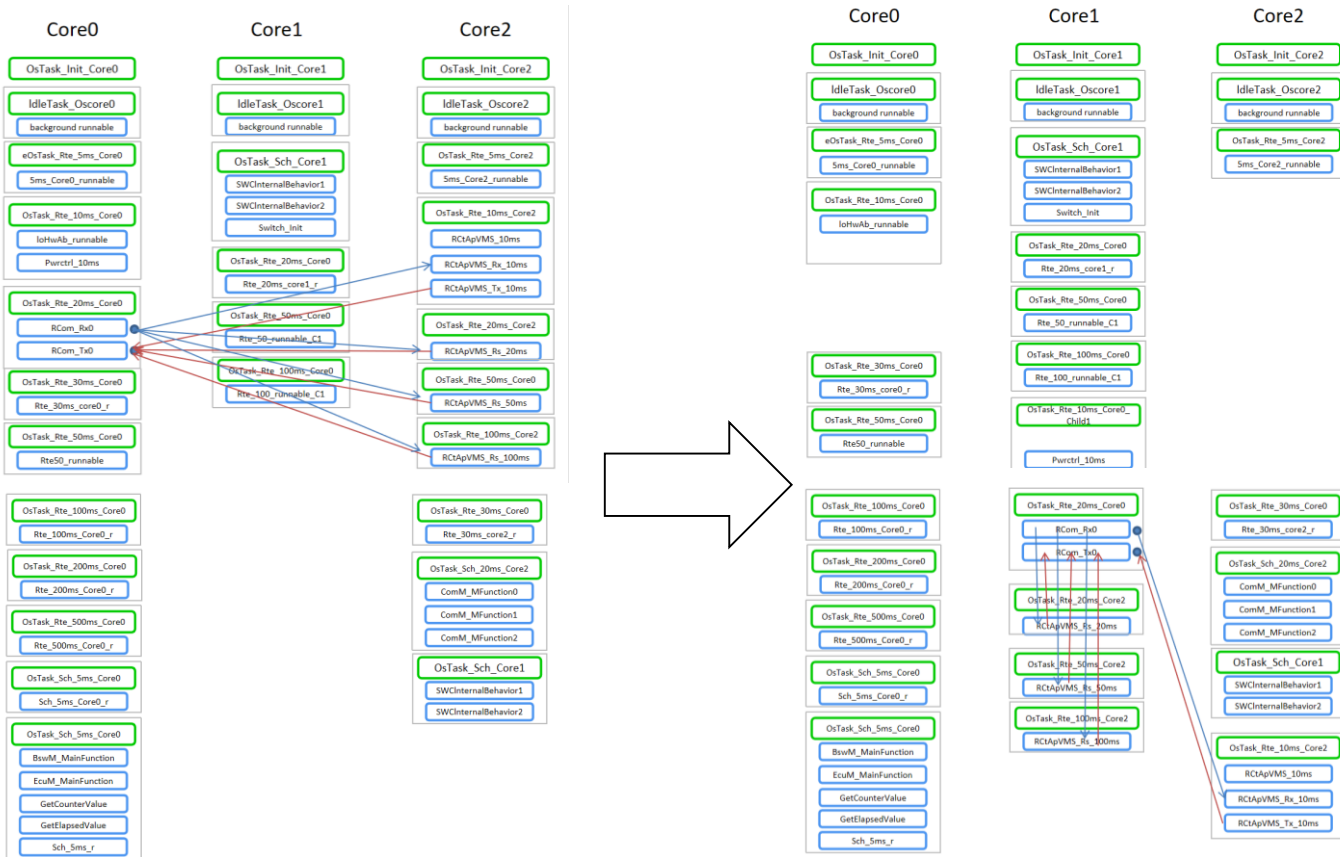
3.2.5 TA——Optimization

3、优化任务结构：

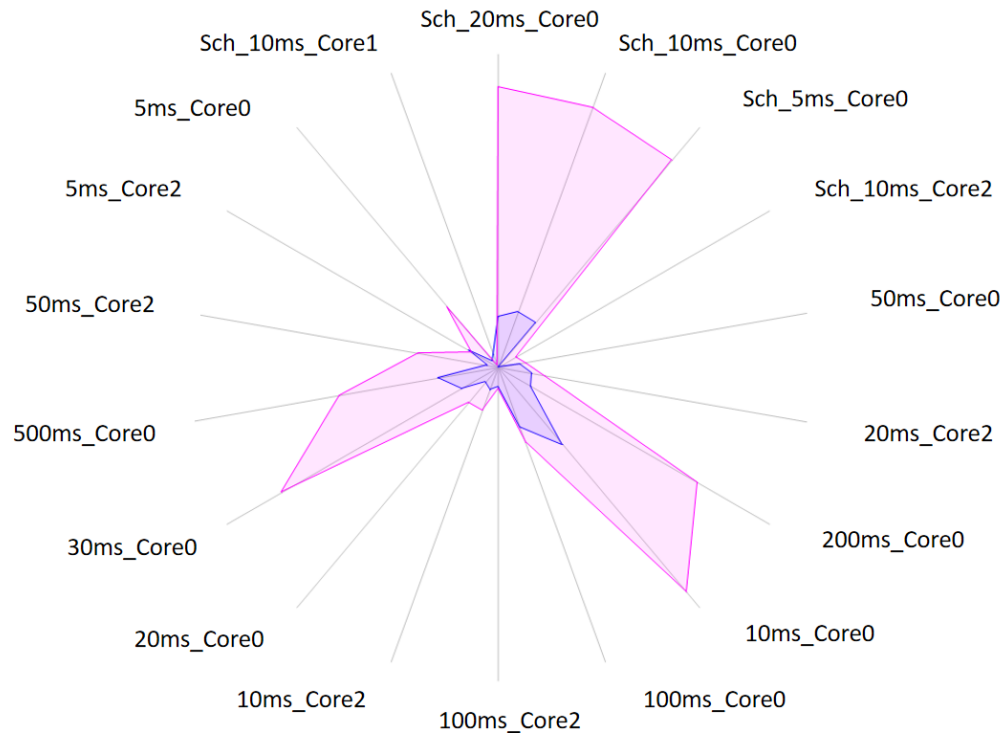
将OsTask_Rte_10ms_Core0拆分为两部分，移动其他几个Task到Core1中，既平衡了各核的负载率，又降低了核间通信。

| Task 名称 | 原程序 | | | | Solution_31 | | |
|------------------------|-------|-------|-------|-------------------------------|-------------|-------|-------|
| | Core0 | Core1 | Core2 | | Core0 | Core1 | Core2 |
| OsTask_Rte_10ms_Core0 | ● | | | OsTask_Rte_10ms_Core0 | ● | | |
| | | | | OsTask_Rte_10ms_Core0_Child_1 | | ● | |
| OsTask_Rte_20ms_Core0 | ● | | | OsTask_Rte_20ms_Core0 | | ● | |
| OsTask_Rte_20ms_Core2 | | | ● | OsTask_Rte_20ms_Core2 | | ● | |
| OsTask_Rte_50ms_Core2 | | | ● | OsTask_Rte_50ms_Core2 | | ● | |
| OsTask_Rte_100ms_Core2 | | | ● | OsTask_Rte_100ms_Core2 | | ● | |

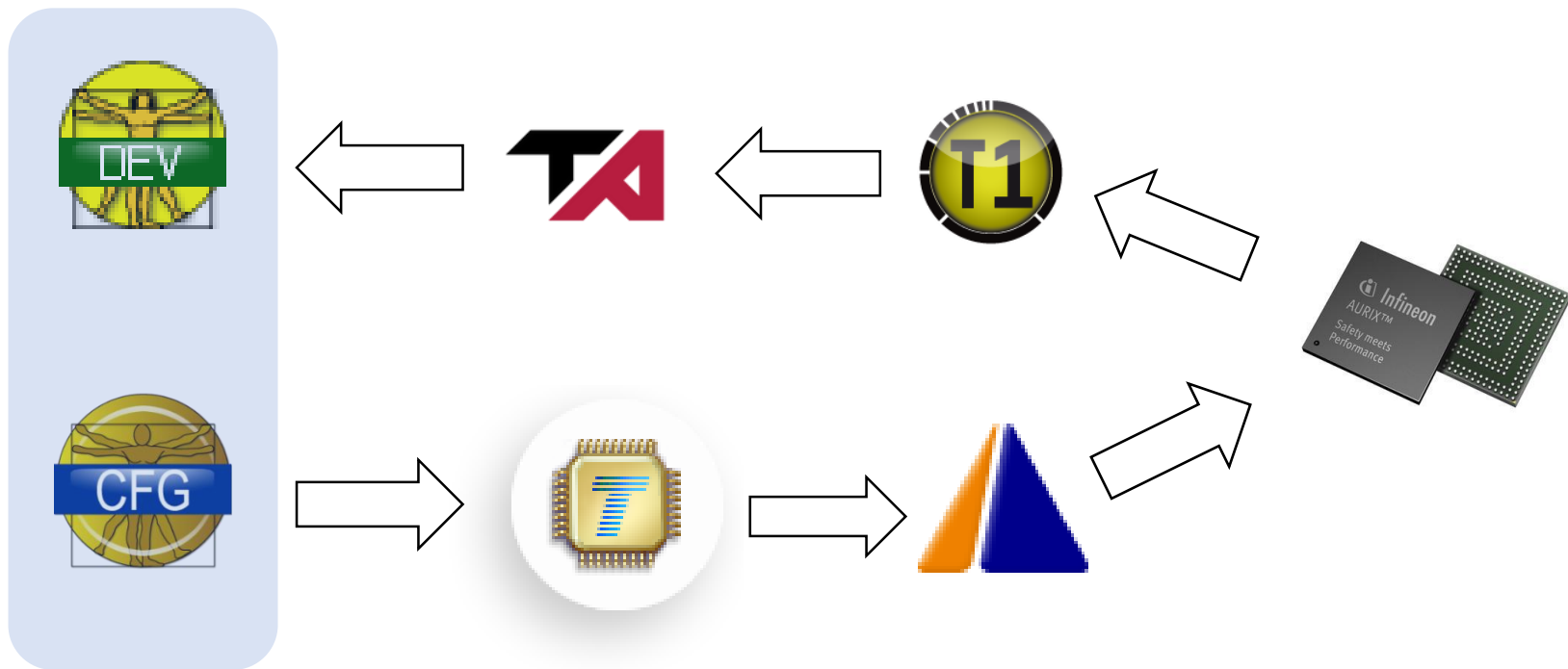
3.2.5 TA——Optimization



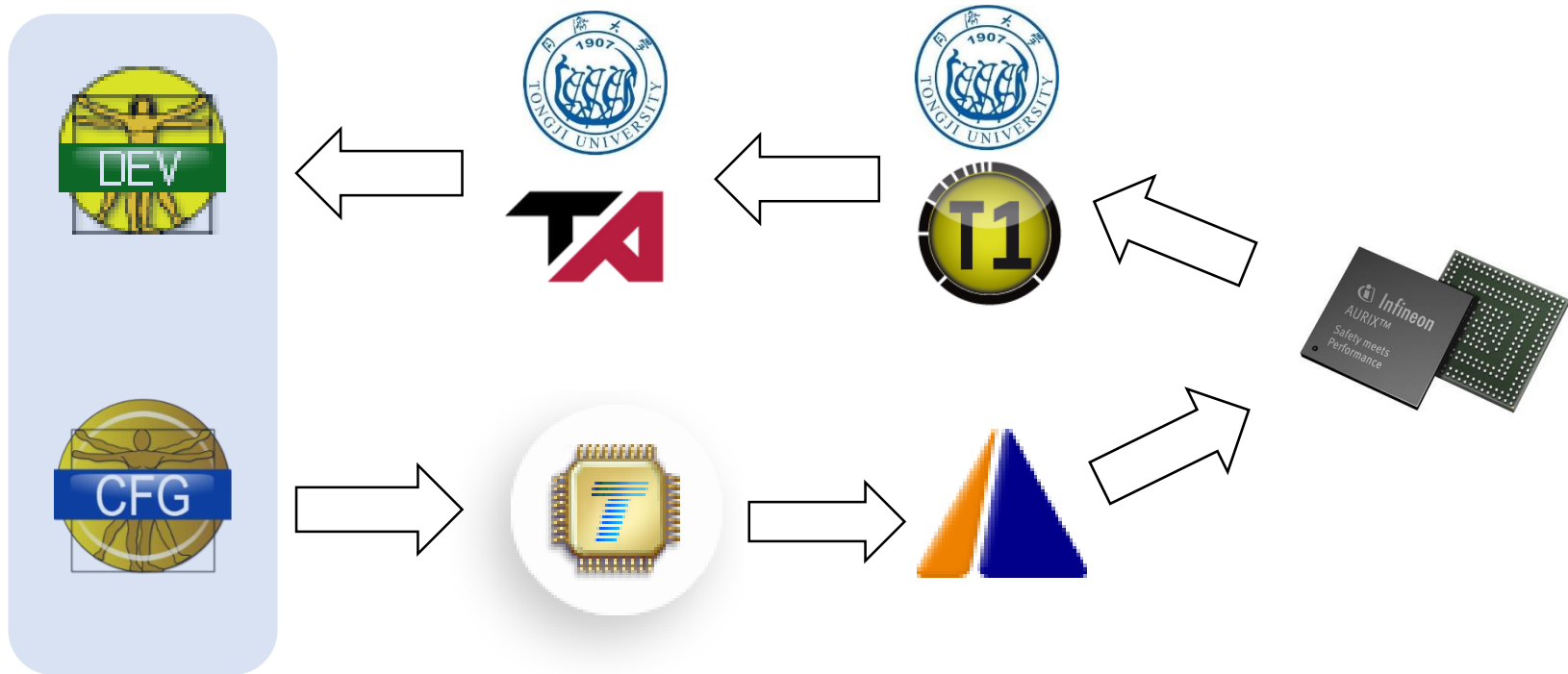
3.2.5 TA——Optimization



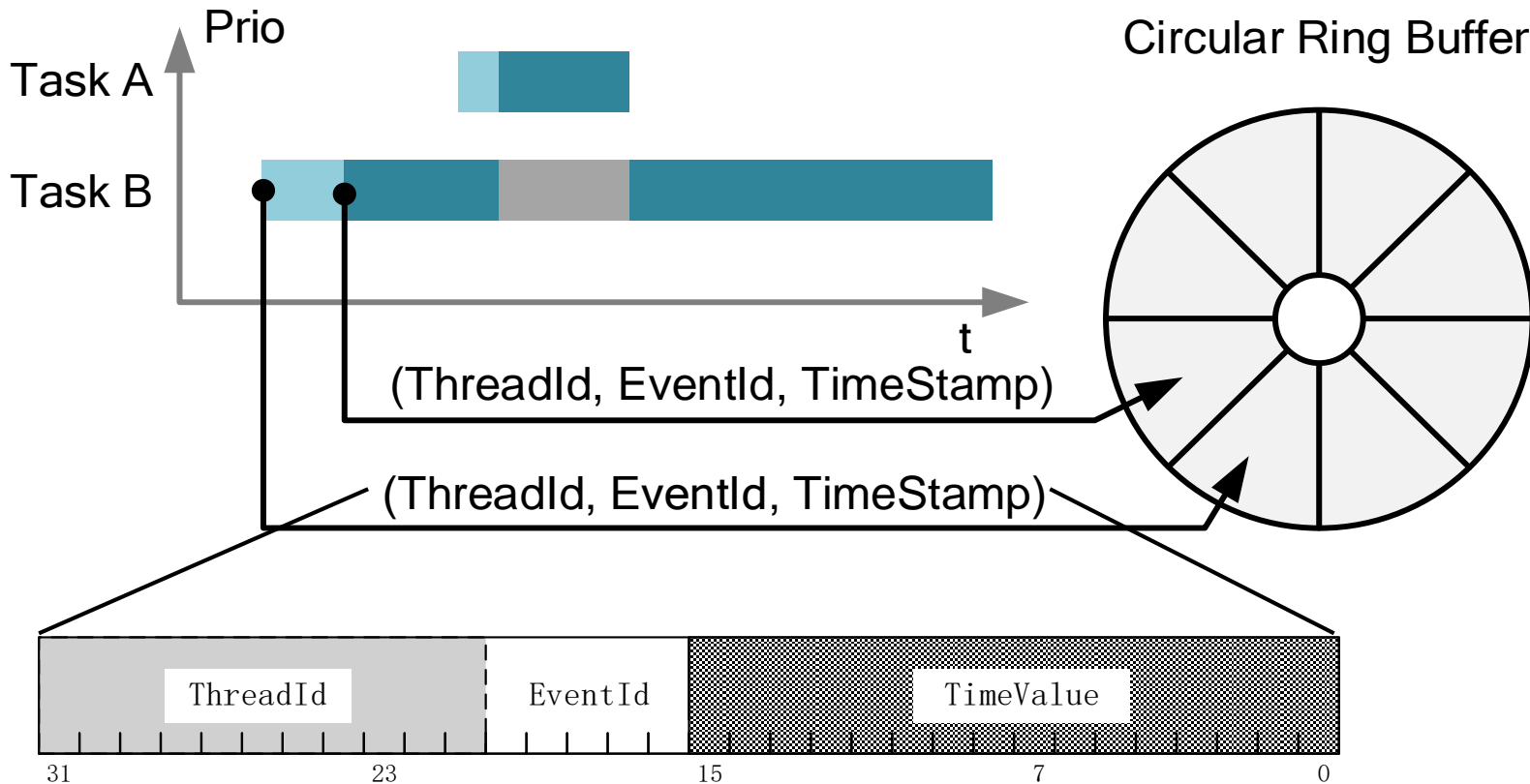
3.2 基于TA的AUTOSAR多核OS建模与优化



3.3 自主开发的AUTOSAR多核OS监控与优化



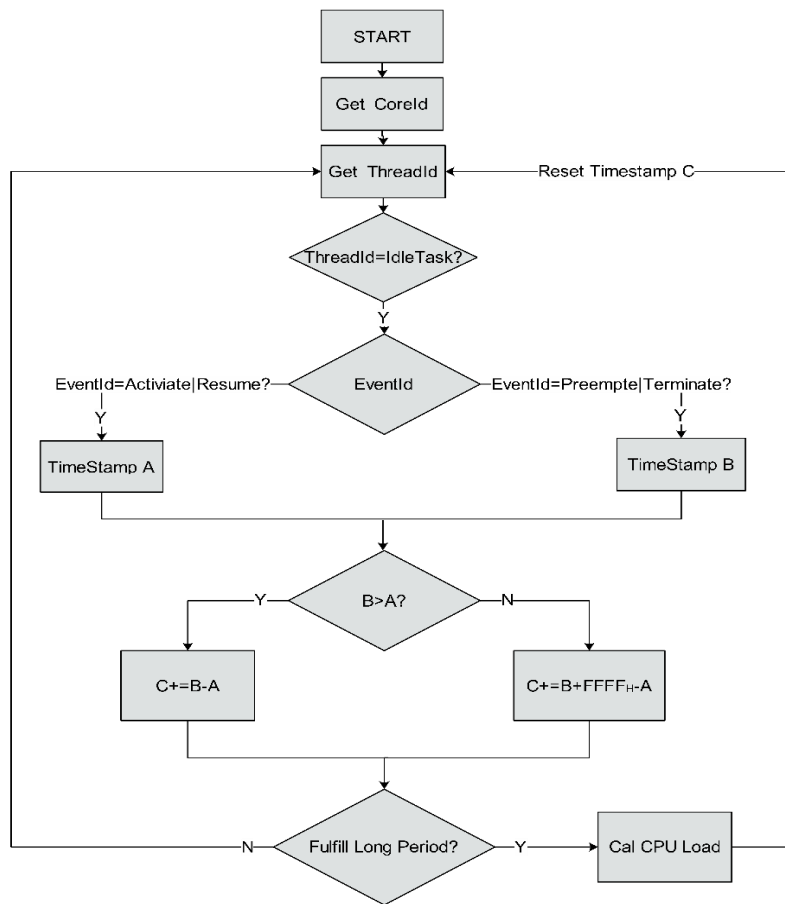
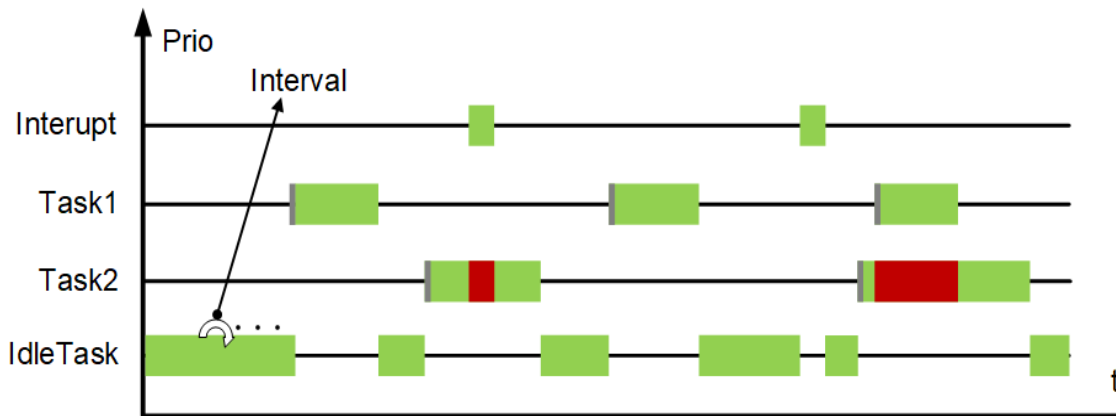
3.3.1 监控方案设计



3.3.1 监控方案设计

平均负载率:

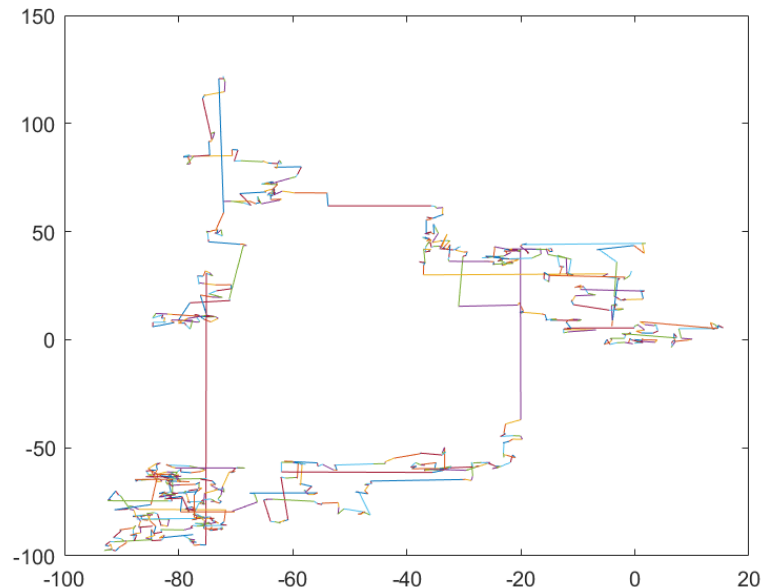
某一个时间段内处理器内核使用状况的统计指标



3.3.2 优化方案设计——优化算法

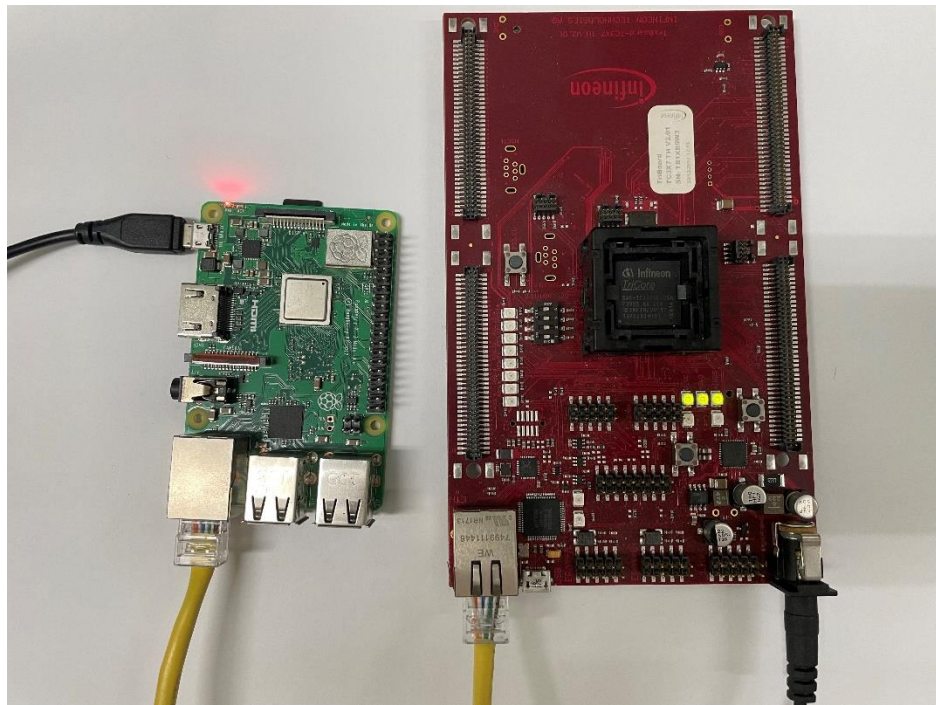
布谷鸟搜索算法CS (Cuckoo Search) : 2009年由Yang等人基于自然界中布谷鸟的育雏行为和动物觅食轨迹的莱维飞行提出的元启发式优化算法。

布谷鸟俗称杜鹃，不筑巢孵卵，而是将自己的卵放在别的鸟巢中，由其他鸟代为孵化、育雏。布谷鸟寻找宿主行进的路线服从莱维飞行规律，以此寻找最适合的寄生巢穴，提高卵的存活率。



3.3 自主开发的AUTOSAR多核OS监控与优化

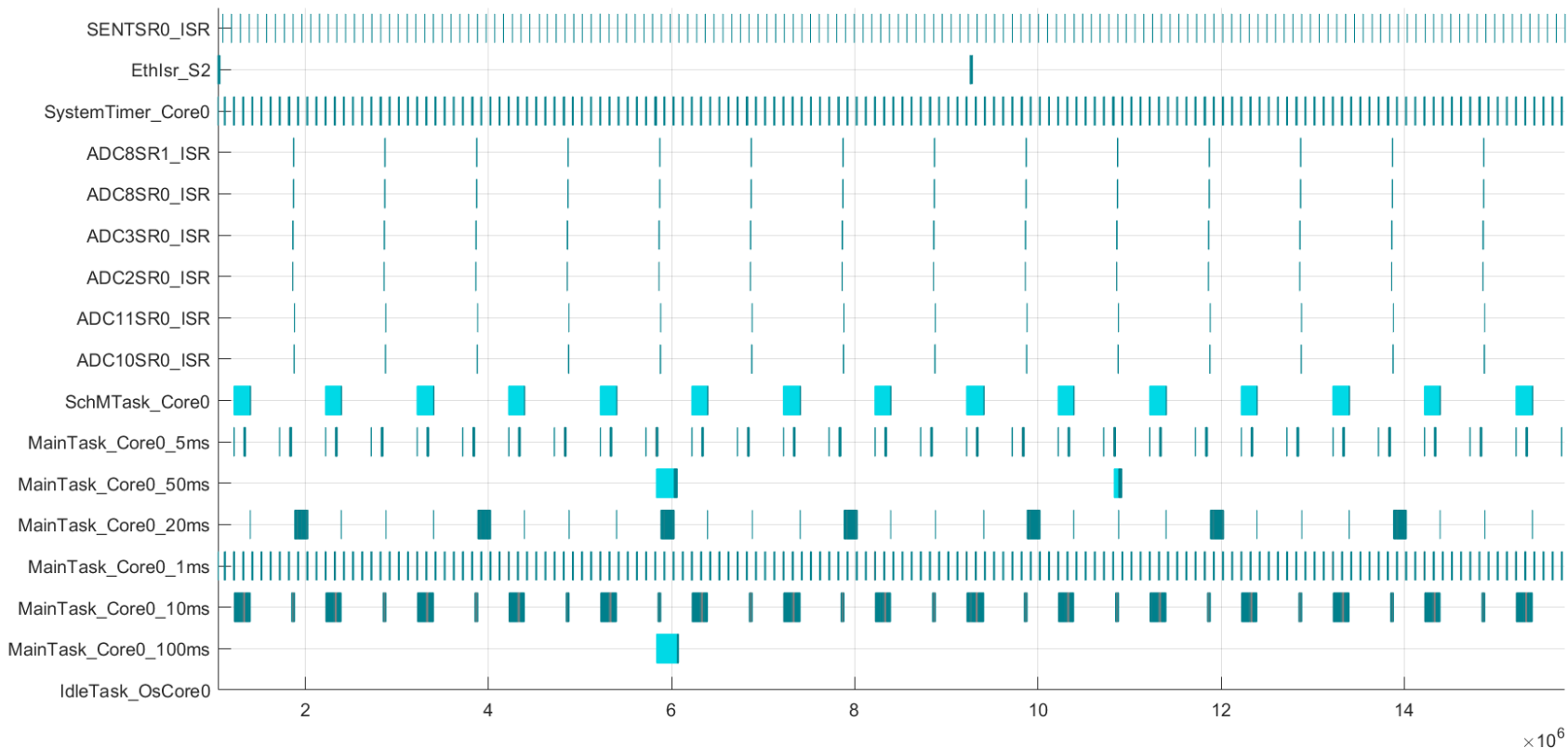
3.3.3 实验平台验证



```
pi@raspberrypi: ~/Mars/MasterJob_FinishedCode
文件(F) 编辑(E) 标签(T) 帮助(H)
pi@raspberrypi:~ $ cd Mars/MasterJob_FinishedCode/
pi@raspberrypi:~/Mars/MasterJob_FinishedCode $ ./CpuLoad_EXE
Udp Transmit start
Please enter command for AURIX
1 For CPU Load
2 For System Info
1
Command transmit successfully
Receiving data.....
Core0 CpuLoad: 46.186%
Core1 CpuLoad: 26.595%
Core2 CpuLoad: 26.324%
Core3 CpuLoad: 10.704%
Core4 CpuLoad: 11.871%
Core5 CpuLoad: 0.936%
Received completed
```

```
pi@raspberrypi: ~/Mars/MasterJob_FinishedCode
文件(F) 编辑(E) 标签(T) 帮助(H)
pi@raspberrypi:~ $ cd Mars/MasterJob_FinishedCode/
pi@raspberrypi:~/Mars/MasterJob_FinishedCode $ ./printinfo_EXE
Please enter command for AURIX
1 For CPU Load
2 For System Info
2
UDP Connecting with AURIX...
Receive Frame Completed
Frames Received
Frames restored in Frames.txt
pi@raspberrypi:~/Mars/MasterJob_FinishedCode $
```

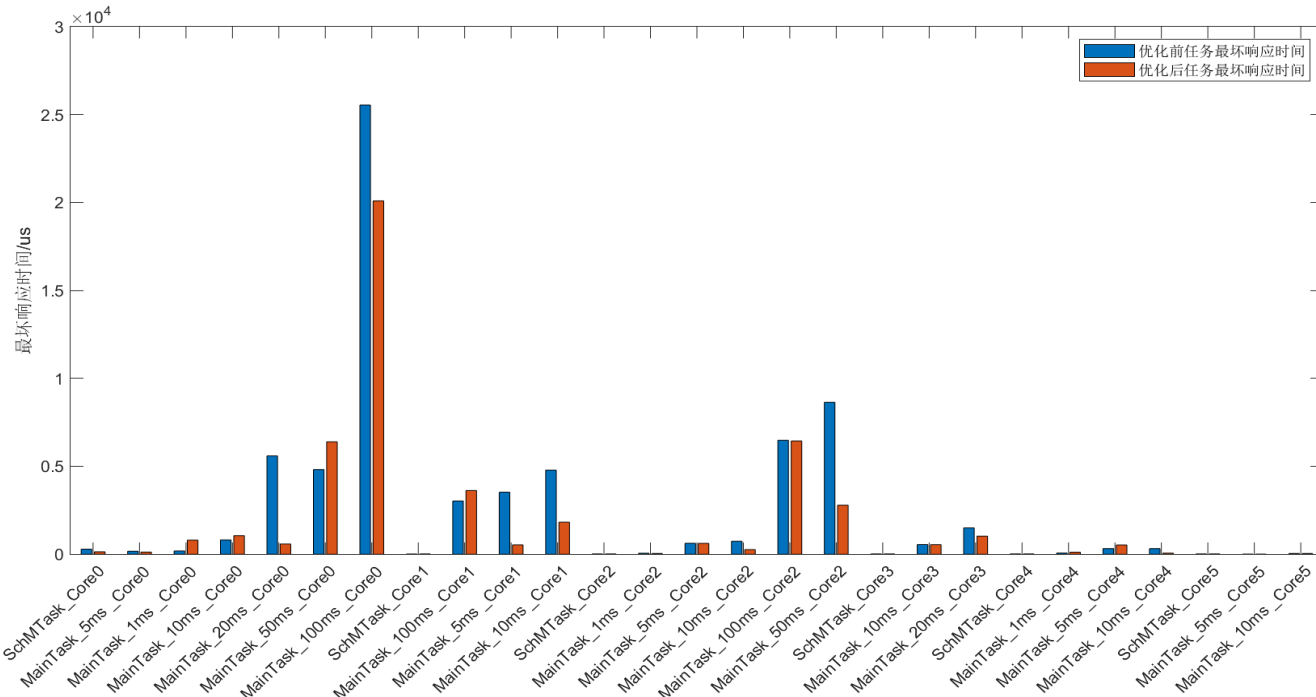
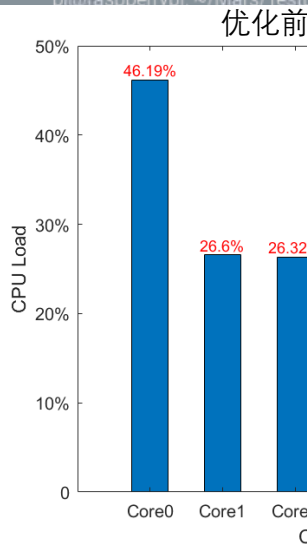
3.3.4 监控实验结果



3.3 自主开发的AUTOSAR多核OS监控与优化

3.3.5 优化实验结果

pi@raspberrypi: ~/Mars/TestETHTrans/Completed_Code



```
Core4 86 MainTask_10ms_Core2
Core5 41 MainTask_100ms_Core2
Core2 50 MainTask_50ms_Core2
Core3 29 SchMTask_Core3
Core4 84 MainTask_10ms_Core3
Core0 04 MainTask_20ms_Core3
Core4 40 SchMTask_Core4
Core2 79 MainTask_1ms_Core4
Core4 83 MainTask_5ms_Core4
Core2 84 MainTask_10ms_Core4
Core5 29 SchMTask_Core5
Core2 33 MainTask_5ms_Core5
Core3 10 MainTask_10ms_Core5
Overall Parameter:1.644 RT Parameter:0.669 CPU Load Parameter:0.975
pi@raspberrypi:~/Mars/TestETHTrans/Completed_Code $
```

多核OS实时监控

- ✓ 监控任务时序，防止功能错误
- ✓ 监控数据流，分析数据交互
- ✓ 监控内核负载，评估系统性能
- ✓ 监控任务运行，评估时序指标

多核OS建模优化

- ✓ 建模仿真，利于前期多核OS架构设计
- ✓ 模拟运行，便于分析设计缺陷
- ✓ 算法优化，充分发挥多核处理器性能

自主多核OS监控与优化方案的初探

- ✓ 尝试将软件监控程序与仿真优化功能整合，对AUTOSAR多核OS进行实时监控，基于监控数据对多核OS的任务部署进行优化计算，依此给出多核OS的优化建议

感谢各位聆听!

