

异构算力硬件调度关键技术研究

陈 榕

上海交通大学

2025. 2

陈榕

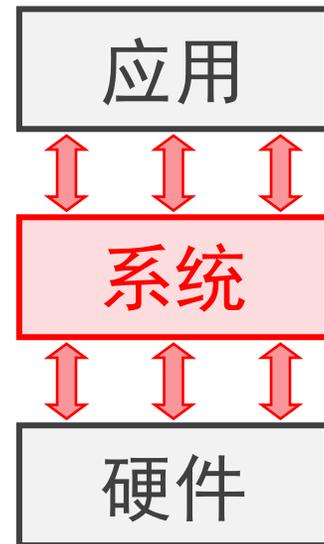
- ▶ 并行与分布式系统研究所（IPADS），上海交通大学

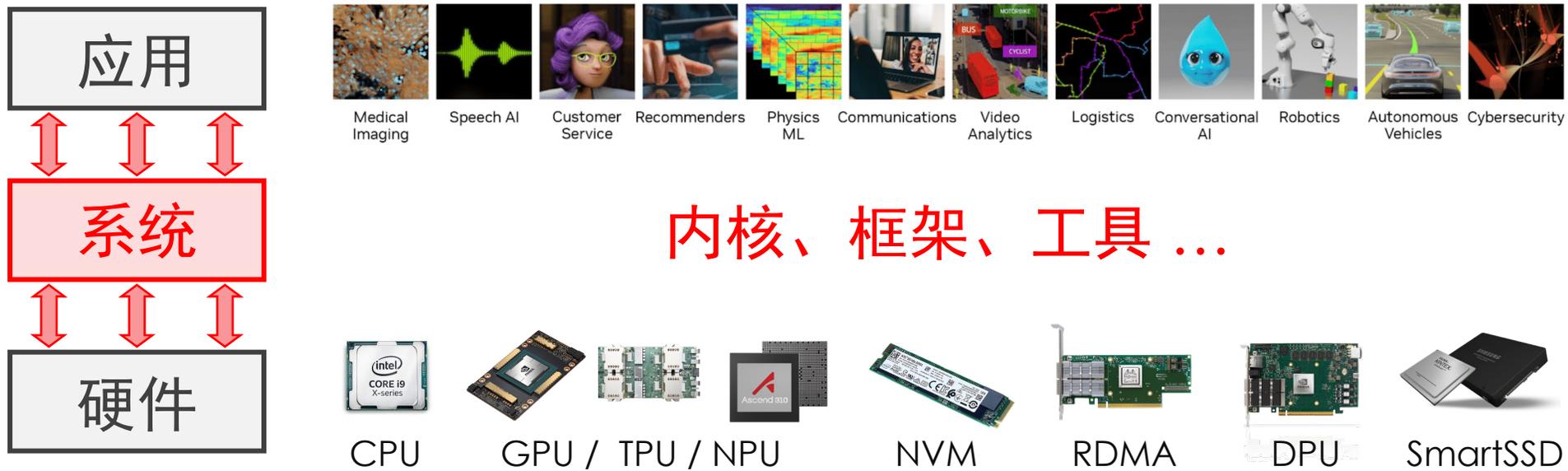
研究领域：**基础系统软件（操作系统、分布式系统等）**

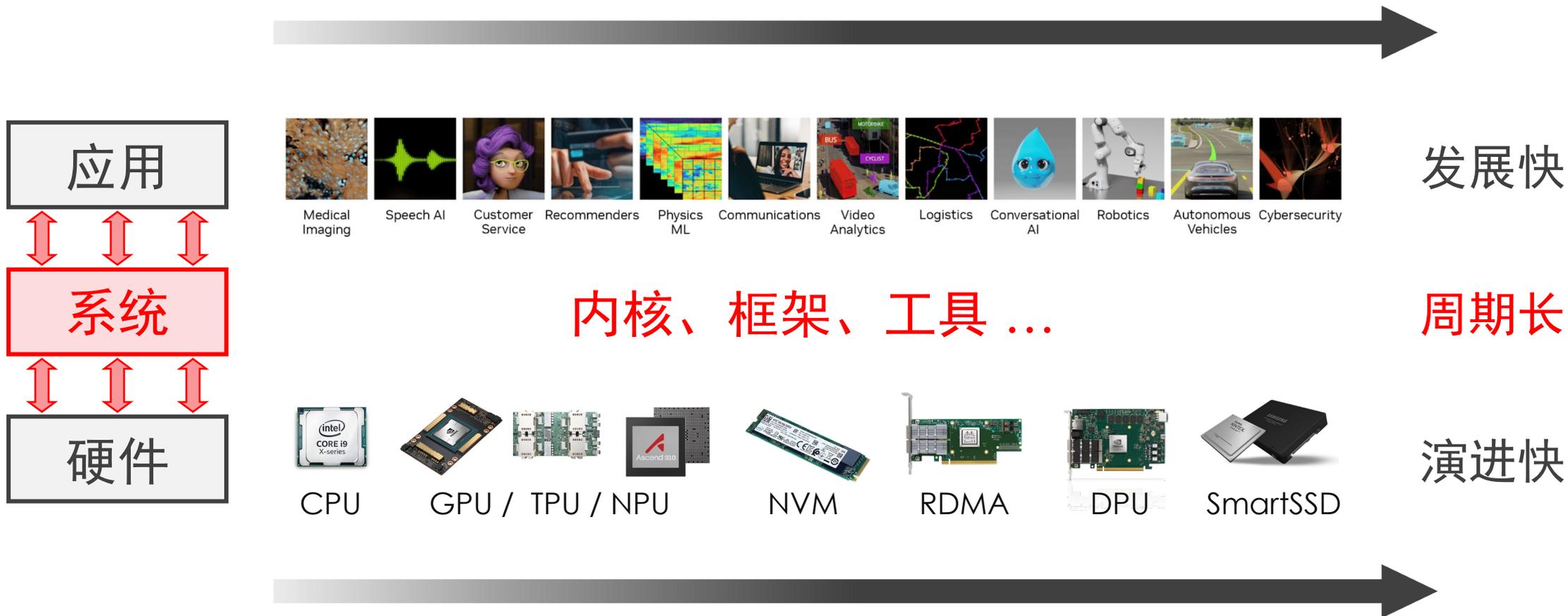
- ▶ ACM Distinguished Member

“ for contributions to improving the performance and scalability of distributed systems ”

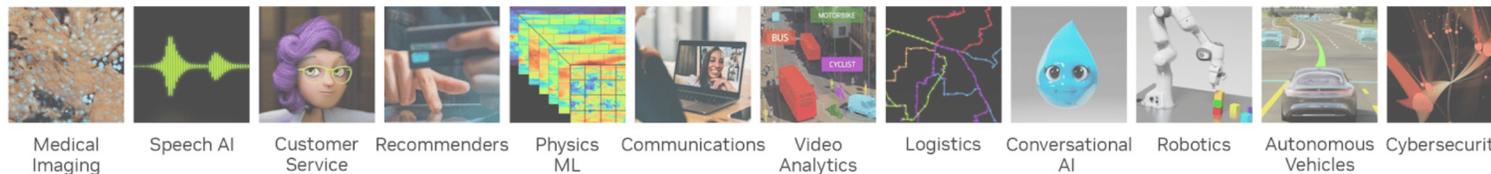
- ▶ OSDI/SOSP（13篇）、最佳论文奖（EuroSys 2024/2015）
- ▶ 2020年 华为“奥林帕斯先锋奖”（第一完成人）







应用



发展快

需求

高吞吐、低时延、可扩展、大规模 ...

发展趋势

系统

内核、框架、工具 ...

源自“共性”

能力

算力、存力、带宽、持久、隔离 ...

演进趋势

硬件



演进快

系统软件研究——智能时代



应用

智能应用



需求



系统



能力

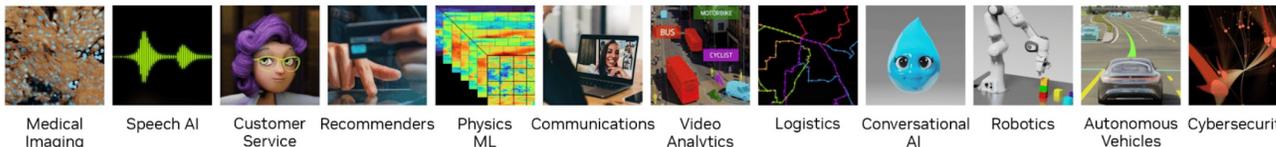
硬件

系统软件研究——智能时代



应用

智能应用



需求

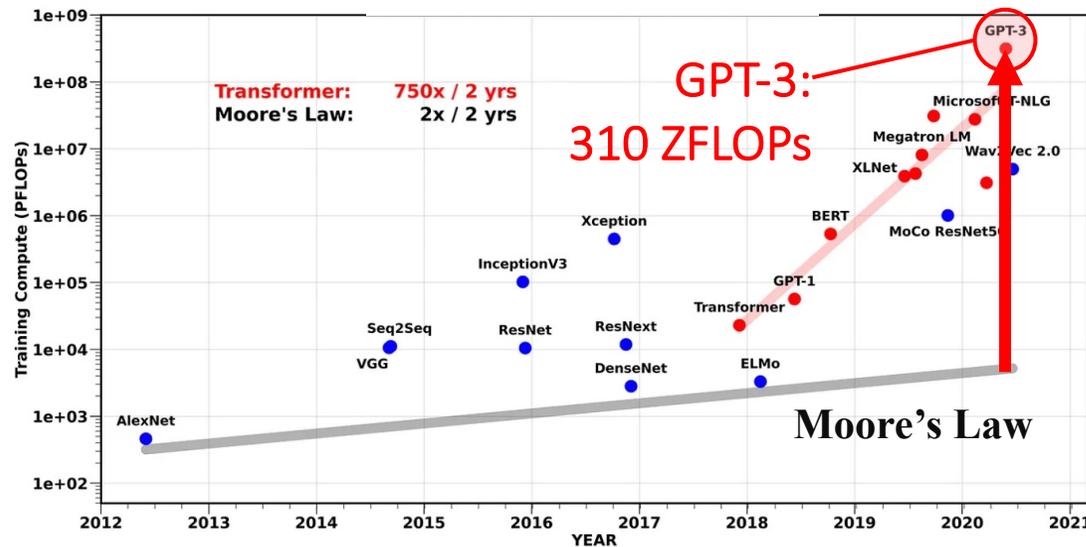
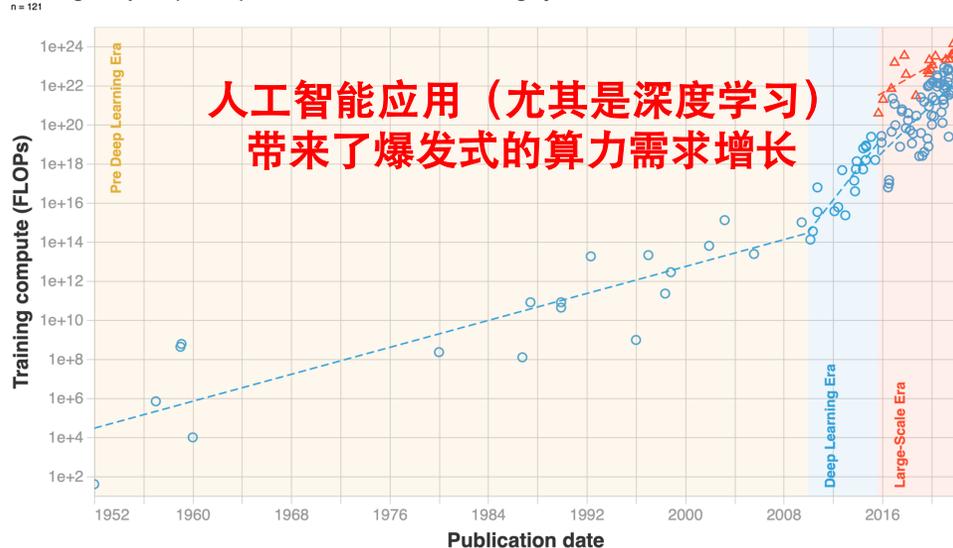
大算力

系统

能力

硬件

Training compute (FLOPs) of milestone Machine Learning systems over time

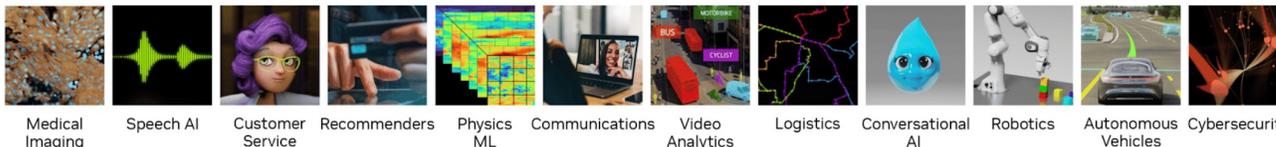


系统软件研究——智能时代



应用

智能应用



需求

大算力 + 强实时、高性价比、大内存、..

↑

系统



智能车

1. 障碍物检测
2. 红绿灯识别
3. 语音助理
4. 疲劳监测
5. ...



↓

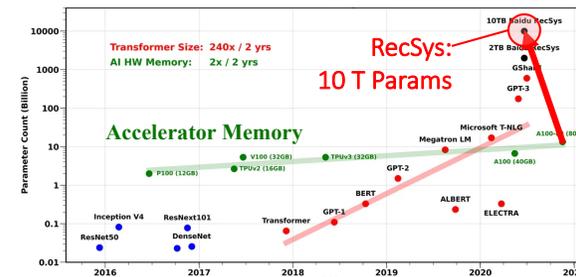
能力

硬件

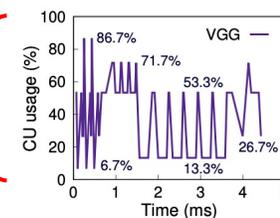


典型DNN任务: 3.5~13.6ms

Model	ResNet	DenseNet	VGG	Inception	Bert
#Kernels	307	207	55	146	205
Exec. Time	13.6	3.5	4.4	8.3	5.4



内存墙



算力需求
亚毫秒级
剧烈波动

系统软件研究——智能时代



应用

智能应用



需求

大算力 + 强实时、高性价比、大内存、..

系统

GPU

TPU / NPU / ASIC ...

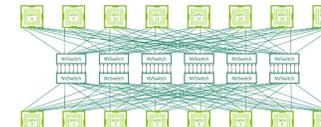
NVLink / NVSwitch / RDMA



A100
108 SMs
6,912 cores



900GB/s total



3.6TB/s BW

能力

大算力、领域加速、异构互联

硬件

GPU、NPU/XPU、NVLink/NVSwitch/RDMA

系统软件研究——智能时代



应用

智能应用



需求

大算力 + 强实时、高性价比、大内存、..

系统



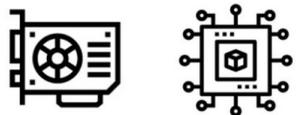
异构算力
操作系统
关键技术



能力

大算力、领域加速、高速互联

硬件



GPU、NPU/XPU、NVLink/NVSwitch/RDMA

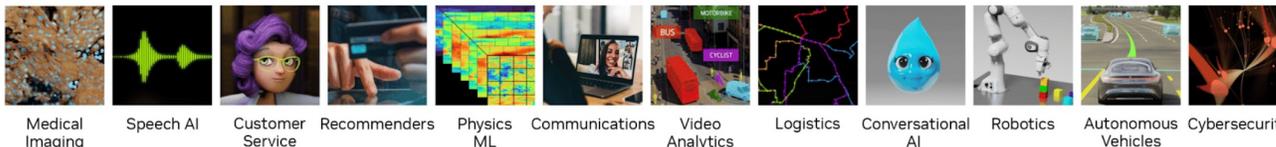


系统软件研究——智能时代



应用

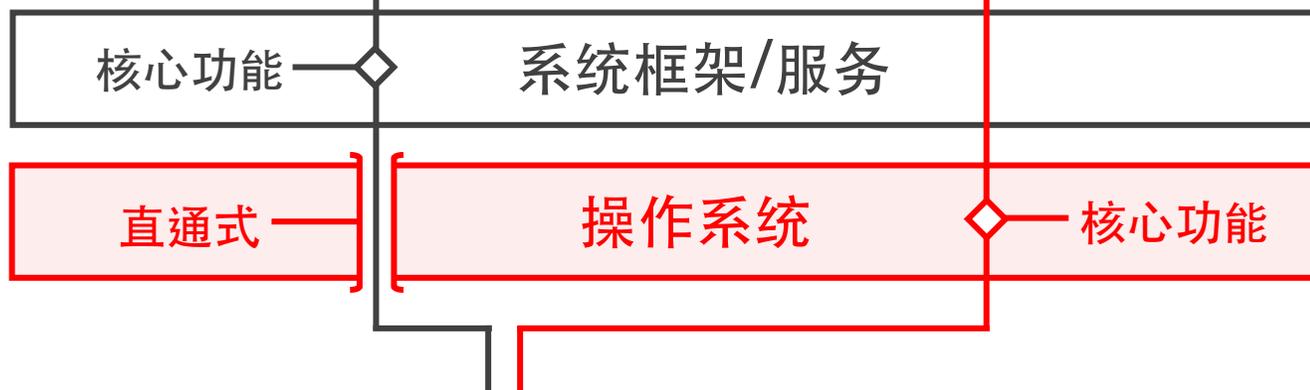
智能应用



需求

大算力 + 强实时、高性价比、大内存、..

系统



异构算力
操作系统
关键技术

能力

大算力、领域加速、高速互联

硬件



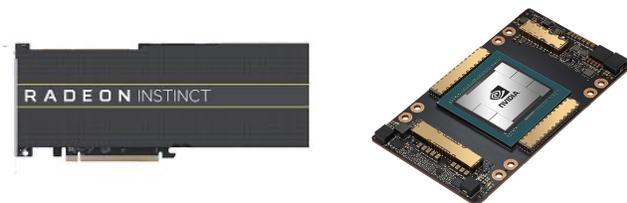
GPU、NPU/XPU、NVLink/NVSwitch/RDMA



算力硬件繁荣



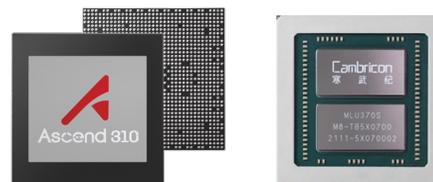
GPU



TPU



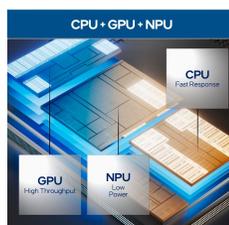
NPU



SoC



AI·PC



What is an AI PC?

A PC with new NPU silicon that brings new AI experiences in productivity, creativity, and security through a combination of the CPU, GPU, and the NPU.

AI·PC 是一个“混合体”

在硬件上集成了**混合AI算力单元**
且能够本地运行**个人大模型**
创建个性化的**本地知识库**
实现**自然语言交互**

——《AI·PC产业(中国)白皮书》

- Intel Core Ultra
- Comes with new NPU, CPU, and GPU powered silicon
- Comes with Copilot
- Has the Copilot key

- Apple M1
- 8-core CPU
- 8-core GPU
- Secure Enclave
- 16-core Neural Engine

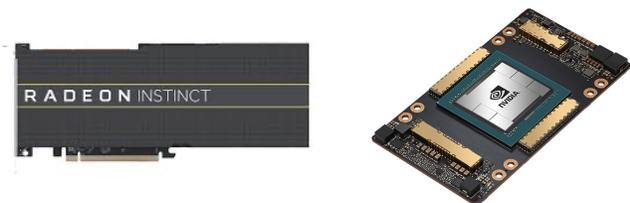
- AMD RYZEN AI
- Leads the AI PC era
- World's first x86 processor with integrated NPU
- AI PC
- 荣耀MagicBook Pro 16
- 发布时间: 2024年3月18日 19:30

- Lenovo AI PC
- Lenovo CES 24
- Smarter technology for all

算力硬件多任务调度现状



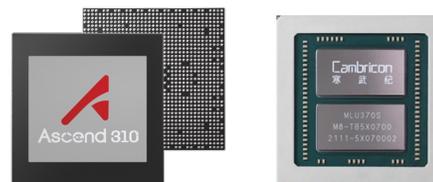
GPU



TPU



NPU



SoC



现有算力硬件（XPU: GPU / NPU / DSA / ...）任务调度方法

- × 硬件调度：功能有限（First-Come-First-Serve, RR）、不同硬件差异巨大
- × 软件调度：绑定特定软硬件功能（+修改）、实现工作量大、方法迁移困难

多任务调度问题演示



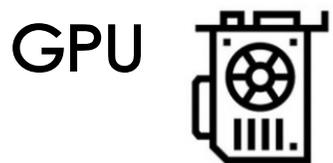
NPU
原生调度

NPU
抢占式调度

XPU
协同调度

App#1: <https://github.com/fangfufu/Linux-Fake-Background-Webcam>
App#2: <https://github.com/ggerganov/whisper.cpp>

GPU多任务调度



高并发算力



GPU Architecture (Nvidia Ampere)

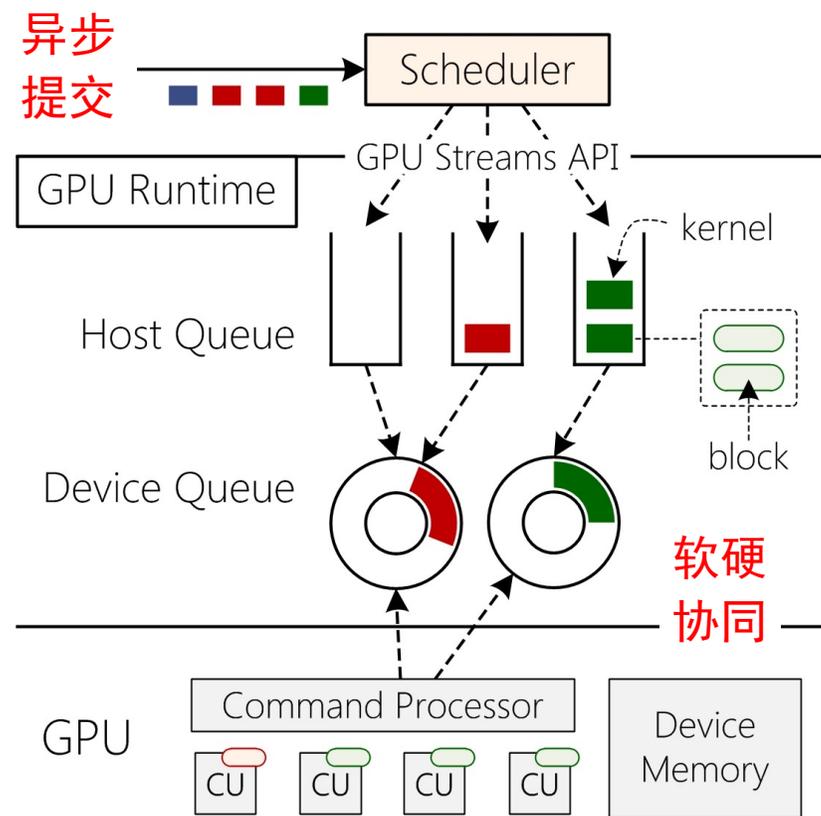
1 GPU
x 8 GPCs / GPU
x 8 TPCs / GPC
x 2 SMs / TPC
= 128 SMs*

Streaming Multiprocessor



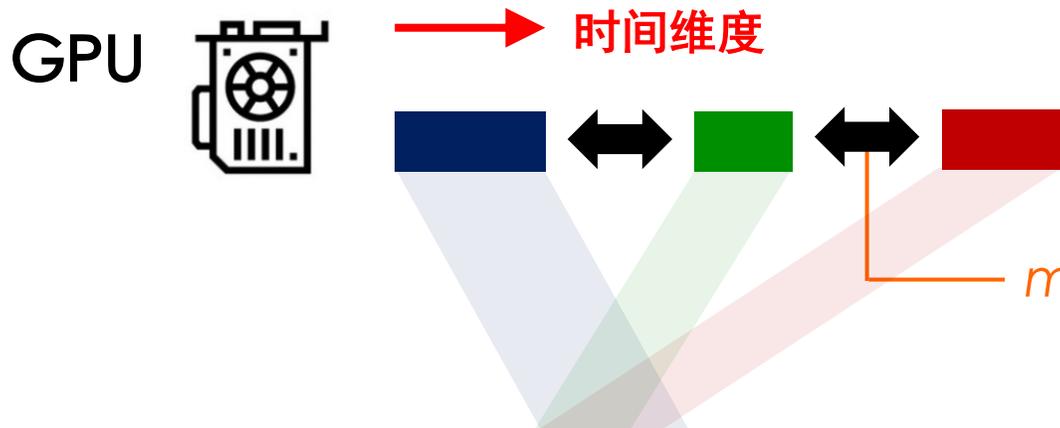
2048 Thds/SM

GPU任务调度



* GPU调度单元 NVIDIA使用SM、ARM使用CU

GPU多任务调度



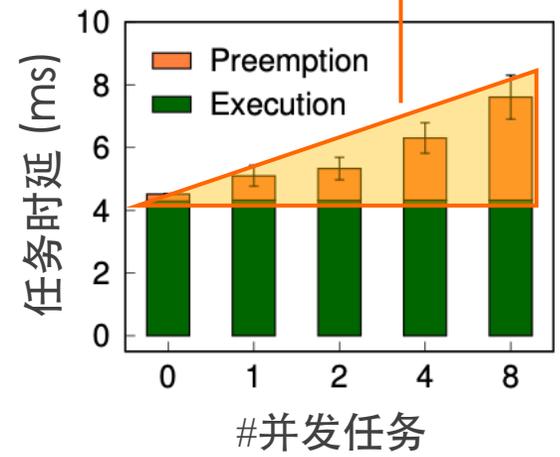
vs. CPU

- 执行时间更长 (5~100ms)
- 调度延迟更低 (~5μs)

典型DNN推理任务: 3.5~13.6ms

Model	ResNet	DenseNet	VGG	Inception	Bert
#Kernels	307	207	55	146	205
Exec. Time	13.6	3.5	4.4	8.3	5.4

Testbed : AMD Radeon Instinct MI50 GPU



关键技术: GPU实时任务抢占

* 未考虑GPU内存切换、任务加载等开销

问题/挑战

1. 大算力硬件状态多
任务切换慢 } $300\mu s$

GPU A100 

- 128 SMs
- 256 KB regs/SM
- 164 KB shmem/SM

 CPU

$< 1\mu s$
切换延迟

思路/方法

关键洞见：GPU任务多有“幂等性”



重置执行中的任务（不保存状态）

切换延迟 $5\mu s$



问题/挑战

1. 大算力硬件状态多 } 300 μ s
任务切换慢

GPU A100 

- o 128 SMs
- o 256KB regs/SM
- o 164KB shmem/SM

CPU 

< 1 μ s
切换延迟

2. 软硬协同异步提交 } > 1ms
任务清理慢

典型DNN推理: 50~300+任务

- o ResNet(307), BERT(205), VGG(55)

思路/方法

关键洞见: GPU任务多有“幂等性”



重置执行中的任务 (不保存状态)

切换延迟 5 μ s

关键设计: 垂直全栈清理

[软] Host Qs: 软件重置队列

[软-硬] Dev Qs: 编译插桩+主动退出

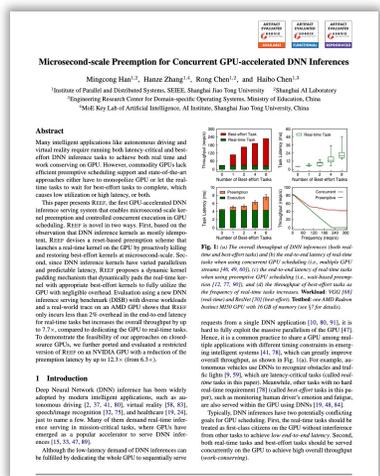
[硬] GPU SM: 硬件指令重置

} ~30 μ s
清理延迟



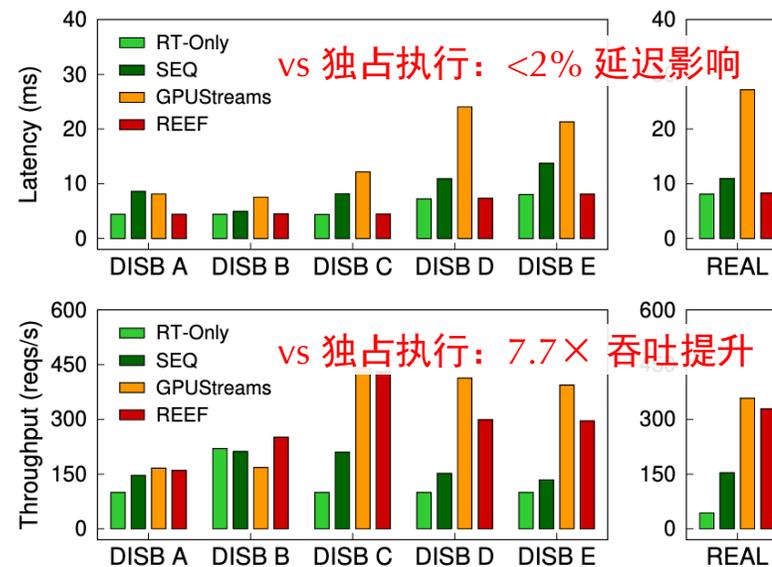
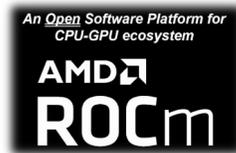
Microsecond-scale Preemption for Concurrent GPU-accelerated DNN Inferences

- ▶ 平台：量产GPU（AMD MI50 GPU）、开源软件栈（AMD ROCm）
- ▶ 支持优先级调度：高优先级（实时任务 / RT）、低优先级（后台任务 / BE）
- ▶ 多任务执行： $<2\%$ RT任务延迟影响、 $7.7\times$ 吞吐提升、百倍任务抢占延迟降低



OSDI 2022

“首次在量产GPU上实现了微秒级任务抢占和动态可控算力共享”



¹ Microsecond-scale Preemption for Concurrent GPU-accelerated DNN Inferences. OSDI 2022.

从“专用”到“通用”



- 利用AMD GPU 重置能力
- 修改驱动、运行时、应用
- 实现：5,500 LoC (C++)



AMD GPU

开源软件栈
(ROCm)

- ✓ 抢占性能好、灵活性高
- ✗ 硬件/软件栈依赖性强
- ✗ 开发/部署/升级难度大



成果得到学界
权威会议认可



¹ Microsecond-scale Preemption for Concurrent GPU-accelerated DNN Inferences. OSDI 2022.

从“专用”到“通用”



- 利用AMD GPU 重置能力
- 修改驱动、运行时、应用
- 实现：5,500 LoC (C++)
- 利用NV GPU时间片机制
- 进程级调度、不兼容MPS
- 实现：100 LoC (C++)



AMD GPU

开源软件栈
(ROCm)



NV GPU

闭源软件栈
(CUDA)

- ✓ 开发/部署难度低
- × 硬件/软件栈依赖性强
- × 功能和应用场景受限



滴滴当代及下一代
自动驾驶组成部分

从“专用”到“通用”



- 利用AMD GPU 重置能力
- 修改驱动、运行时、应用
- 实现：5,500 LoC (C++)
- 利用NV GPU时间片机制
- 进程级调度、不兼容MPS
- 实现：100 LoC (C++)



AMD GPU
开源软件栈
(ROCm)



NV GPU
闭源软件栈
(CUDA)

高优先级

低优先级

```
hmc@d502: ~ x hmc@d502: ~ x hmc@d501: /data2/hr x hmc@d501: /data2/hr x + - - - - -
requery_penalty = 0.000, presence_penalty = 0.000
dry_multiplier = 0.000, dry_base = 1.750, dry
_allowed_length = 2, dry_penalty_last_n = 4096
top_k = 40, top_p = 0.950, min_p = 0.050, xtc
_probability = 0.000, xtc_threshold = 0.100, typical_
p = 1.000, top_n_sigma = -1.000, temp = 0.800
mirostat = 0, mirostat_lr = 0.100, mirostat_e
nt = 5.000
sampler chain: logits -> logit-bias -> penalties -> d
ry -> top-k -> typical -> top-p -> min-p -> xtc -> te
mp-ext -> dist
generate: n_ctx = 4096, n_batch = 2048, n_predict = -
1, n_keep = 0

== Running in interactive mode. ==
- Press Ctrl+C to interject at any time.
- Press Return to return control to the AI.
- To return control without starting a new line, end
your input with '/'.
- If you want to submit another line, end your input
with '\n'.
- Using default system message. To change it, set a
different value via -p PROMPT or -f FILE argument.

system

You are a helpful assistant

> tell me a story
```

```
hmc@d502: ~ x hmc@d502: ~ x hmc@d501: /data2/hr x hmc@d501: /data2/hr x + - - - - -
different value via -p PROMPT or -f FILE argument.

system

You are a helpful assistant

> tell me a story
Here's a story for you:

In a small village nestled in the rolling hills of Pr
ovence, there lived a young girl named Sophie. Sophie
was known throughout the village for her extraordina
ry talent: she could create the most exquisite, intri
cate paper flowers. Her creations were so lifelike th
at they seemed to bloom before your very eyes.

Sophie's love affair with paper flowers began when sh
e was just a little girl. Her grandmother, a skilled
florist, would spend hours teaching Sophie the art o
f folding and shaping paper into delicate petals. As S
ophie grew older, she refined her skills, experimenti
ng with different colors, textures, and techniques. H
er creations became so popular that people would trav
el from all over to commission a bouquet from Sophie.

One day, the village was preparing for its annual Fêt
e de la Sainte-Vierge, a grand celebration in honor o
f the Virgin Mary. The villagers were busy decorating
the town square, hanging
```

从“专用”到“通用”



- 利用AMD GPU 重置能力
- 修改驱动、运行时、应用
- 实现：5,500 LoC (C++)
- 利用NV GPU时间片机制
- 进程级调度、不兼容MPS
- 实现：100 LoC (C++)



AMD GPU

开源软件栈
(ROCm)



NV GPU

闭源软件栈
(CUDA)

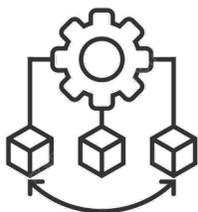


XPU多任务调度需求



Portability: 方法难以跨硬件移植

- × 依赖特定硬/软件特征, 难以复用
- × NPU/AI芯片: 无抢占调度方法



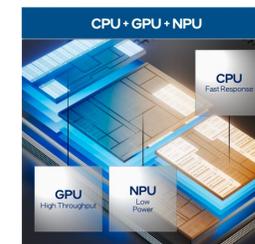
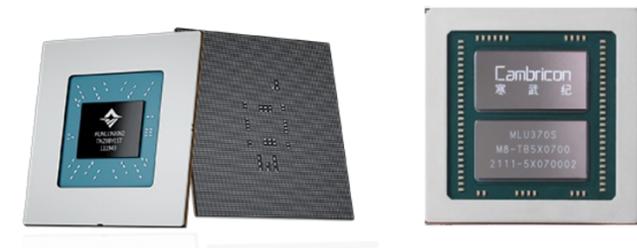
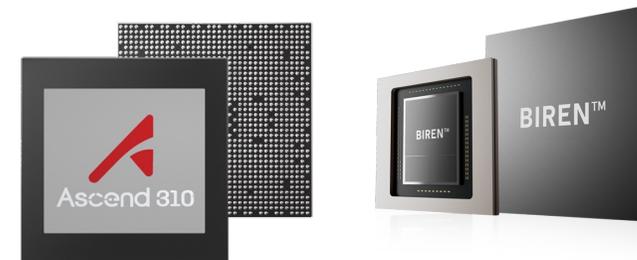
Uniformity: 调度抽象和接口不统一

- × 调度策略难以在不同平台迁移
- × 难以实现多XPU的混合协同调度



Evolvability: 软硬件实现紧密耦合

- × 很难集成新发布或未公开硬件特性
- × 很难排除对过时或禁用功能的依赖

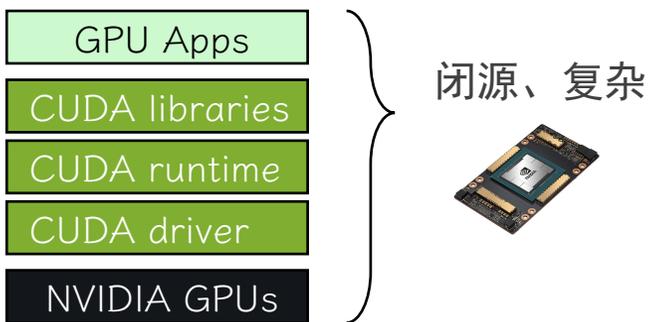


问题/挑战

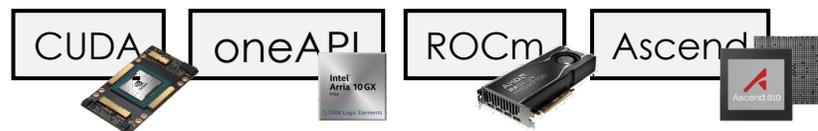
1. 如何提供XPU通用调度?

× XPU软件栈闭源、复杂

包括：驱动、运行时、应用

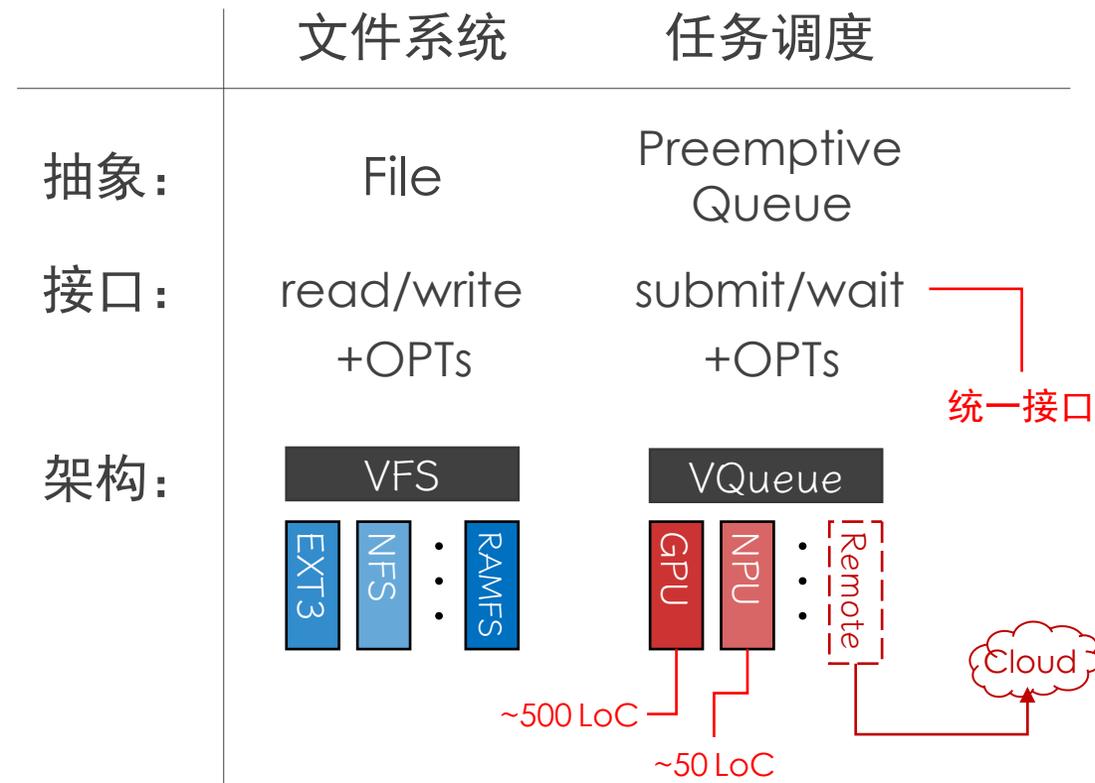


× XPU软件栈不兼容



思路/方法

关键思路：借鉴内核的统一抽象设计



问题/挑战

2. 如何支持不同种类XPU?

× XPU硬件架构差异大

GPU: CUDA/ROCm 程序

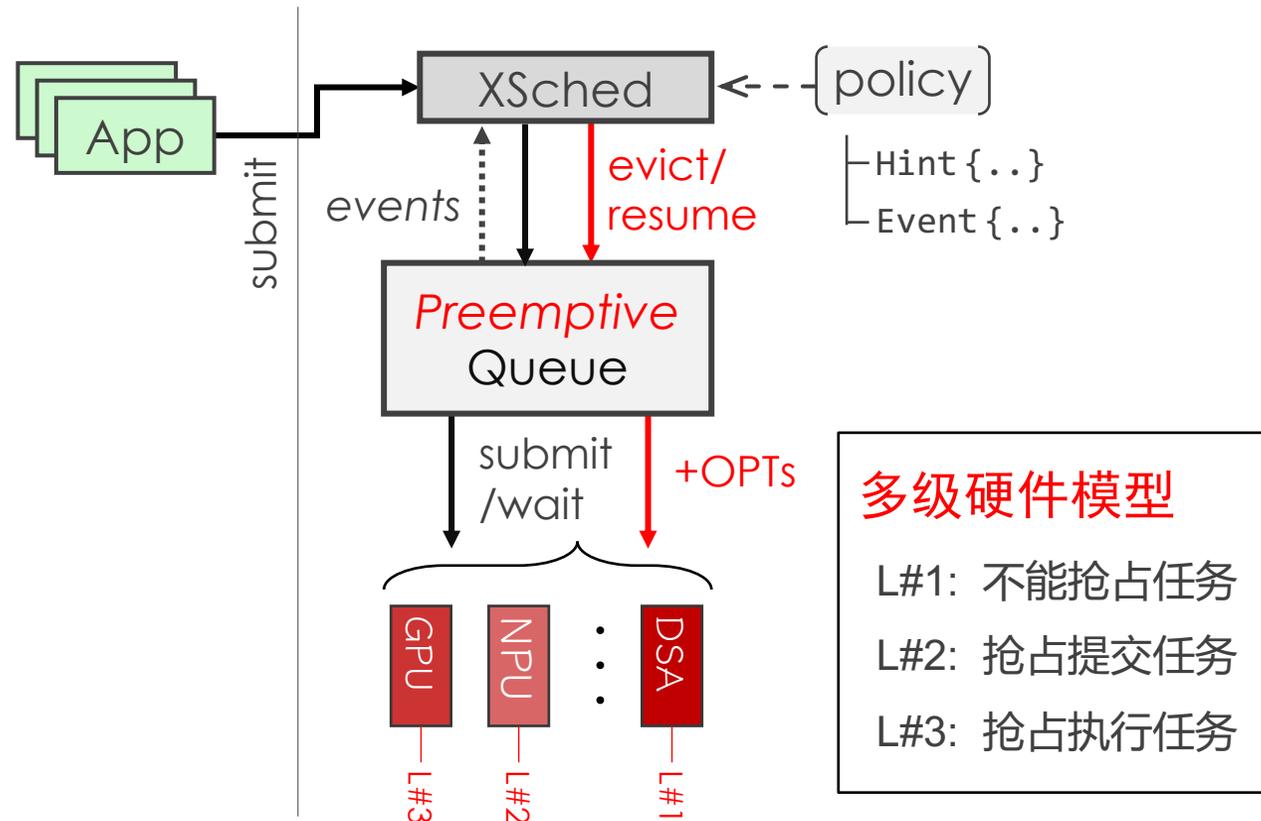
NPU: 预设命令/功能

× XPU硬件实现差异大

- 提交/等待 (基础)
- 中断
- 计算单元重置
- 内存刷新
- ...

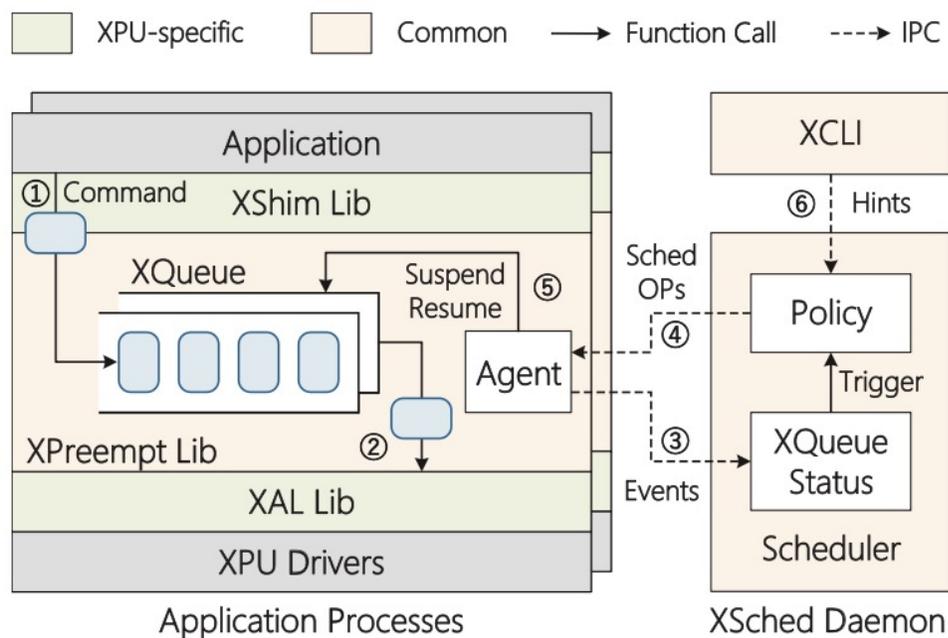
思路/方法

关键思路：基于能力的多级硬件模型



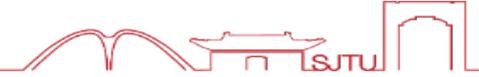
Preemptive Scheduling for Diverse XPU using Multi-level Hardware Model

- ▶ 支持各类XPU（类型、品牌、型号）
- ▶ 动态调用劫持、统一调度接口、灵活分级实现



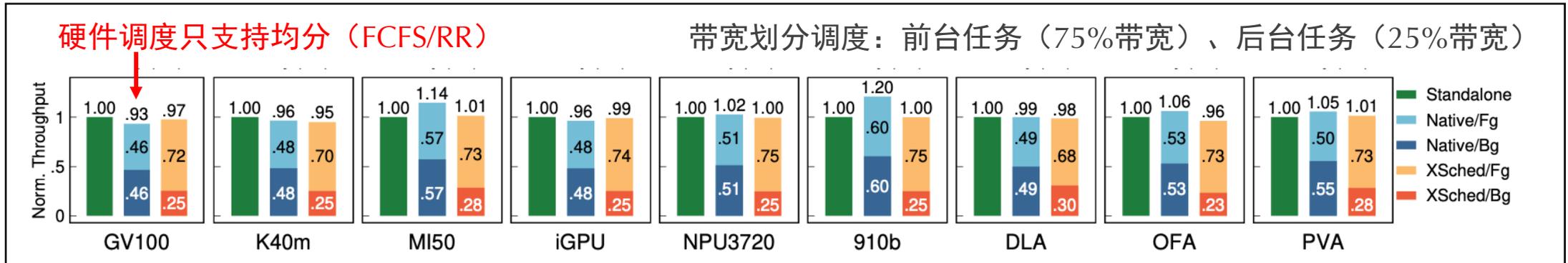
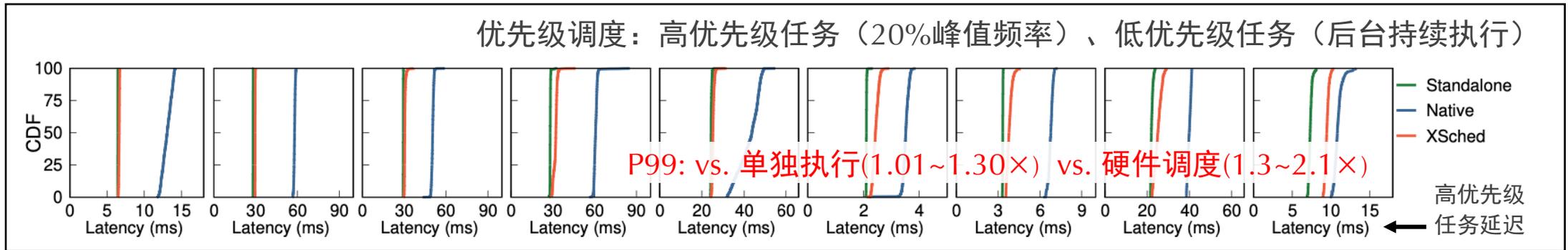
Hardware	Driver	XShim	Lv1	Lv2	Lv3
NVIDIA K40m NVIDIA GV100	CUDA	486	549	622 693	○ 265
AMD MI50	HIP	504	912	●	●
Intel Arc iGPU Intel NPU3720	LevelZero	456	301	● 240	● ○
Ascend 910b	ACL	177	291	●	○
NVIDIA DLA	CUDLA	213	232	○	○
NVIDIA OFA NVIDIA PVA	VPI	220	206	○ ○	○ ○

通用XPU多任务调度



Preemptive Scheduling for Diverse XPUs using Multi-level Hardware Model

- ▶ 支持各类XPU（类型、品牌、型号）
- ▶ 动态调用劫持、统一调度接口、灵活分级实现



应用和硬件的发展演进是系统软件研究的**原动力**

“应用需求”与“硬件能力”是系统软件研究的**重要抓手**

“赋能赋智”带来**算力外需求**，亟需基础系统软件的**关键支撑**

我们的一些初步探索——**异构算力硬件调度**

感谢！