上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

SJTU

L1-1. 什么是操作系统?

宋卓然

上海交通大学计算机系

songzhuoran@sjtu.edu.cn

我是谁？

上海交通大学计算机系

（先进体系结构实验室）

助理研究员

电信群楼3-125

https://songzhuoran.github.io

软件算法

AI推理

- 基于近似计算的AI推理加速算法 [ICCAD'18, TODAES'19]

AI训练

- AI训练加速算法 [DATE'19, submitted to ICCD'22]

- AI专用加速器 [ISCA'20, MICRO'20, DAC'22, TODAES'22, DATE'22, submitted to TCAD'22, MICRO'22]
- 基于GPU的神经网络加速器 [DAC'20]
- 面向高效神经网络的可重构加速器 [HPCC'20, ISPA'22]

AI领域专用加速器

- ReRAM容错机制 [TCAD'20]
- 基于ReRAM的神经网络加速器 [GLSVLSI'20, ISCAS'21, TCAD'22 , submitted to ICCD'22]

存算一体架构

硬件架构

先进体系结构实验室

实验室简要介绍：

实验室总体研究方向是：人工智能芯片架构设计，计算机体系结构，GPU、FPGA和ASIC硬件加速器及其软件技术开发，上层应用解决方案设计，包括数字图像识别、云游戏性能优化等。实验室与多家公司以及企业均有合作。

主要研究方向：

GPGPU和AI芯片体系结构设计

AI、游戏、VR、AR算法加速

基于新型器件的体系结构等

先进体系结构实验室

实验室大厅

实验室机房（芯片制作、测试）

计算机体系结构研究国内领先，产业合作密切

长期居国内首位，世界20-30位

CSRankings: Computer Science Rankings

CSRankings is a metrics-based ranking of top computer science institutions around the world. **Click on a triangle** (▶) to expand areas or institutions. **Click on a name** to go to a faculty member's home page. **Click on a pie** (the alter a name or institution) to see their publication profile as a pie chart. **Click on a Google Scholar icon** (𝔊) to see publications, and **click on the DBLP logo** (ℓ) to go to a DBLP entry.

公认计算机体系结构领域4大顶会发表论文统计数据

国内排名

1	▶ Shanghai Jiao Tong University	9.5	13
2	▶ Tsinghua University	6.7	11
3	▶ Peking University	5.4	7

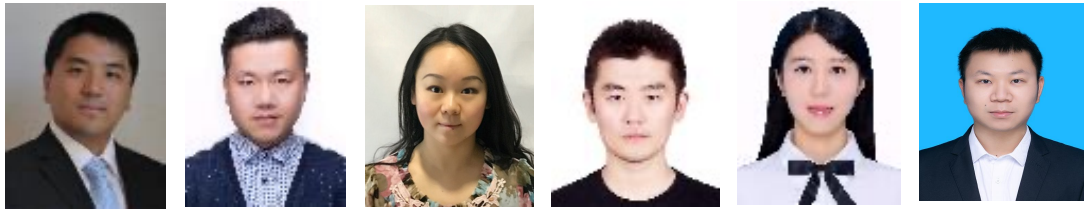
亚洲排名

1	▶ KAIST	17.2	9
2	▶ Seoul National University	12.0	8
3	▶ Shanghai Jiao Tong University	9.5	13
4	▶ Tsinghua University	6.7	11

国际排名

26	▶ University of Texas at Austin	9.6	7
27	▶ Shanghai Jiao Tong University	9.5	13
27	▶ University of Rochester	9.5	8

研究力量雄厚，高校-企业合作紧密



项目名称	负责人	经费(万)	华为部门
支持稀疏的AI 编译优化技术	蒋力	98	2012中软
端侧稀疏化深度学习神经网络训练框架	蒋力	46	
面向通信系统的存算一体实现研究	蒋力	138	2012 中央 研究院
基于ReRAM的高效可靠DNN加速器技术研究	蒋力	111	
基于AI应用场景的GPGPU架构研究	梁晓峣	278	
存储和计算融合的体系结构研究	梁晓峣	20	
D项目写书	梁晓峣	72	Cloud BU
基于MDC的高性能新型算法模型优化技术	梁晓峣	125	车BU
面向昇腾算子开发的技术	梁晓峣	51	2012 海思
Manchester项目	梁晓峣	235	
D芯片软硬件联合优化合作项目技术咨询	蒋力	26	2012媒体
低时延异构SoC总线通信评估与优化	蒋力	74	数通
跨Die互连架构通信评估与优化	叶瑶瑶	80	
近cache加速技术合作项目委托研发	蒋力	80	无线

多部门长期合作，近3年完成/在研**14项**，年均经费超**500万**

通用GPU架构

国内研究高性能、高能效的SIMD领域专用计算架构（GPGPU）的顶尖机构

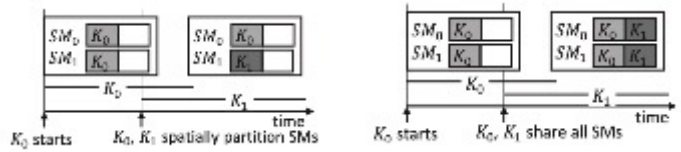
通用GPU架构与应用

- 计算机体系结构**4大顶级会议**上发表**10篇**论文
- 英伟达 Fermi, Kepler等主流GPU产品主架构师
- **两次入选**国际计算机体系结构年度最佳论文（**IEEE Micro TOP PICKS** from Computer Architecture Conferences），传统认定卓越研究成果的形式，代表在学术界和工业界最具影响力的研究，是**目前亚洲地区入选该奖项最多的研究人员**

- 面向GPU的实时图像处理系统与算法优化

已在某国产军用无人机型号上使用

同步多内核GPU

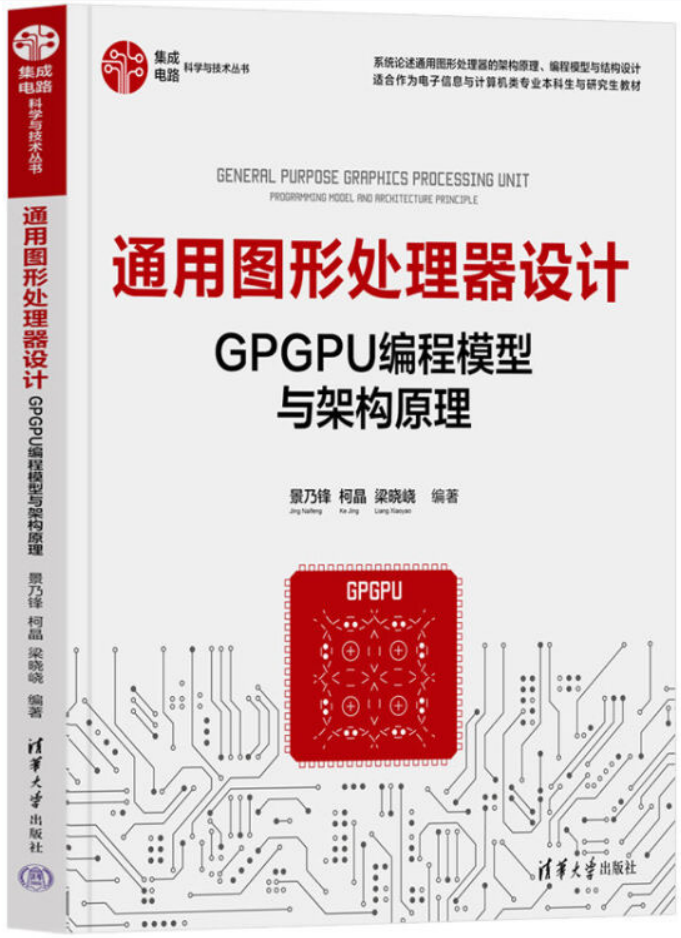


-- 发表于 **HPCA** 16 **Mutulu, Gupta等认可有效提升20%性能**

Onur Mutlu，Eth 顶尖存储架构专家，保持体系结构顶会论文记录

Rajesh K. Gupta，UCSD Qualcomm 荣誉教授，CS系主任

通用GPU架构



通用图形处理器设计
GPGPU编程模型与架构原理

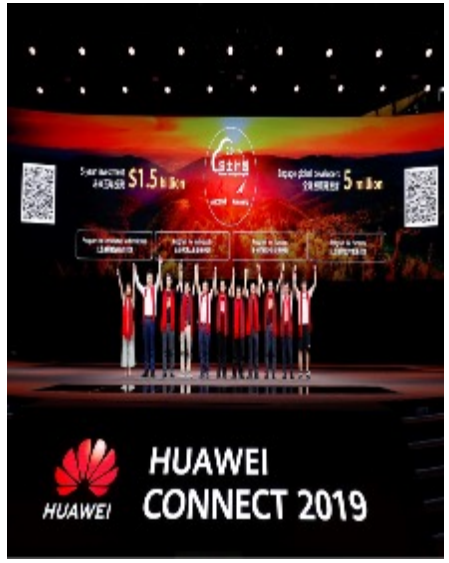
清华大学出版社，ISBN：9787302604648

国内第一本通用GPGPU芯片体系结构和设计的专业书籍。

以通用GPGPU芯片为基础平台进行展开，重点深入了芯片的架构设计原理，架构设计理念和程序优化技术，同时以结合AI应用展现性能优势和特有价值。

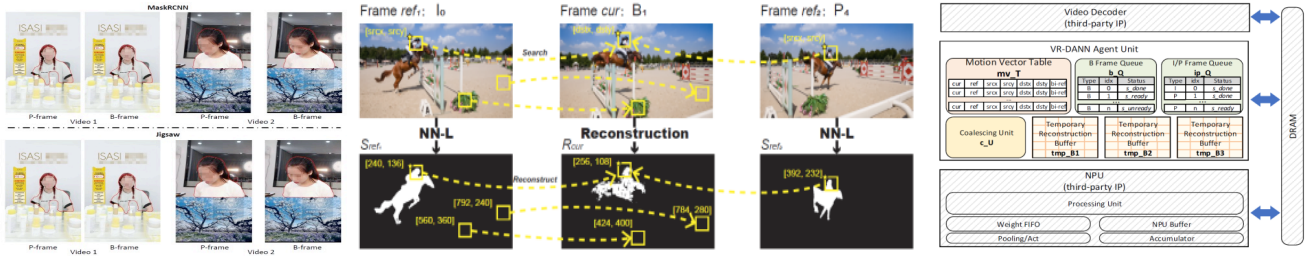
通用超大规模AI芯片架构设计

华为昇腾AI处理器架构研究和推广



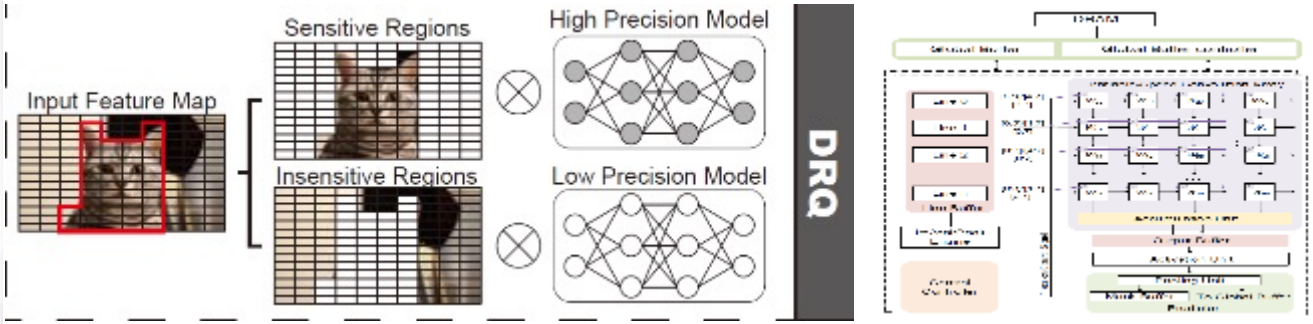
- 6个专利授权华为AI处理器研发，获得**华为年度最佳合作奖**
- 编著《昇腾AI处理器架构与编程》，已经成为**AI芯片领域流行的参考书**
- 共同**发起并深度参与华为沃土计划2.0**，为建立广泛生态贡献力量

紧耦合视频解码器的AI芯片架构



- 首次提出耦合视频解码器的AI芯片架构，精度基本不损失的前提下实现**2.5倍加速比**
- 原型系统**部署在阿里巴巴视频直播平台**，性能提升可观
- 论文发表在计算机体系结构顶级会议 [**MICRO**’ 20]

基于特征图特性的CNN量化方法

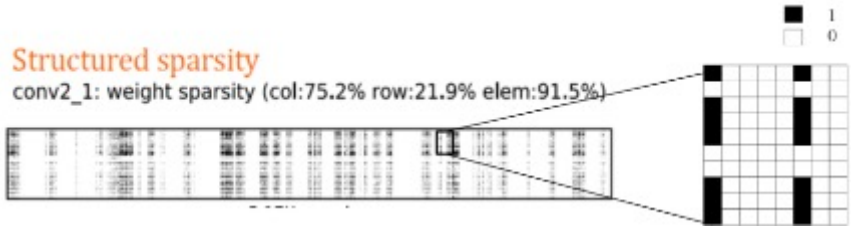


- 首次提出基于动态预测特征图敏感性的量化方法，实现**10倍以上加速比**
- 论文发表在计算机体系结构顶级会议[**ISCA**’ 20]

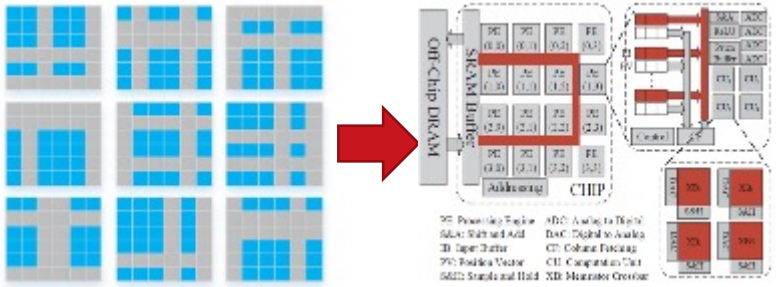
稀疏化存内计算架构（CiM）

- **关键问题：CiM架构数据与存储绑定，计算与字/列线绑定，无法利用随机稀疏性**
- **创新点：提出了结构稀疏性算法与数据流架构设计，CiM使DNN压缩**

首创存算一体架构DNN压缩技术



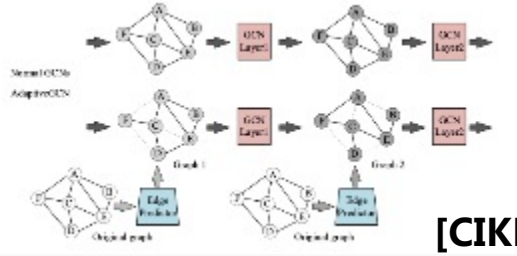
- 首次提出针对CiM的DNN**结构稀疏化压缩**算法，相同面积存算一体芯片实现**10倍DNN推理性能提升**
- 论文 [DATE18] 3年内**引用破百**



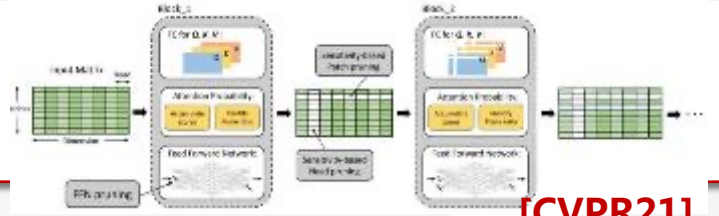
- 首次提出CiM**细粒度稀疏化架构**，进一步提升**20倍性能**
- 发表在CCF-A类会议[**DAC'20**]
- 经过华为产品线测试应用

稀疏图时序

- **负载均衡的GCN动态剪边压缩算法，提升GPU平台40%性能**，发表于**信息检索顶会CIKM**
- **利用注意力预测无信息Head/Patch的动态结构化剪枝，减少Vision Transformer40%计算**



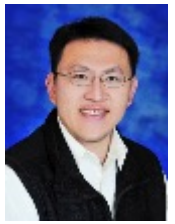
[CIKM21]



[CVPR21]

原创性获业界专家认可

"inspired by literature that exploits sparsity in DNNs"
--**ACM/IEEE/AAAS Fellow UIUC讲席教授 Josep Torrellas [MICRO20]**



IEEE Fellow, NTHU教授/台积电主任 张孟凡[TCAD21]

properties of CIM macros. RECOM [32] was the first CIM-based accelerator to support sparse DNN processing. Through appropriate design of the corresponding circuit, the weights which are all zero in the same rows or columns can be skipped. ...

第一个支持稀疏DNN的存内计算架构

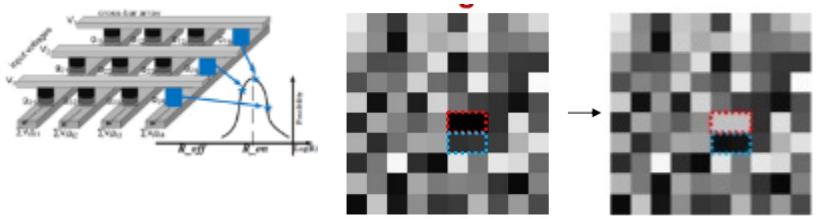
ReCom is the first to exploit the sparsity of neural networks for ReRAM-based accelerators [11]. It explores the weight sparsity only

IEEE Fellow伊利诺伊理工大学CS前系主任 孙贤和[ICS21]

软硬协同CiM可靠性提升技术

- **关键问题：忆阻器器件工艺偏差与缺陷严重，导致计算操作数误差，引起神经网络的精度大幅受损**
- **创新点：利用神经网络近似算法特性，提出编码映射，与模型重训练映射主动规避缺陷器件**

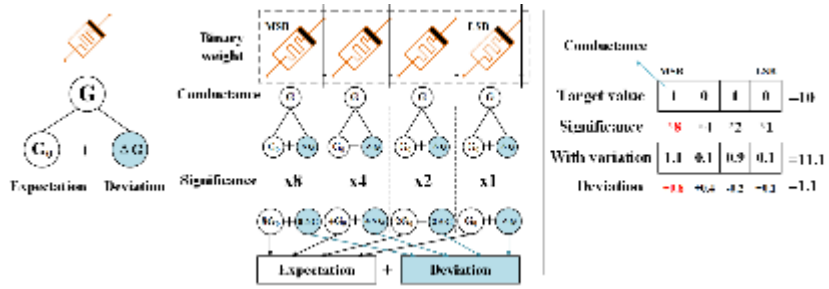
首创DNN重训练容错技术



- 提出了二分旁路电流测试算法，**读写测试次数从N减少到log_2N，减少40%成本**
- 首次提出**重新训练神经网络改变权重大小主动规避忆阻器缺陷与偏差**，提升神经网络**精度40%，发表在EDA顶级会议DAC**
- 利用bit级冗余资源的软硬协同容错机制，使**深度卷积神经网络精度损失在1%以内，发表在EDA顶级期刊TCAD**
- 使实验室级忆阻器AI芯片走向演示级

论文引用破百，在当期会议论文集中排名第二

首创随机均匀编码映射法

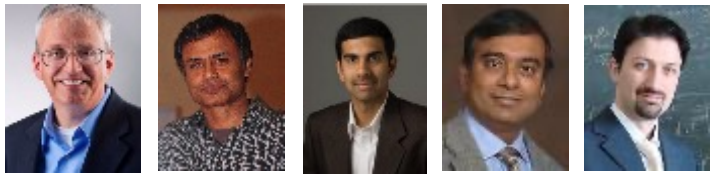


- 首创**随机均匀编码及映射方法**，有效规避忆阻器工艺偏差
- **首次在大数据集深度神经网络中保持精度损失1%**
- **发表3篇论文系统性阐述这一机制[DATE20,21]，以及EDA顶级期刊TCAD，其中一篇获得最佳论文提名**

北京大学黄如院士认为本方法在大数据集DNN也能**有效抵抗忆阻器工艺偏差**

In this case, to make the best use of the coding flexibility brought by stochastic coding and find the optimal mapping way, a variation-aware optimal mapping has been proposed, which can greatly reduce the weight variations, even for deep neural networks with large datasets. More details can be found in Ref. [33].

原创性获业界专家认可



美国工程院院士Bill Dally，杜克大学讲席教授 Chakrabarty，**西北大学ECE主任** Memik，**普渡大学讲席教授**Raghunathan **12篇论文**引用并作出改进

清华大学集成电路学院院长吴华强认为本方法**有效容忍忆阻器故障**

to solve the over forming problem [82]. The accuracy degradation caused by a stuck device can be solved by optimizing the weight mapping method, such as introducing redundancy RRAM rows [83] or remapping the synaptic weight considering the distribution of stuck devices [84].

IEEE&IBM Fellow Eleftheriou在**Nature**中认可**“首次用训练解决器件偏差和缺陷”**

13篇Nature作者麻省大学杨建华评价本方法**决定了神经网络训练性能**

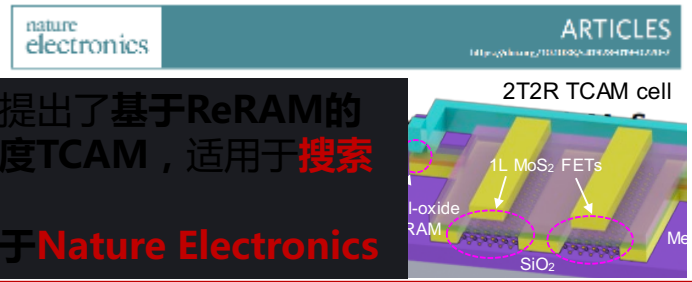
香港科大工程学院院长Tim Cheng认为本方法**有效保证神经网络精度**

which greatly alleviates the accuracy degradation. The authors in [50] proposed a software and hardware co-design methodology to effectively preserve the classification accuracy of CNN with few on-device training iterations on RRAM-crossbars. Kim et al. [51] proposed an algorithm and

存搜一体架构 (Search-in-Memory)

存搜器件

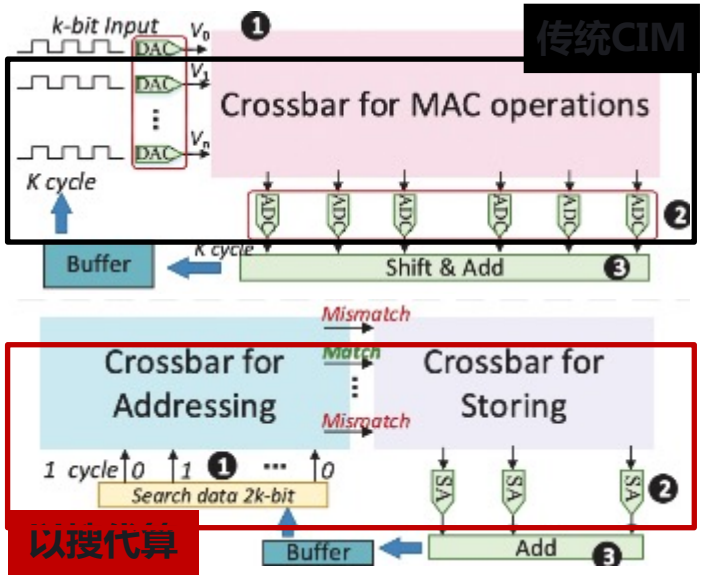
首次提出了**基于ReRAM的高密度TCAM**，适用于**搜索匹配**
发表于**Nature Electronics**



忆阻器之父，台积电首席科学家Philip Wong在综述文章中评价：这一TCAM**搜索能力高、能效高**，3D集成**非常适合模拟人脑记忆系统**

potential catastrophic forgetting issues in a learning system. A hardware AM realization that leverages the integration of HfO_x RRAM and MoS_2 FETs has been reported [27]. In this work, low leakage and robust current control lead to high search capacity and energy efficiency. Owing to low temperature fabrication, the combination of RRAM and MoS_2 can be further integrated into a high-rise monolithic 3D system, approaching a closer emulation of human memories in terms of ultra-dense connectivity for learning and memory functionalities.

通用GEMM的存搜一体架构



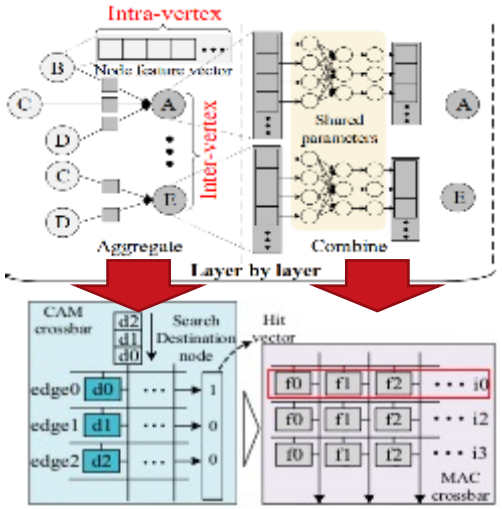
- 首次提出**GEMM 存搜一体架构**，**TCAM以搜代算（乘法），无ADC；**
- **模型无关，TCAM容量只受量化影响**

[GLSVLSI21]

图卷积存搜一体架构

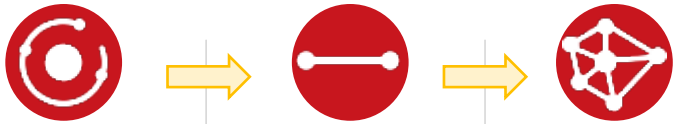
低效
图算法聚合算子用**稀疏邻接矩阵**

高效
用**稠密邻接表**配合CAM搜索



- 首次提出**图算法的存搜一体架构**，用**TCAM邻接表**快速搜索相邻聚合节点
- **74倍性能提升（vs. GPU）**
- 发表于**EDA顶级会议DAC21**

神经形态计算研究



	数据点	时间线	超维空间
算法	人工神经网络（ANN） 稀疏化、量化	脉冲神经网络（SNN） 学习方法	记忆启发超维计算（HDC）编码训练
架构	存算一体CiM	存算一体CiM+存搜一体SiM	

降低延迟 | **节省能耗** | **简化计算**

CiM
在投
- SiM哈希
- SiM 图像检索
- PIM映射搜索

SNN
- SNN转换器[AAAI22]
- 基于SIM的SNN[FoN]

HDC
- PIM HDC
- HDC编码
- HDC数据库

课程信息

- 助教：齐春宇；邮箱：qichunyu@sjtu.edu.cn
- 课程ppt
  - https://songzhuoran.github.io
  - Canvas
- 课程书本
  - Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne, “Operating System Concepts”, 9th Edition, John Wiley & Sons, Inc.

课程考核

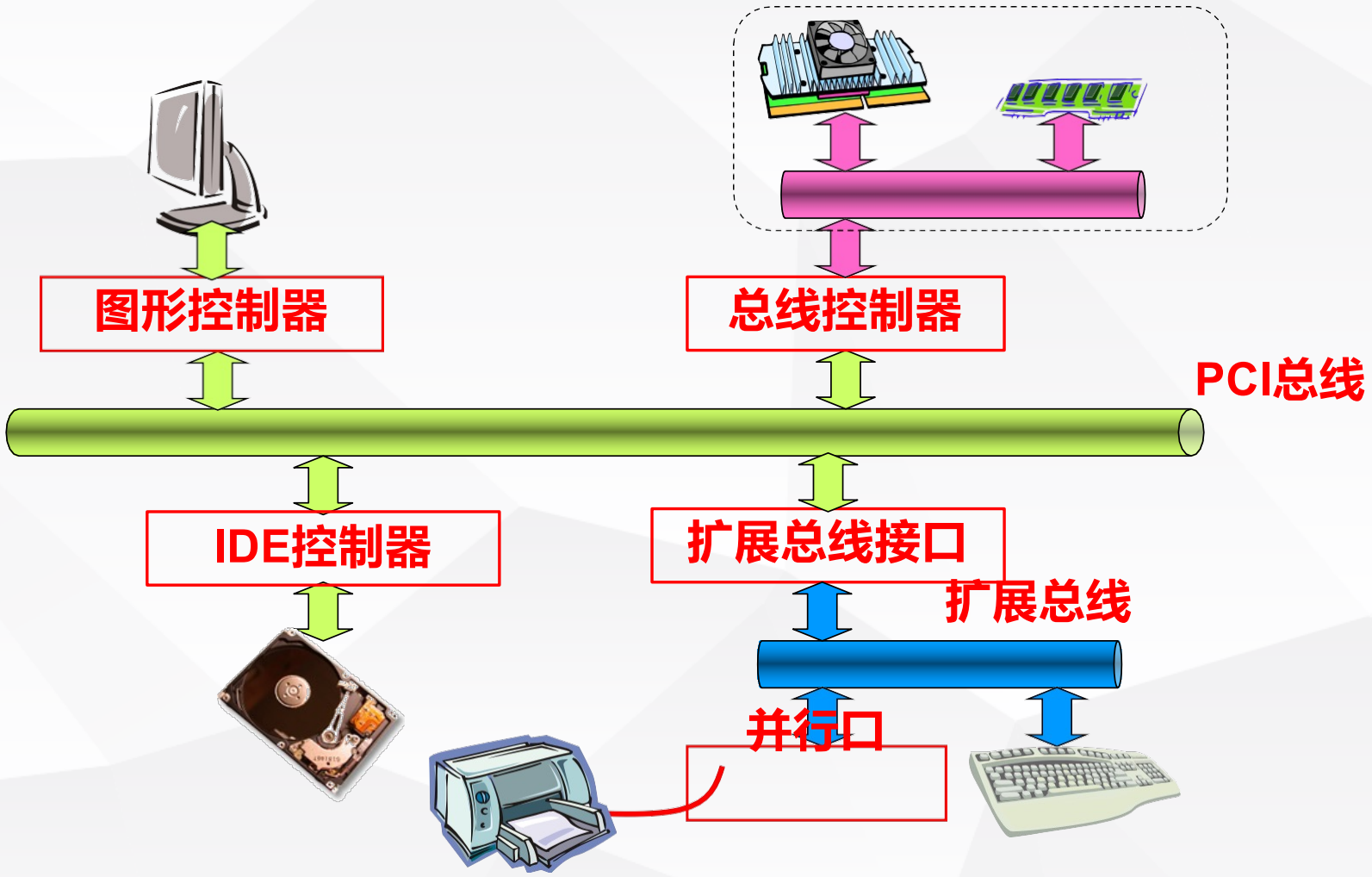
- 分数比例
  - 课堂作业 20%
  - 课后作业 20%
  - 课堂演讲 20%
  - 考试 40%
- 课堂演讲
  - 3人一组
  - 每组30分钟演讲
  - 主题为面向新算法（AI算法）、新硬件（GPU A100、V100、tensor core）的操作系统设计
    - 参考会议包括： USENIX ATC、HPCA、ASPLOS、OSDI、NSDI、EuroSys
    - 年份包括2020-2023

学术诚信

- 不要与其他同学共享你的作业/程序
- 如果需要用到别人发表的工作中的内容，请适当引用

NO PLAGIARISM!

先给大家看副图

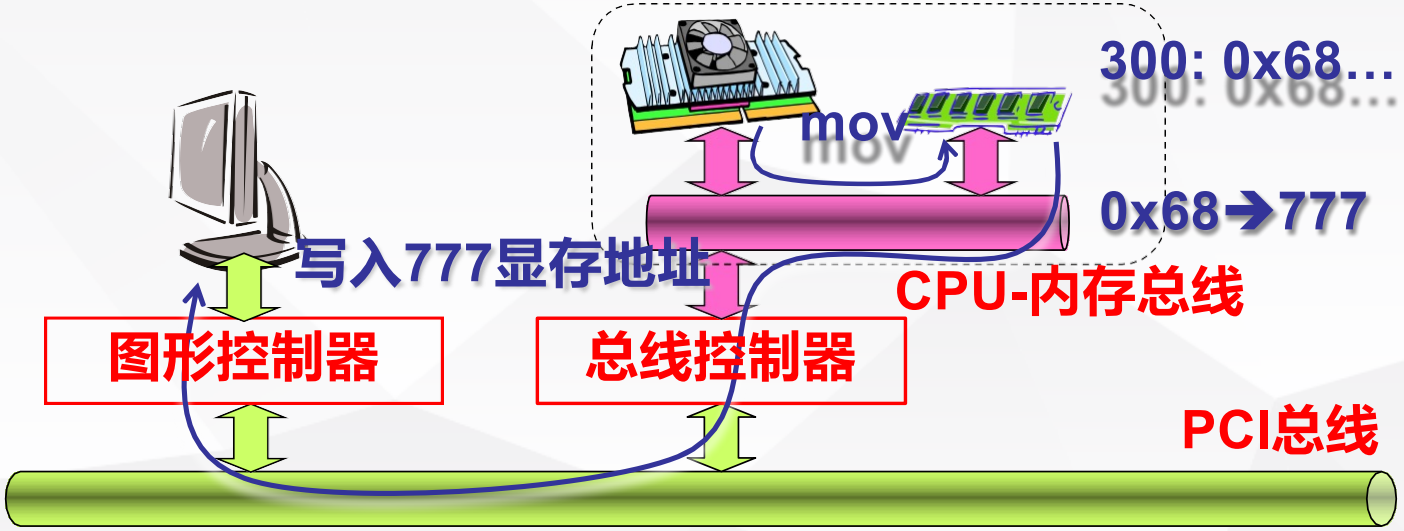


是计算机

- 计算机
- 用一句话说一说计算机 专业要干什么?
- 计算机有了那就解决这个问题吧：屏幕上输出 “hello!”

用计算机帮助人们解决一些实际问题

- 计算机有了那就解决这个问题吧：屏幕上输出“hello!”

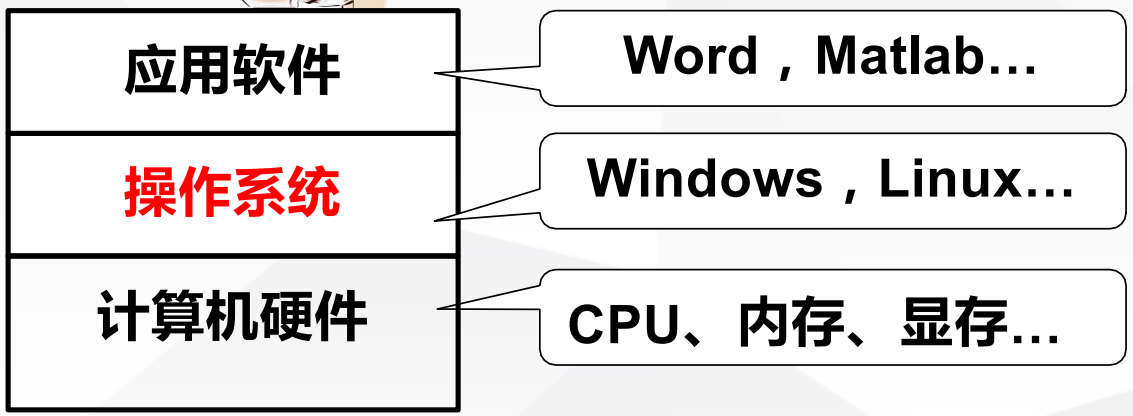


是计算机，更确切的说是计算机硬件

- 这个东西是计算机硬件，有人戏称为裸机

看来需要给计算机硬件穿上衣服啊！

printf（“hello!”）



在穿上了衣服的计算机上再次：屏幕上输出

“hello!”

操作系统是在硬件和应用之间的软件层

是计算机硬件和应用之间的一层软件

- 方便我们使用硬件，如使用显存...
- 高效的使用硬件，如开多个终端(窗口)

从 Hello World 说起

```c
#include <stdio.h>

int main()
{
    printf("Hello World!\n");
    return 0;
}
```

```
bash$ gcc hello.c -o hello

# 运行一个hello world程序
bash$./hello
Hello World!

# 同时启动两个hello world程序
bash$ ./hello & ./hello
[1] 144
Hello World!
Hello World!
[1]+ Done        ./hello
```

运行Hello时，操作系统的作用是？

操作系统考虑的一些问题

- hello 这个可执行文件存储在什么位置?是如何存储的?
- hello 这个可执行文件是如何加载到 CPU 中运行?
- hello 这个可执行文件是如何将"Hello World!"这行字输出到屏幕?
- 两个hello 程序同时运行的过程中如何在一个 CPU 中运行?

操作系统需要：1、服务应用；2、管理应用

操作系统为应用提供的服务

为应用提供计算资源的抽象

- CPU:进程/线程，数量不受物理CPU的限制
- 内存:虚拟内存，大小不受物理内存的限制
- I/O设备:将各种设备统一抽象为文件，提供统一接口

为应用提供线程间的同步

- 应用可以实现自己的同步原语(如spinlock)
- 操作系统提供了更高效的同步原语(与线程切换配合, 如pthread_mutex)

为应用提供进程间的通信

- 应用可以利用网络进行进程间通信(如loopback设备)
- 操作系统提供了更高效的本地通信机制(具有更丰富的语义，如pipe)

操作系统对应用的管理

生命周期的管理

- 应用的加载、迁移、销毁等操作

计算资源的分配

- CPU:线程的调度机制
- 内存:物理内存的分配
- I/O设备:设备的复用与分配

安全与隔离

- 应用程序内部:访问控制机制
- 应用程序之间:隔离机制，包括错误隔离和性能隔离

操作系统 = 管理 + 服务

管理和服务的目标有可能存在冲突

- 服务的目标:单个应用的运行效率最大化
- 管理的目标:系统的资源整体利用率最大化
- 例:单纯强调公平性的调度策略往往资源利用率低
  - 如细粒度的round-robin导致大量的上下文切换

操作系统的功能:管理

避免一个流氓应用独占所有资源

方法-1:每10ms发生一个时钟中断(时间片)

- 调度器决定下一个要运行的任务

方法-2:可通过信号等打断当前任务执行

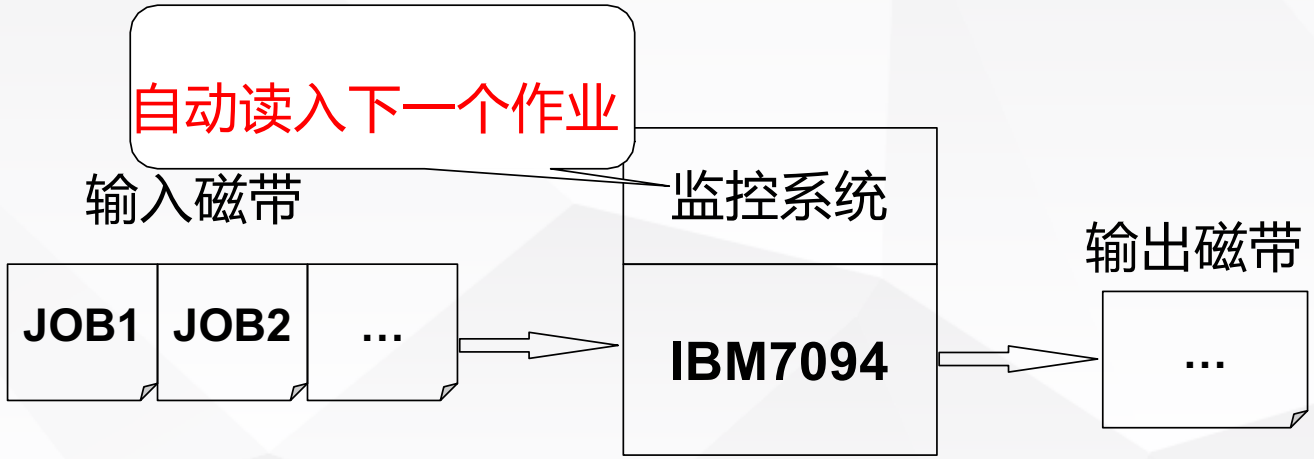
- 如:kill -9 1951

rogue.c

```
int main () {
    while (1);
}
```

培根：读史使人明智　　操作系统的简史

- **(1955-1965)**计算机非常昂贵，上古神机**IBM7094**，造价在**250**万美元以上
- 计算机使用原则：只专注于计算
- 批处理操作系统(Batch system)

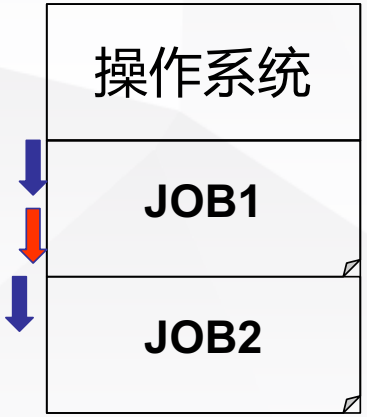


典型代表：**IBSYS**监控系统

从IBSYS到OS/360(1965-1980)

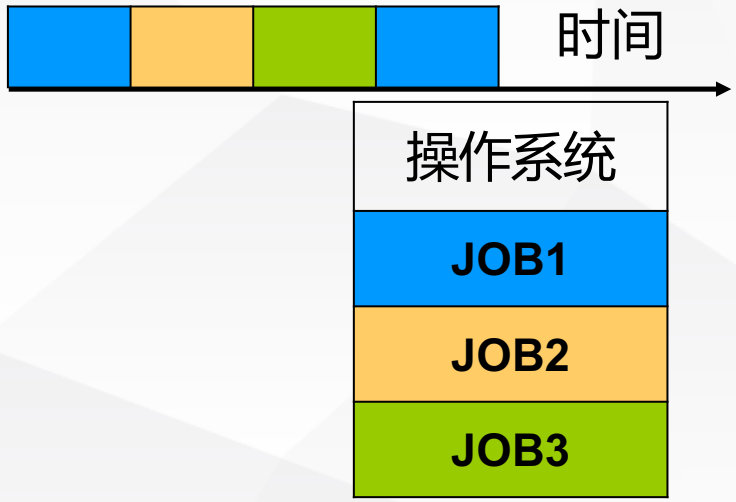
- 计算机开始进入多个行业：科学计算(IBM 7094)，银行(IBM 1401)
  - 需要让一台计算机干多种事
  - 多道程序(multiprogramming)
  - 作业之间的切换和调度成为核心：因为既有IO任务， 又有计算任务，需要让CPU忙碌
  - 典型代表：IBM OS/360(360表示全方 位服务) ，开发周期5000个人年

多进程结构和进程管理概念萌芽!



从OS/360到MULTICS(1965-1980)

- 计算机进入多个行业，使用人数增加
  - 如果每个人启动一个作业，作业之间快速切换
  - 分时系统(timesharing)
  - 代表：MIT MULTICS (MULTiplexed Information and Computer Service)
  - 核心仍然是任务切换，但是资源复用的思想对操作系统影响很大，虚拟内存就是一种复用



从MULTICS到UNIX(1980-1990)

- 小型化计算机出现，PDP-1每台售价120,000美元，不足7094的5%
  - 越来越多的个人可以使用计算机
  - 1969年：贝尔实验室的Ken Thompson、 Dennis Ritchi等在一 台没人使用的PDP-7上开发一个简化MULTICS，就是后来的UNIX
  - UNIX是一个简化的MULTICS，核 心概念差不多，但更灵活和成功

Ken Thompson Dennis Ritchi

从UNIX到Linux(1990-2000)

- 1981，IBM推出IBM PC；个人计算机开始普及
  - 很多人可以用计算机并接触UNIX
  - 1987年Andrew Tanenbaum发布了 MINIX(非常类似UNIX)用于教学
  - Linus Torvalds在386sx兼容微机上学 习minix，作出小Linux于1991年发布
  - 1994年，Linux 1.0发布并采用GPL协 议，1998年以后互联网世界里展开了 一场历史性的Linux产业化运动

IBSYS->OS/360->MULTICS->Unix->Linux

- 历史使人明智

核心思想、技术

- 作为管理者，操作系统要让多个程序合理推进，就是进程管理
- 用户通过执行程序来使用计算机(吻合 冯诺依曼的思想)
- 多进程(用户)推进时需要内存复用等等

软件实现

- 对于操作系统，实现很重要OS/360->UNIX
- 需要真正的群体智慧 UNIX->Linux

操作系统家庭

- UNIX-family: BSD(Berkeley Software Distribution), System-V, GNU/Linux, MINIX, Nachos, OS X, iOS
- BSD-family: FreeBSD, NetBSD, OpenBSD
- System-V-family: AIX, HP-UX, IRIX, Solaris
- Linux-family: Red Hat, Debian, Ubuntu, Fedora, openSUSE, Linux Mint, Google's Android, WebOS, Meego
- MS-DOS, Microsoft Windows, Windows Mobile, Win-CE, WP8
- AmigaOS
- Tiny-OS, LynxOS, QNX, VxWorks