确定性执行中间件需求说明

1. **文档编写/修改记录**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 版本  Version | 状态  Status | 编制人  Author | 编制日期  Date | 批准人  Approver | 批准日期  Approve Date | 内容描述  Description |
| V1.0 | In Review | 祝含颀 | 2022-4-14 |  |  | 发布稿 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**目 录**

**[0.](#_Toc101441799)****[文档编写/修改记录](#_Toc101441799)** [2](#_Toc101441799)

**[1.](#_Toc101441800)****[目的](#_Toc101441800)** [4](#_Toc101441800)

**[2.](#_Toc101441801)****[范围](#_Toc101441801)** [4](#_Toc101441801)

**[3.](#_Toc101441802)****[定义和缩写](#_Toc101441802)** [5](#_Toc101441802)

**[3.1.](#_Toc101441803)****[缩写](#_Toc101441803)** [5](#_Toc101441803)

**[3.2.](#_Toc101441804)****[定义](#_Toc101441804)** [5](#_Toc101441804)

**[4.](#_Toc101441805)****[正文](#_Toc101441805)** [6](#_Toc101441805)

**[4.1.](#_Toc101441806)****[需求： 工作流调度器（任务配置管理）](#_Toc101441806)** [6](#_Toc101441806)

**[4.2.](#_Toc101441807)****[需求： 工作流调度器（分析工具）](#_Toc101441807)** [6](#_Toc101441807)

**[4.3.](#_Toc101441808)****[需求： 工作流调度器（工作流调度器）](#_Toc101441808)** [7](#_Toc101441808)

**[4.4.](#_Toc101441809)****[需求: 数据流调度器（数据流调度器）](#_Toc101441809)** [7](#_Toc101441809)

**[4.5.](#_Toc101441810)****[需求: 数据流调度器（数据流调度器）](#_Toc101441810)** [8](#_Toc101441810)

**[4.6.](#_Toc101441811)****[需求: 配套服务（监控、诊断、日志）](#_Toc101441811)** [8](#_Toc101441811)

**[附录 A 指标项](#_Toc101441812)** [9](#_Toc101441812)

**[附录 B 系统性能摸底计划](#_Toc101441813)** [10](#_Toc101441813)

# **目的**

为自动驾驶解决方案提供确定性调度中间件、以及相应辅助工具集，保障系统多任务并发执行的

执行**确定性**、智能驾驶数据流和工作流调度的**动态性**、深度学习模型的执行**高效性**。

**确定性：**任务需要按照优先级顺序完成执行。用户的响应时间需要小于截止期D1，用户的执行完成时间需要小于截止期D2。由于一个节点可能同时被多个callback chain所共享，因此任务实际执行的优先级会区别于用户给定的优先级。

**高效性：**系统需支持一定数量的任务。系统中断响应性能等不应过多损耗。系统资源利用率不应过多损耗。

**动态性：**首先，自动驾驶任务内在的执行时间可能根本上是用例相关，不确定的。单纯的最坏时间估计会导致过分悲观的资源分配。第二，由于硬件本身不存在强隔离，任务的同时执行也会造成任务间干涉。第三，自动驾驶工作流本身会随着模式切换，发生动态调整。需要找出对实时性有最坏影响的关键模式。因此不能简单的进行完全静态的优先级指定。优先级分配需要在运行时动态决定。

此外，中间件还需减少上层开发人员所需的**专业知识要求**和**编码负担**。体现在，

**专业知识要求**：一方面由于编码人员不一定具备操作系统、实时调度、通信方面专业知识。另一方面系统集成时，获取所有任务每一个子模块详细的并发数量、状态机、函数调用关系、通信矩阵、最坏执行时间，是非常费时费力的。因此需要在产品中减少对用户暴露出的复杂性。

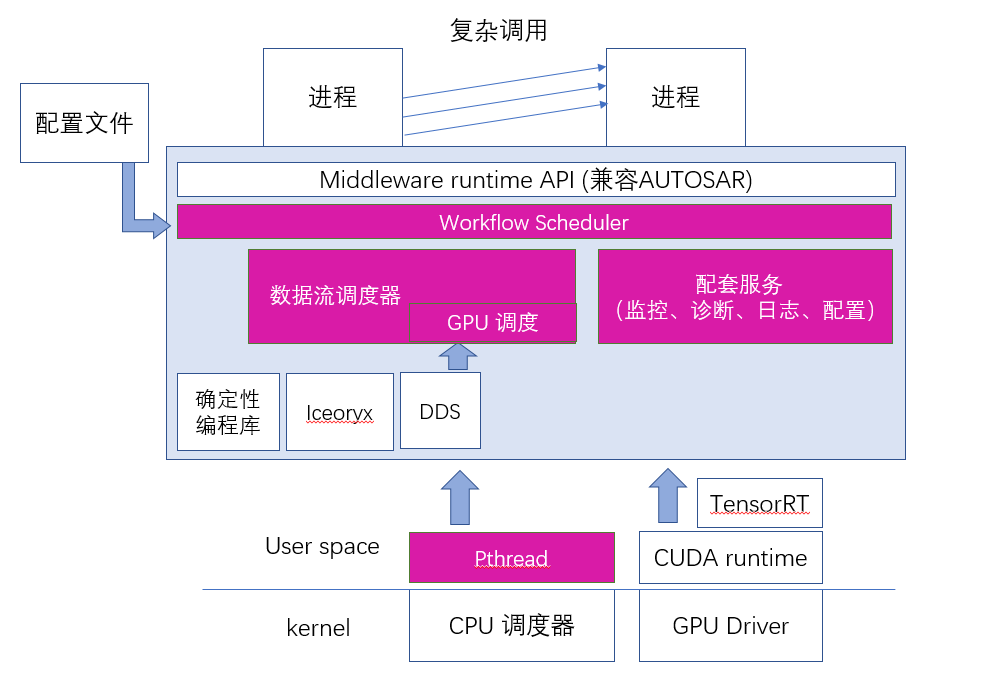
**编码负担**：需减少API对上层应用逻辑的变更，以减少编码负担。同时应具备自动标定的能力，不应在新的应用加入，或任务逻辑发生变化时，对整个系统的时序重新进行手工设计。

# **范围**

运行环境：支持打有PREEMPT-RT补丁的Linux系统，运行环境为C++11，只支持单语言。运行平台为多核CPU + GPU 环境。可支持大于等于一块GPU。 支持跨SOC调度。

任务类型：对于调用本中间件通信、执行管理等API启动的周期性任务，可以提供确定性执行顺序、确定性时间保证。对于其他任务，不进行任何保证。上层任务集最终以用户态线程组或进程组的形态向上呈现。

应用接口：对上参照AP AUTOSAR的API规范。通信模块没有限定DDS版本，可以后期适配不同的DDS实现。

**

阶段性交付物：

工作流调度器、数据流调度器、配套服务、优化后的线程管理模块

# **定义和缩写**

## **缩写**

|  |  |
| --- | --- |
| 缩写 | 描述 |
| 无 | 无 |

## **定义**

|  |  |
| --- | --- |
| 定义 | 描述 |
| 验证标准 | 通过何种方式及何种结果来验证该需求已经实现  a) Review b) Test：c) Review & Test |

# **正文**

## **需求： 工作流调度器（任务配置管理）**

**Description:**

初始化并分发配置文件。向上提供AUTOSAR兼容API

**内在属性**：是否存在多种服务质量模式（低精度推理，跳帧执行等）

**互联参数**：任务的前序节点

**资源参数**：任务最好最坏执行时间（此项可为空）

**时间参数**，包括周期、截止期，可忍受的波动范围（此项可为空）

**Difficulty:** 低

**Impact:** 需求变动导致影响低

**Verification Method:** Test & Review

**Requirement:** 用户指定

## **需求： 工作流调度器（分析工具）**

**Description:**

1.产生一些虚假任务，模拟不同任务在不同硬件上的资源负载

2.对任务离线进行单任务、多任务共存情况下最坏时间估计、最好时间估计

3.离线对任务的资源占用、到达时间建立一个**预测模型**。在线进行预测。

**Difficulty:** 高

**Impact:** 需求变动导致影响大

**Verification Method:** Test & Review

**Requirement:** 硬件资源服务，用户配置

## **需求： 工作流调度器（工作流调度器）**

**Description:**

1. 接收配置管理的配置，离线调用分析工具进行分析。运行时在线接受分析工具的预测结果。

2. 统一进行工作流和数据服务的调度，进行任务的**可调度性分析**。如果一个新任务启动时，或已运行任务发生模式变更时，进行可调度性分析，若不通过，则报告监控器。根据可调度性分析结果，进行动态的确定任务启动时间、结束时间、是否降低其服务质量。

**Difficulty:** 高

**Impact:** 需求变动导致影响大

**Verification Method:** Test & Review

**Requirement:** 硬件资源服务，用户配置

## **需求: 数据流调度器（数据流调度器）**

**Description:**

1. 通过调度算法动态编排**callback chain**任务的CPU绑定和优先级。

2. 通过API劫持的方式截获CUDA相关指令，并根据当前硬件环境和CUDA指令特性、数据特性进行编排发射。不会涉及算子内部调优。

**Difficulty:** 高

**Impact:** 需求变动导致影响大

**Verification Method:** Test & Review

**Requirement:** 硬件资源服务，工作流调度器

## **需求: 数据流调度器（数据流调度器）**

**Description:**

1. 接受上层应用信息，通过pthread库对线程、进程进行同步状态管理，以及动态绑定CPU，静态绑定GPU，以及确定优先级关系。将本层的信息通过CPU调度器 、PREEMPT补丁配置接口**传递给内核**

2. 可能的内核改动

**Difficulty:** 高

**Impact:** 需求变动导致影响大

**Verification Method:** Test & Review

**Requirement:** 数据流调度器

## **需求: 配套服务（监控、诊断、日志）**

**Description:**

1.**周期性收集**系统内的现有硬件资源，提供给数据流调度器。以一定周期提供服务，频率根据系统的运行情况动态变化。监听内容包括GPU利用率、 CPU利用率、 吞吐率、Cache 命中率、memory 利用率、bus、memory带宽。硬件资源服务需要长期运行，需要维持较低的开销

2.监控硬件资源服务的输出和数据流调度器的执行情况，诊断是否存在**异常状态变化**，例如未来可能的超时行为，服务质量降低。

3.允许进入调试模式，在调试模式下可以指定数据流调度器中的某条链路，或某个应用使其输出额外的任务**编排表**，并存储在本地。支持通过诊断工具进行数据筛选，统计分析，可视化。

4.考虑AUTOSAR兼容性

**Description:** 需要通过何种办法实现需求

**Difficulty:** 中

**Impact:** 需求变动导致影响中

**Verification Method:** Test & Review

**Requirement:** 无

**附录 A 指标项**

|  |  |  |
| --- | --- | --- |
| 名称 | 指标 | 上限、下限 |
|  | 任务99.9th完成时延 |  |
|  | 任务99.9th响应时延 |  |
|  | 代码修改 < x行 |  |
|  | 任务相对截止期抖动 |  |
|  | 任务错过截止期次数 |  |
|  | 各个API调用开销 |  |
|  | 调度器产生的CPU开销 |  |
|  | 系统平均功耗 |  |
|  | Cyclicttest 响应时延 |  |
|  | 最大支持任务数量 |  |
|  | 与CUDA graph 、CyberRT、Driveworks性能对比 |  |
|  | 额外显存开销 |  |

**附录 B 系统性能摸底计划**

搭建一个稳定的开发平台，确认可用的硬件资源、软件资源

* 1. **工作流调度工作计划**

对需要调用的 TensorRT/CUDA 库API列表进行性能测量，预计两周

对多任务之间干涉进行测量，预计两周

* 1. **Kernel 工作计划**

搭建Drive OS kernel编译环境，预计1周

对 RT-Preempt补丁、 Pthread 源码进行学习，预计2 周

对 RT-Preempt补丁进行测量，预计2周