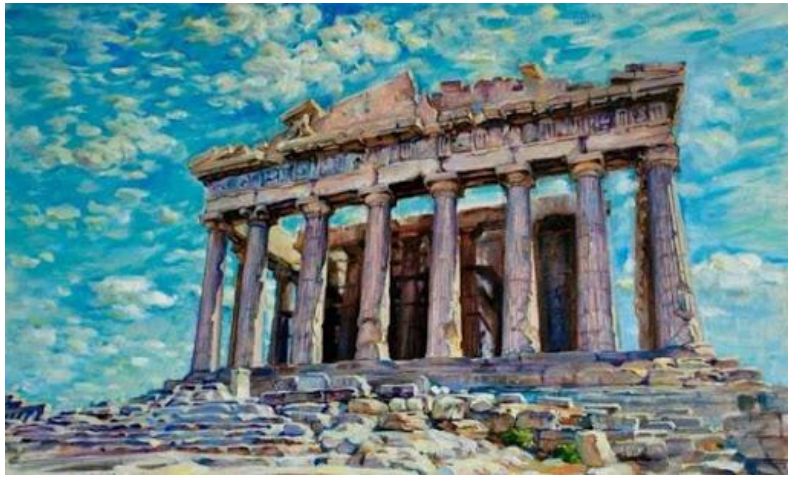
嵌入式系统联谊会
www.esbf.org

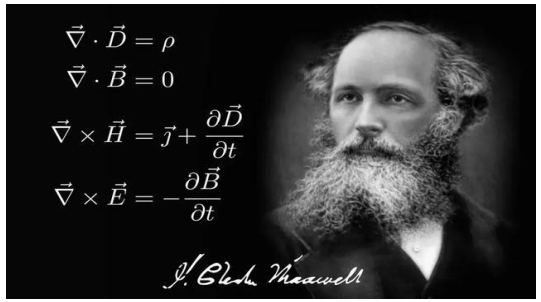
AI时代下的嵌入式操作系统研究探索

尹德帅 北京理工大学 博士
18611172832（微信同号）

探索

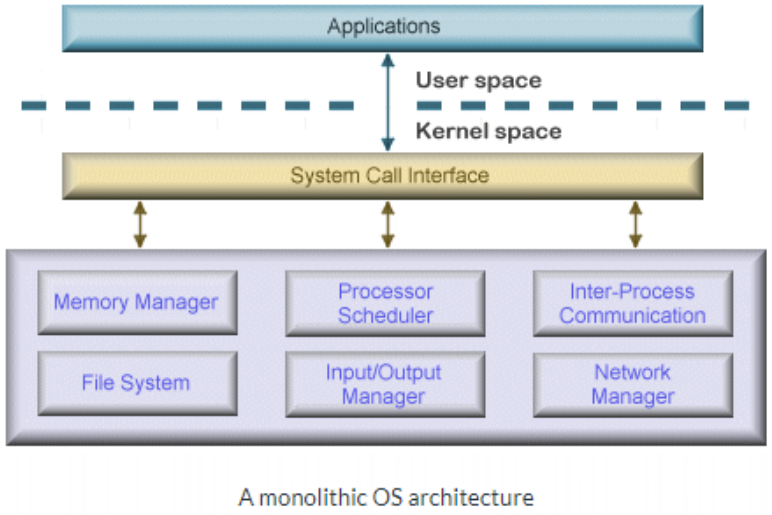


科学研究的四类范式

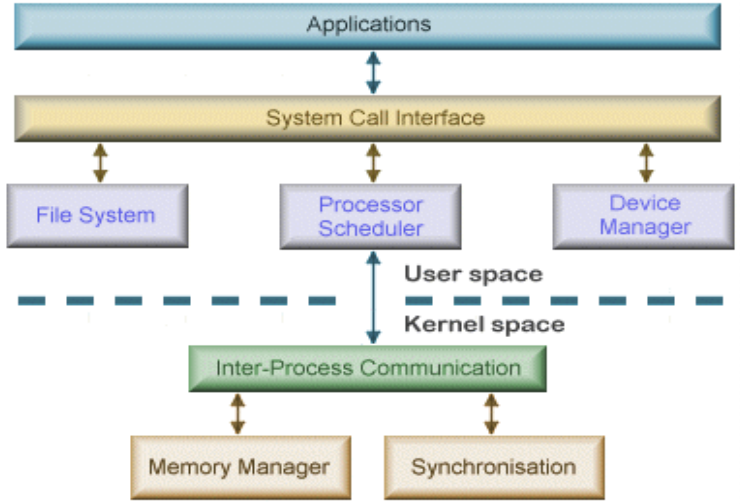
科学 第一范式	科学 第二范式	科学 第三范式	科学 第四范式
实验思维-科学归纳	逻辑思维-模型推演	计算思维-仿真模拟	数据思维-数据密集科学
1000年前	数百年前	几十年前	2007年后
• 对自然现象的描述论证 • 对自然现象进行系统归类	• 采用建模方式 • 由特殊到一般进行推演	• 用计算方法模拟复杂现象 • 科学数据可以用模拟的方法获得	• 与大数据密切相关 • **由探索因果关系到关注相关关系**
钻木取火	麦克斯韦方程	天气预报	AI进入高速发展期
			

操作系统架构及研究内容

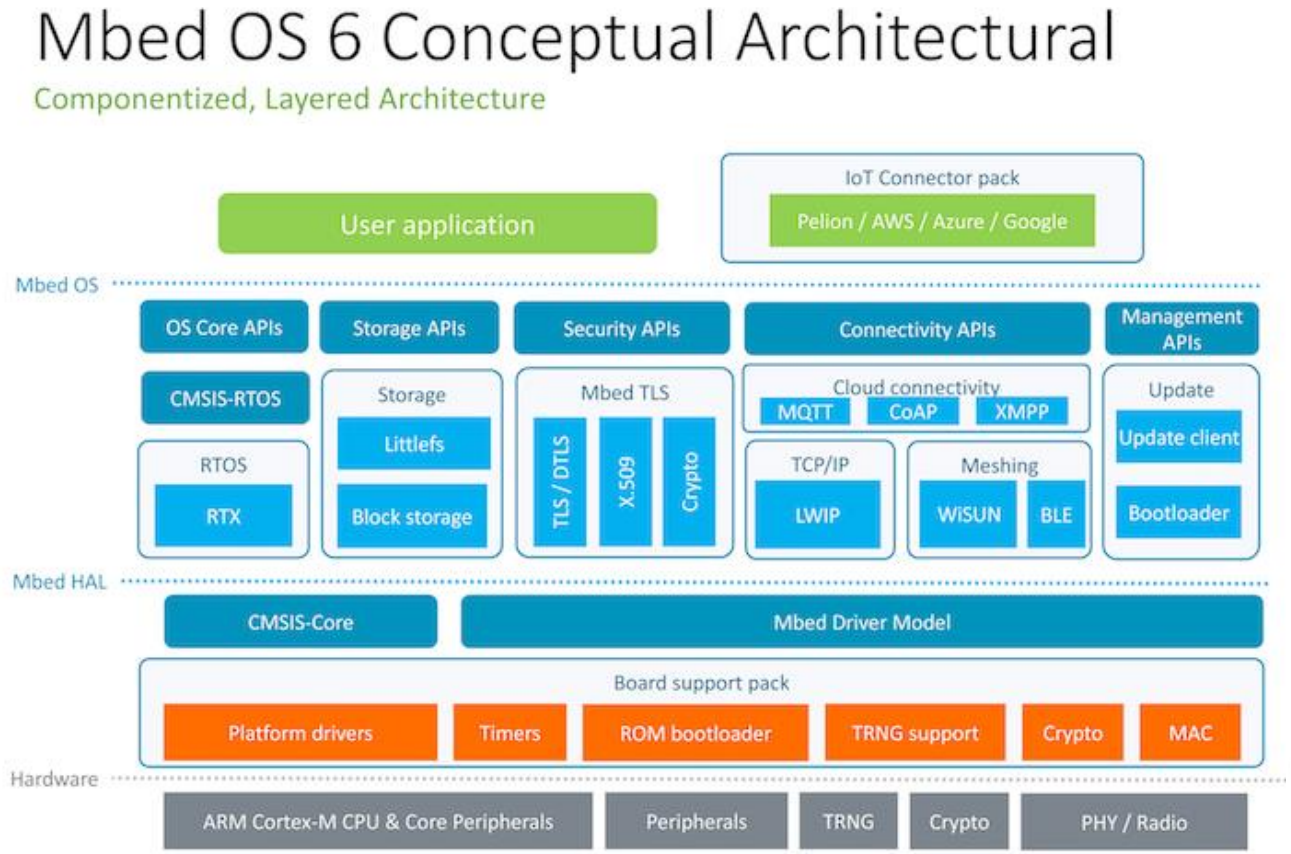
- OS方面的研究，包括内存管理、进程管理、进程间通信、文件系统管理、网络管理、中断处理等等。
- 针对具体应用场景或者业务相关的研究。



A monolithic OS architecture



A microkernel OS architecture



操作系统架构及研究内容

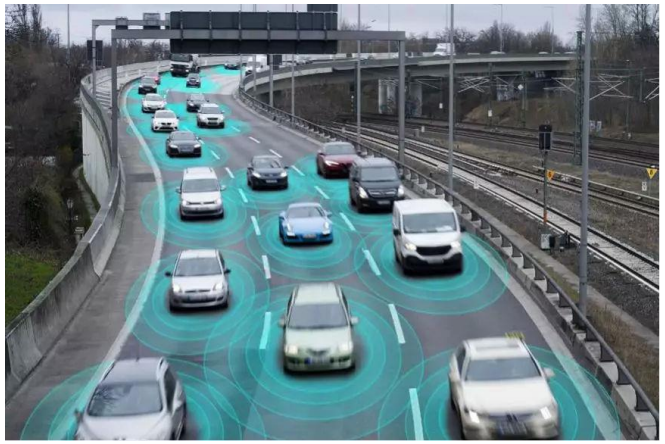
Discussed Features	This Review	Hahm, (2015) [25]	Dong, (2010) [26]	Farooq, (2011) [27]	Chandra, (2016) [28]	Gaur, (2015) [29]
Supported Devices	✓	✓	×	×	×	×
Architectures	✓	×	✓	✓	✓	✓
Scheduling Policies	✓	✓	✓	✓	✓	✓
Real-time Support	✓	×	✓	✓	✓	×
Scheduling Algorithms	✓	×	×	×	×	×
Programming Models	✓	×	×	×	×	✓
Programming Languages	✓	×	✓	×	×	✓
Network Stack	✓	✓	✓	×	✓	✓
Protocols	✓	×	×	✓	×	✓
Reprogramming Techniques	✓	×	✓	×	×	×
Memory Management	✓	×	×	✓	×	×
Energy Efficiency	✓	×	×	×	×	×
Simulators	✓	×	×	×	✓	×
Licenses	✓	✓	×	×	×	×
New Version	✓	✓	×	×	×	×
Documentation	✓	✓	✓	×	×	×
Shell	✓	×	×	×	×	×
Database	✓	×	×	×	×	×
Testing	✓	✓	×	×	×	×
Debugging	✓	✓	✓	×	×	×
Open/Close Standard	✓	✓	×	×	×	×
Code Development	✓	✓	×	×	×	×
Application	✓	×	×	×	×	×
Challenges	✓	×	×	×	×	×

OS边缘计算架构支持 IoT需要AI赋能

1. IoT应用：森林防火、边境监控、智能家居、航空诊断
2. AI能力：图像识别，音视频分析

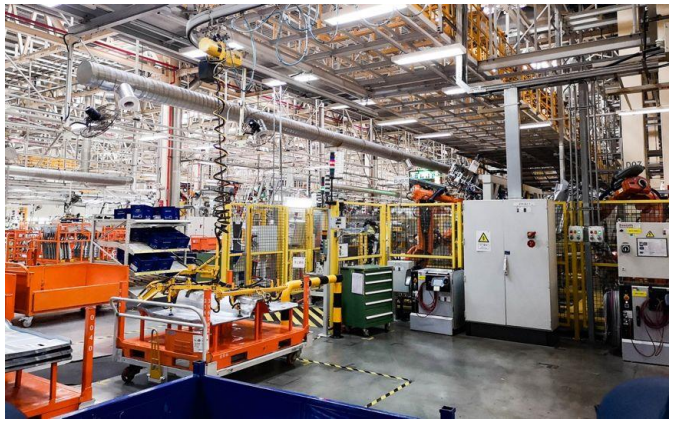
森林防火无人机巡逻

- 目标检测
- 区域识别
- 暗火辨识



高速除草机器人、自动驾驶

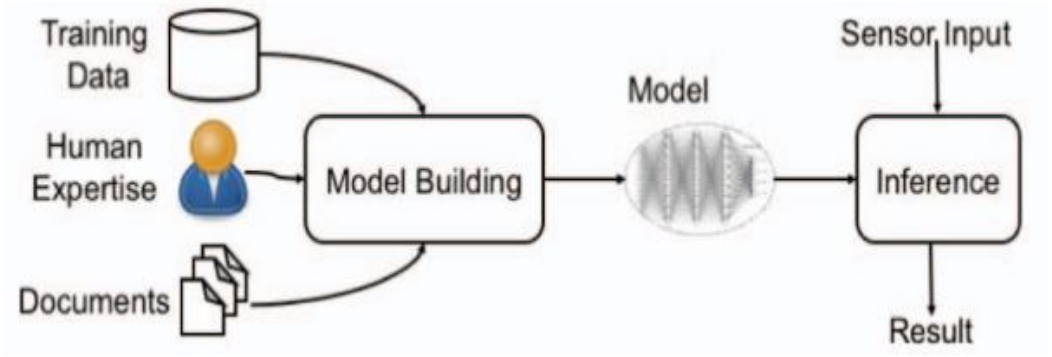
- 高速、精准图像识别



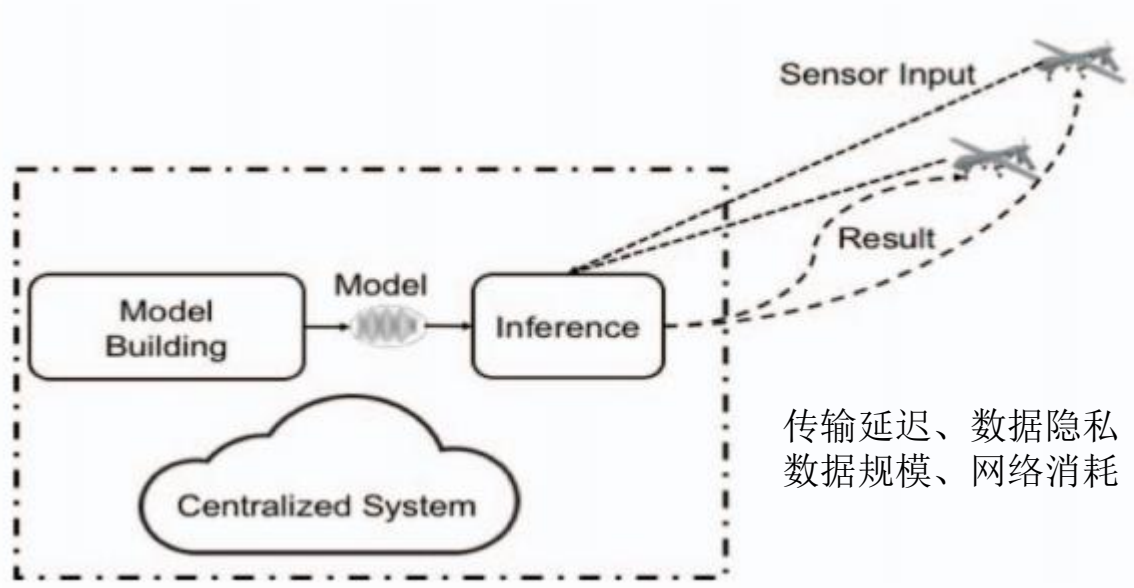
异响检测

- 实时异响检测

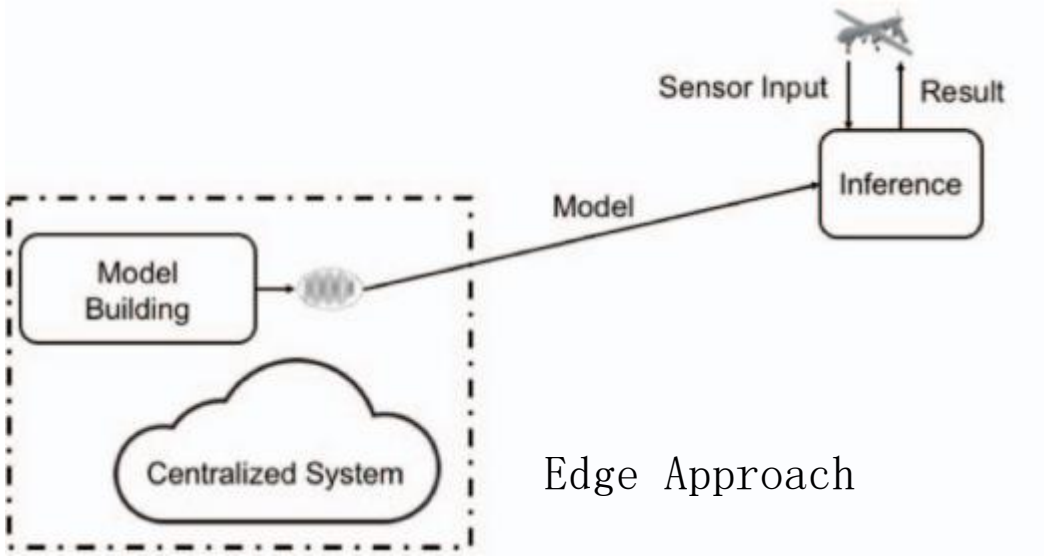
OS边缘计算架构支持 AI in IoT



Abstracted AI approach



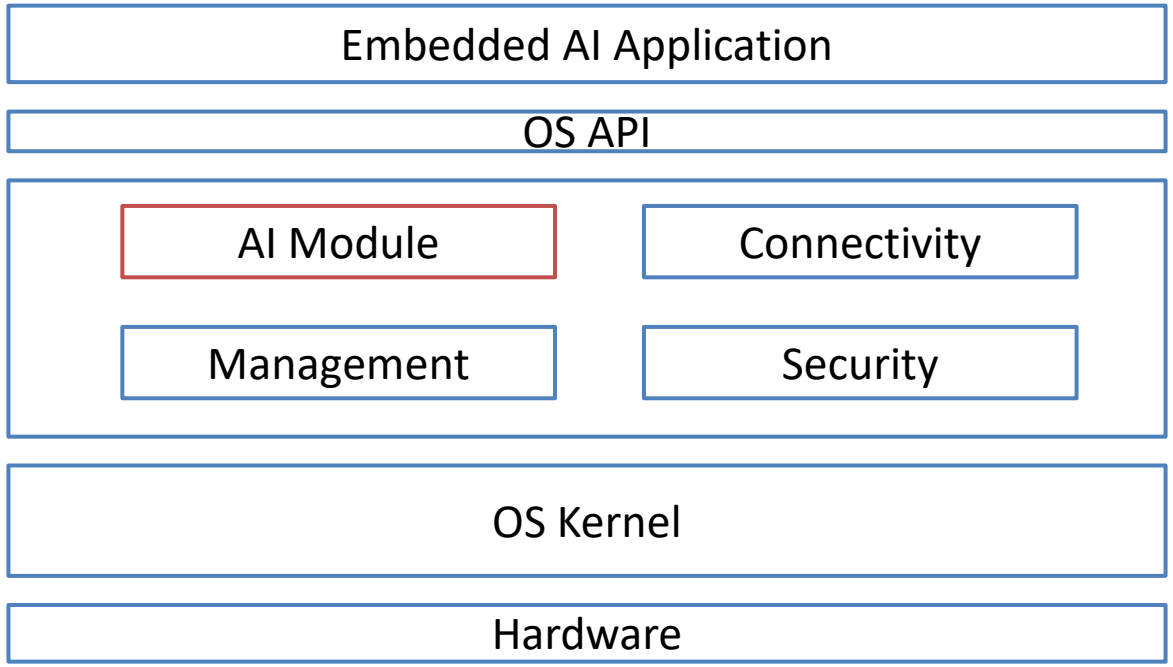
Centralized Approach

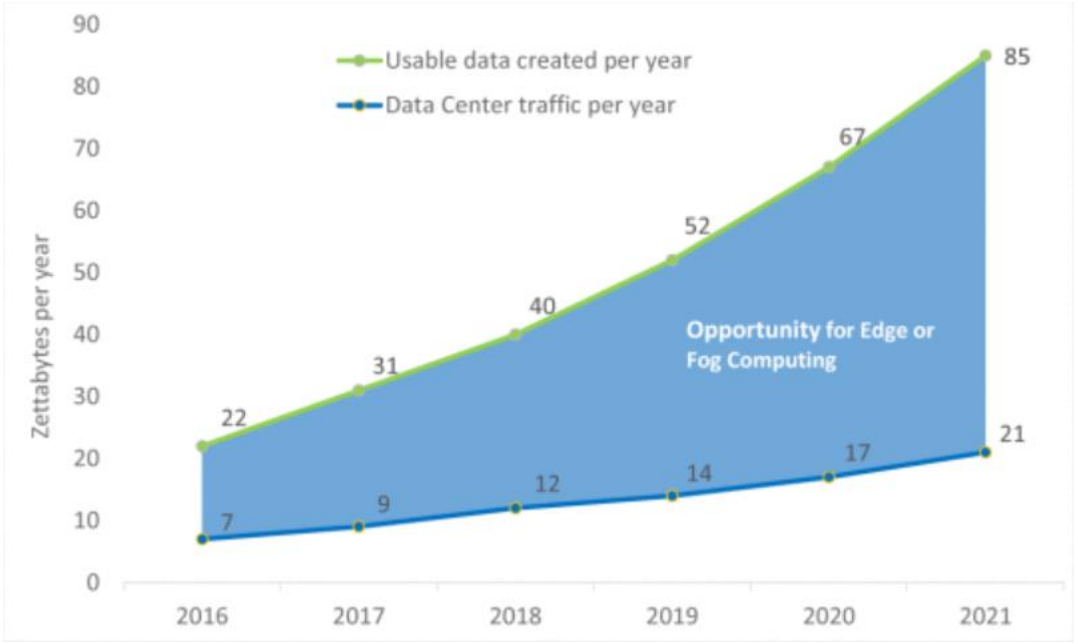


Edge Approach

OS边缘计算架构支持

1. 提升带宽要求应用效率
2. 降低时间敏感应用延迟
3. 确保数据隐私应用可信





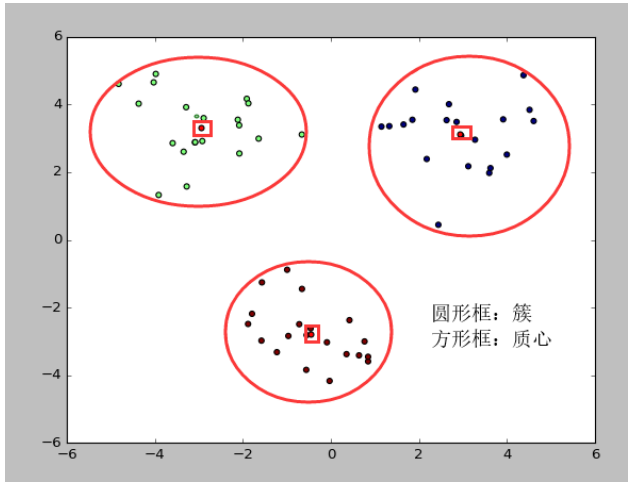
The gap between data created by users/things and data processed by Cloud. (source: Cisco Global Cloud Index)

操作系统研究方向与探索 任务调度业务个性化

要解决的问题

- 操作系统运行在不同的应用场景对任务调度的需求不同
- 操作系统任务调度算法针对不同业务进行个性化适应问题

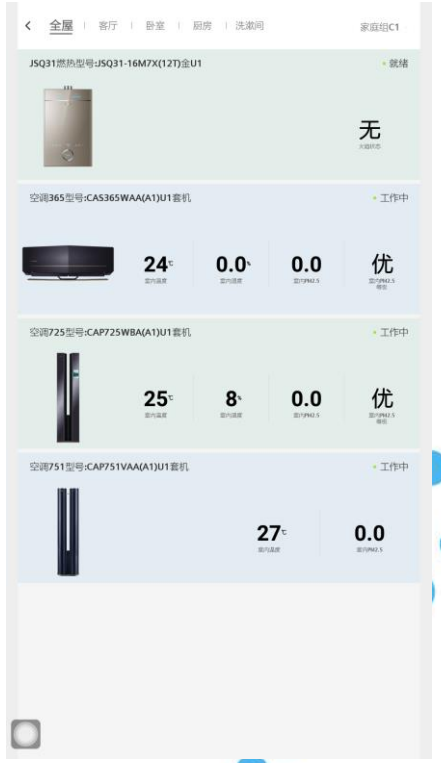
实施效果



- 智能家居控制类任务与照片分享等社交类任务对于实时性要求不同，前者对于内存消耗比较小但对于实时性要求比较高，后者相反。

设计方案

- 利用K-means算法对任务进行聚类，属于无监督学习方法，把任务进行分类，根据不同分类采用不同调度算法
- 任务运行特性收集包括调度频率、内存申请大小及频率、运行时长、中断处理频率、任务调度顺序等等
- 后台根据收集到的数据，进行任务分类，并制定优化策略，并把优化策略下载到设备端
- 操作系统根据最新策略进行优化调度



操作系统研究方向与探索 任务调度优化

要解决的问题

- 操作系统根据任务关联度进行优先调度
- 解决资源限制设备中任务调度需要用户长时间等待问题

设计方案

- 利用Apriori算法探索任务之间的关联关系
- 对于同时发生或者时间维度上接续发生的任务，调整任务调度顺序
- 通常调整TCB任务块来进行任务调度调整

实施效果

- 用户通过家庭留言板（任务）收到的图片与图片查看器（任务）

$$
\begin{aligned}
&\text{Apriori}(T, \epsilon) \\
&\quad L_1 \leftarrow \{\text{large } 1-\text{itemsets}\} \\
&\quad k \leftarrow 2 \\
&\quad \textbf{while } L_{k-1} \neq \emptyset \\
&\qquad C_k \leftarrow \{c = a \cup \{b\} \mid a \in L_{k-1} \wedge b \notin a, \{s \subseteq c \mid |s| = k-1\} \subseteq L_{k-1}\} \\
&\qquad \textbf{for transactions } t \in T \\
&\qquad\quad D_t \leftarrow \{c \in C_k \mid c \subseteq t\} \\
&\qquad\quad \textbf{for candidates } c \in D_t \\
&\qquad\qquad count[c] \leftarrow count[c] + 1 \\
&\qquad L_k \leftarrow \{c \in C_k \mid count[c] \geq \epsilon\} \\
&\qquad k \leftarrow k + 1 \\
&\quad \textbf{return } \bigcup_k L_k
\end{aligned}
$$

操作系统研究方向与探索 异常任务检测

要解决的问题

- 操作系统异常任务检测
- 解决设备运行过程中异常任务检测方法比较单一、检测维度单一问题

实施效果

- 潜在堆栈溢出任务发现

Algorithm 1 $iTree(X, e, h)$

Input: X - input data; e - current height; h - height limit.
Output: an $iTree$.

1: **if** $e \geqslant h$ OR $|X| \leqslant 1$ **then**
2: **return** $exNode\{Size \leftarrow |X|\}$;
3: **else**
4: Randomly select an attribute q;
5: Randomly select a split point p between min and max values of attribute q in X;
6: $X_l \leftarrow filter(X, q < p)$, $X_r \leftarrow filter(X, q \geqslant p)$;
7: **return** $inNode\{ Left \leftarrow iTree(X_l, e+1, h)$,
$Right \leftarrow iTree(X_r, e+1, h)$,
$SplitAttr \leftarrow q, SplitValue \leftarrow p\}$;
8: **end if**

设计方案

- 利用多元离群点算法检测异常任务
- 设计异常检测特征，比如堆栈、内存使用、锁的使用数量等等
- 设定异常任务处理规则
- 利用多元离群点检测算法从多个维度对操作系统任务进行检测，提前预知危险任务并根据预设规则进行处理

操作系统研究方向与探索 总结

- AI在操作系统的应用无论从业务角度还是传统研究角度都有很大的用武之地
- Embedded AI会成为操作系统的标准配置项，如同网络协议、文件系统一样
- 操作系统的研究需要AI注入新活力

谢谢