



# 在轨（卫星）计算： 挑战，机遇和尝试

Mengwei Xu (徐梦炜)  
CS Dept. of BUPT

# 个人介绍



- 徐梦炜，北邮特聘副研究员/博士生导师
  - 入选中国科协青托，北京市科技新星，微软“铸星计划”学者等
  - 主要研究领域：边缘智能系统、卫星计算系统
  - 主页：<https://xumengwei.github.io/>
  - 项目：<https://github.com/UbiquitousLearning>



Bachelor/PhD  
2011-2022



Visiting Scholar  
2018-2019



Asst. Professor  
2020-present



# 大纲

- 卫星技术发展
- 卫星计算需求
- 卫星计算挑战
- 卫星计算尝试



# 大纲

- 卫星技术发展
  - 卫星互联网
  - 微小型卫星（立方星）
  - 卫星COTS硬件
- 卫星计算需求
- 卫星计算挑战
- 卫星计算尝试

# 卫星星座背景

国家



国际地位

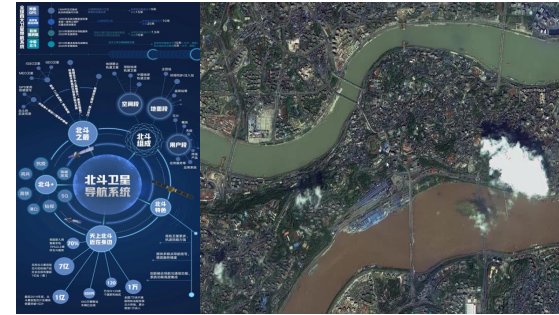
近地轨道6w颗 vs. 星链4.2w颗



国防军事

俄乌冲突、Boeing、神州航天

社会、人民



生活服务

导航、天气预报、灾害救治



科学研究

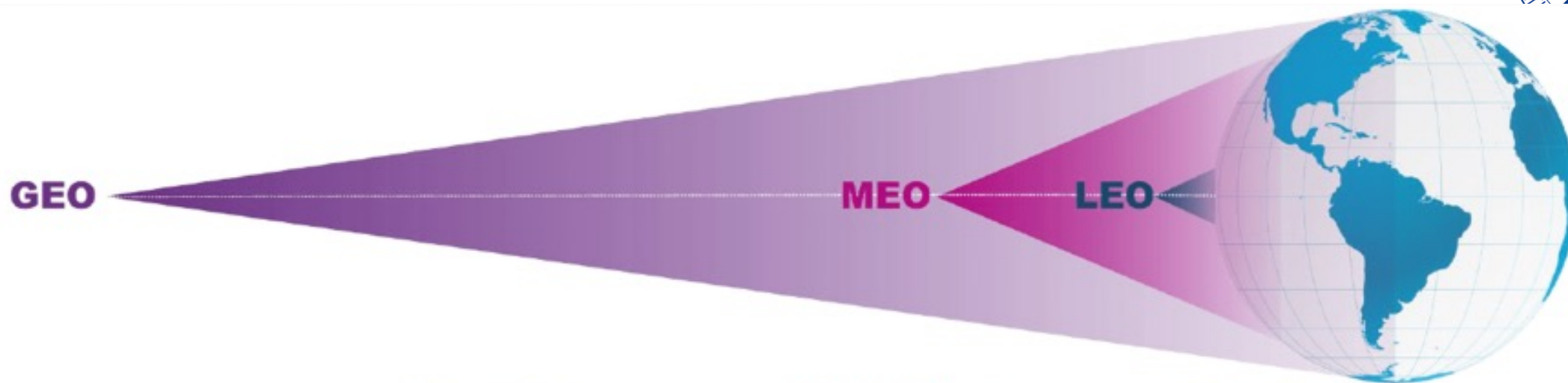
天文、地质、生态、生命科学



# 卫星的代际演化

时间	用途	特征	轨位	示例
1960s~1980s	通信、科研	体积大(10+m <sup>3</sup> ) 重量大(数百公斤) 寿命短(5~10年)、成本高	低轨(1000km)	Intelsat I、Pioneer 系列 东方红1号
1990s~2000s初	通信、导航、 遥感	体积缩小(几立方米) 重量减小(几十公斤) 寿命长(10+年)、成本高	高轨(36000km) 低轨 (500~1500km)	Globalstar公司、 LandSat系列、 高分系列
近5~10年	通信、导航、 遥感	体积、重量因功能而差 异较大 寿命长(10+年)	高轨(36000km) 中轨(20000km) 低轨 (500~1500km)	Intelsat系列 Galileo导航卫星 “中星”系列
未来5~10年	通信、导航、 遥感、计算一 体化	体积、重量进一步减小 成本低廉	天地一体 中高低全覆盖	-

# 卫星互联网现状



	<b>GEO (~36,000 km)</b>	<b>MEO (~8,000 km)</b>	<b>LEO (~1,000 km)</b>
<b>Latency</b>	Medium (~700 m/s)	Low (~150 m/s)	Very low (~50 m/s)
<b>Network size for global services</b>	3 satellites (99% coverage)	6 satellites (96% coverage)	Thousands of satellites (100% coverage)
<b>Data gateways required</b>	Few, fixed	Several, flexible	Numerous, local
<b>Technology readiness level</b>	Proven, deployable technology	Proven, deployable technology	Technology still in development for satellite internet
<b>Cost to deploy network</b>	\$1 - 1½bn	Approx. \$1½bn	\$5 - 15bn
<b>Satellite design life (replacement cycle)</b>	15 years	12 years	5 - 7 years



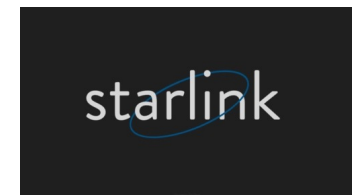
# 卫星互联网现状

## ➤ MEO/GEO卫星

- ✓ 数量稀少，未成体系，导航为主
- ✓ 中国北斗（55颗），美国GPS（28颗），俄罗斯GLONASS（30颗），欧盟Galileo（30颗）

## ➤ LEO卫星互联网

- ✓ Starlink: 截至2023.5，卫星发射总数4469颗，在轨4024颗，目标4.2万颗
- ✓ OneWeb: 截至2023.3，卫星网络总数582颗，差40颗完成低轨星座建设
- ✓ 中国星网: 2024年开始发射，目标1.3万颗卫星





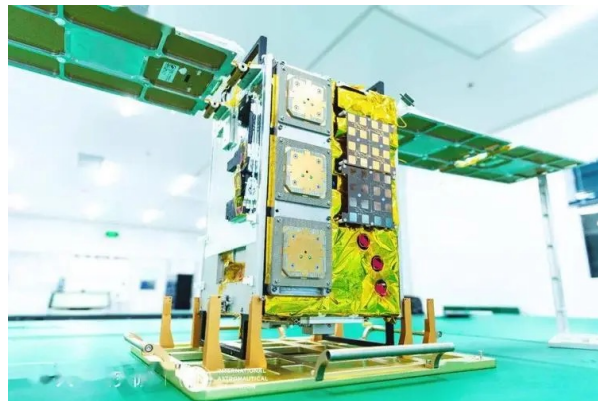
# 卫星微型化

传统卫星



重量 $\geq 100\text{kg}$   
体积 $\geq 1\text{m}^3$

微小型卫星  
(Microsatellite)



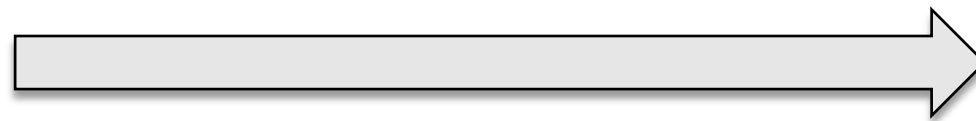
纳卫星  
(NanoSat)



立方星  
(CubeSat)



重量:  $\sim 1\text{kg}$   
体积:  $1\text{U}(10*10*10\text{cm})$



小型化、标准化、规模化

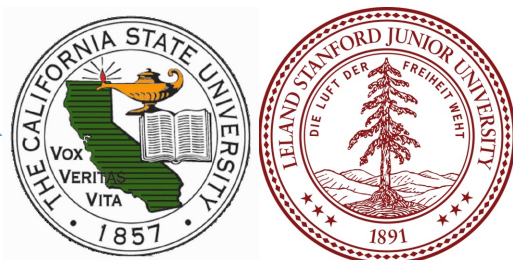
# 立方星发展历程

## 初期



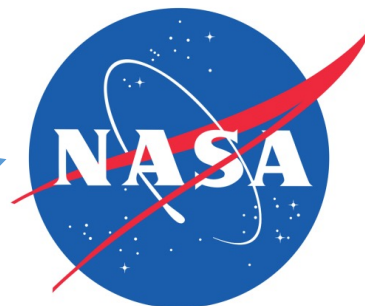
1999年，加州州立大学卡拉·卡里达斯教授提出“微小卫星”概念

2000年，加州州立大学与斯坦福大学推出立方星标准规范-CubeSat标准



## 标准化

## 兴起



2003年，NASA推出首个立方星项目 CubeSat Launch Initiative(CSLI)

至今，政府、企业、高校参与立方星设计制造，立方星座蓬勃发展



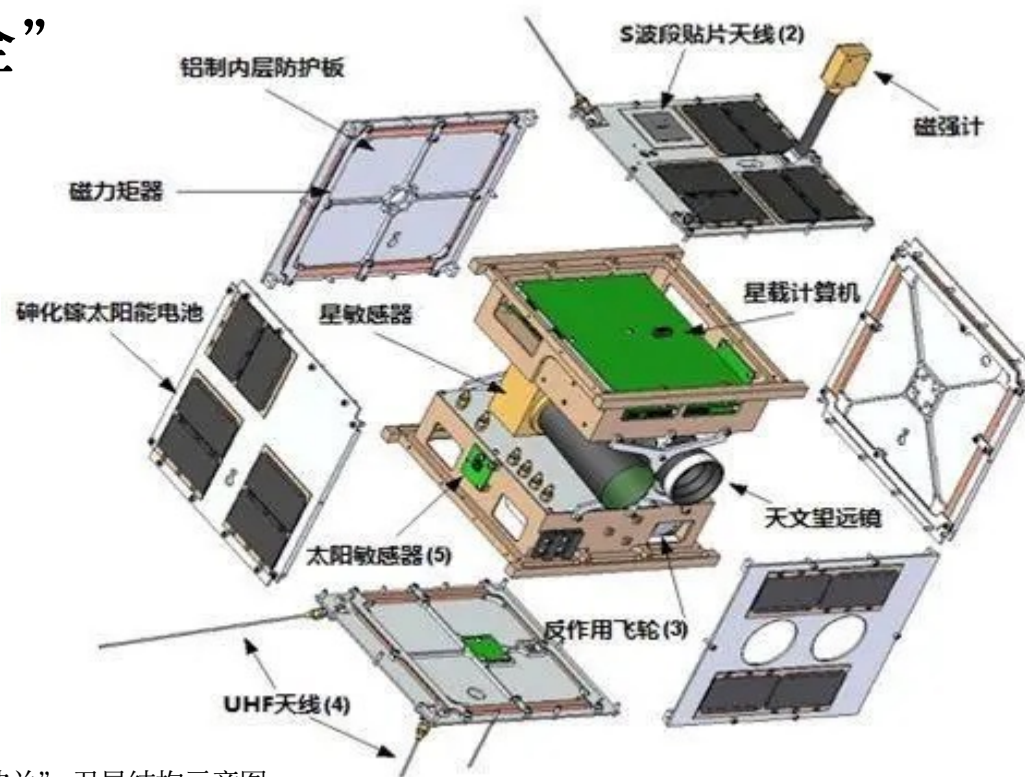
## 发展

# 立方星概述

## ➤ 综合优势突出

- ✓ 成本低
- ✓ 开发快
- ✓ 敏捷性
- ✓ 扩展（协同）性

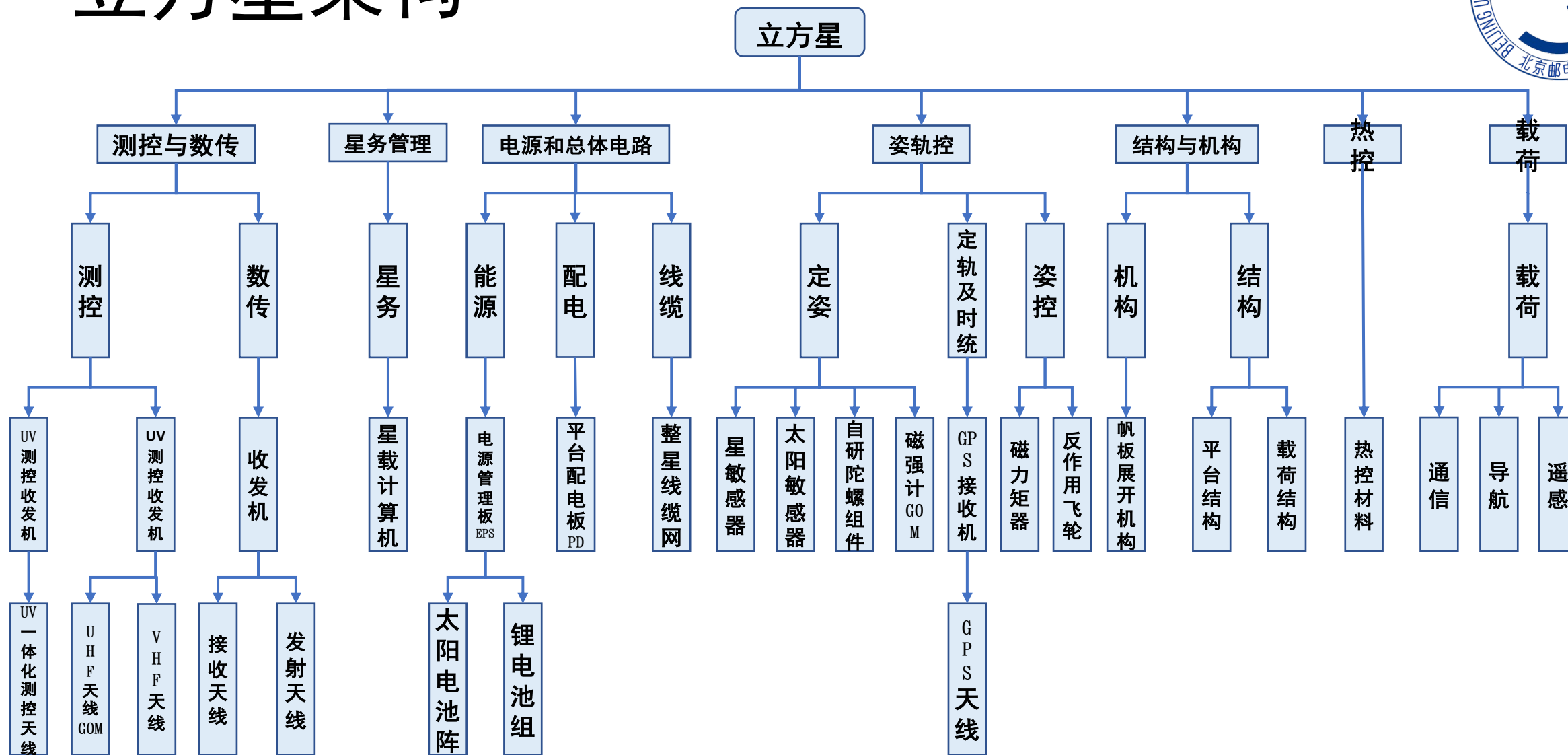
## ➤ “麻雀虽小，五脏俱全”



图片来源：波兰微小卫星技术的崛起与发展：“明亮目标探测器-波兰”卫星结构示意图

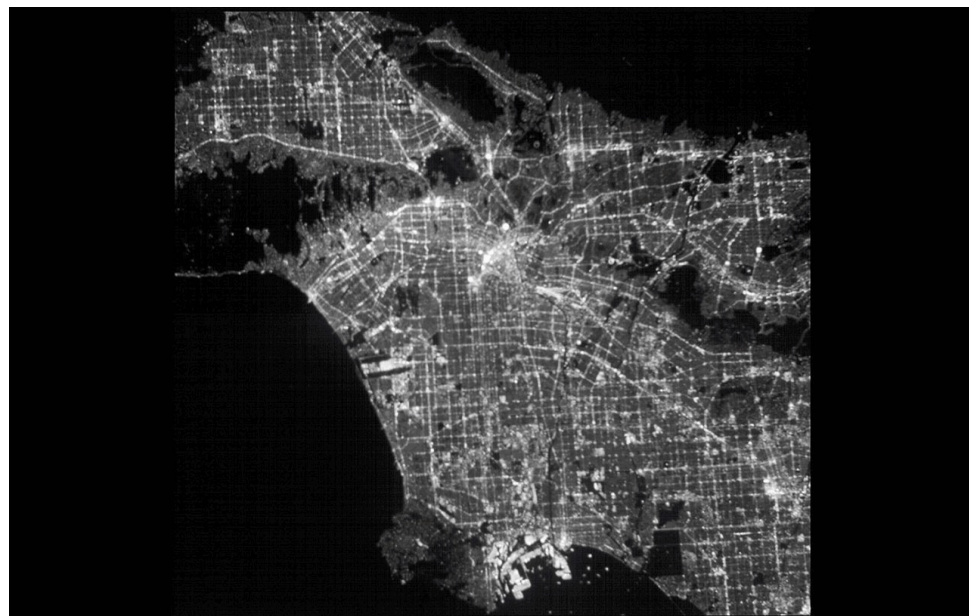


# 立方星架构



# 立方星功能

- 科学研究
- 技术验证
- 教育教学
- 通讯遥感
- 空间探索
- 气象监控
- ○ ○ ○



NASA Planet-Hunter立方星ASTERIA拍摄的LA夜景  
(分辨率: 30m)

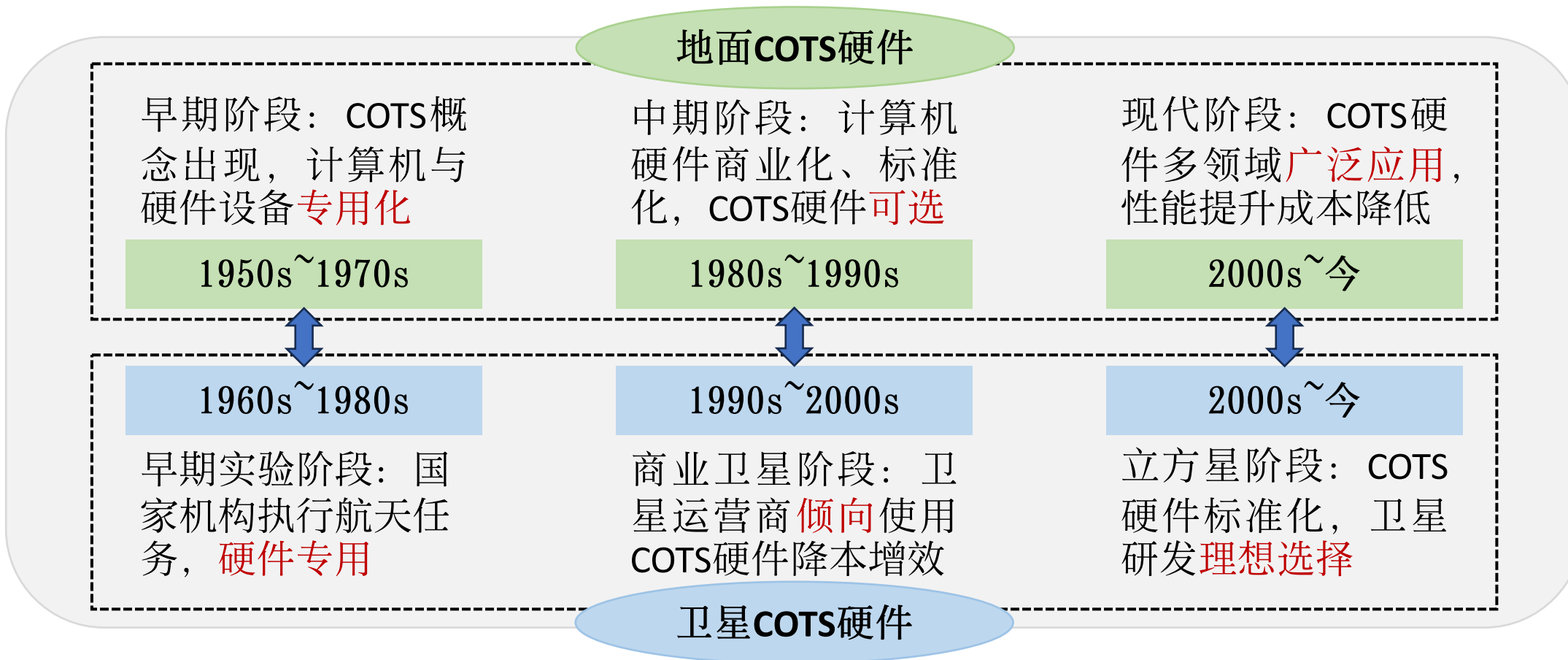


NASA Mars Cube One在距离地球1百万千米处拍摄，同时短暂承担火星着陆器-地球的通信relay功能。

[1] <https://www.nasa.gov/feature/jpl/a-pale-blue-dot-as-seen-by-a-cubesat>  
[2] <https://www.nasa.gov/feature/jpl/planet-hunter-cubesat-images-los-angeles>

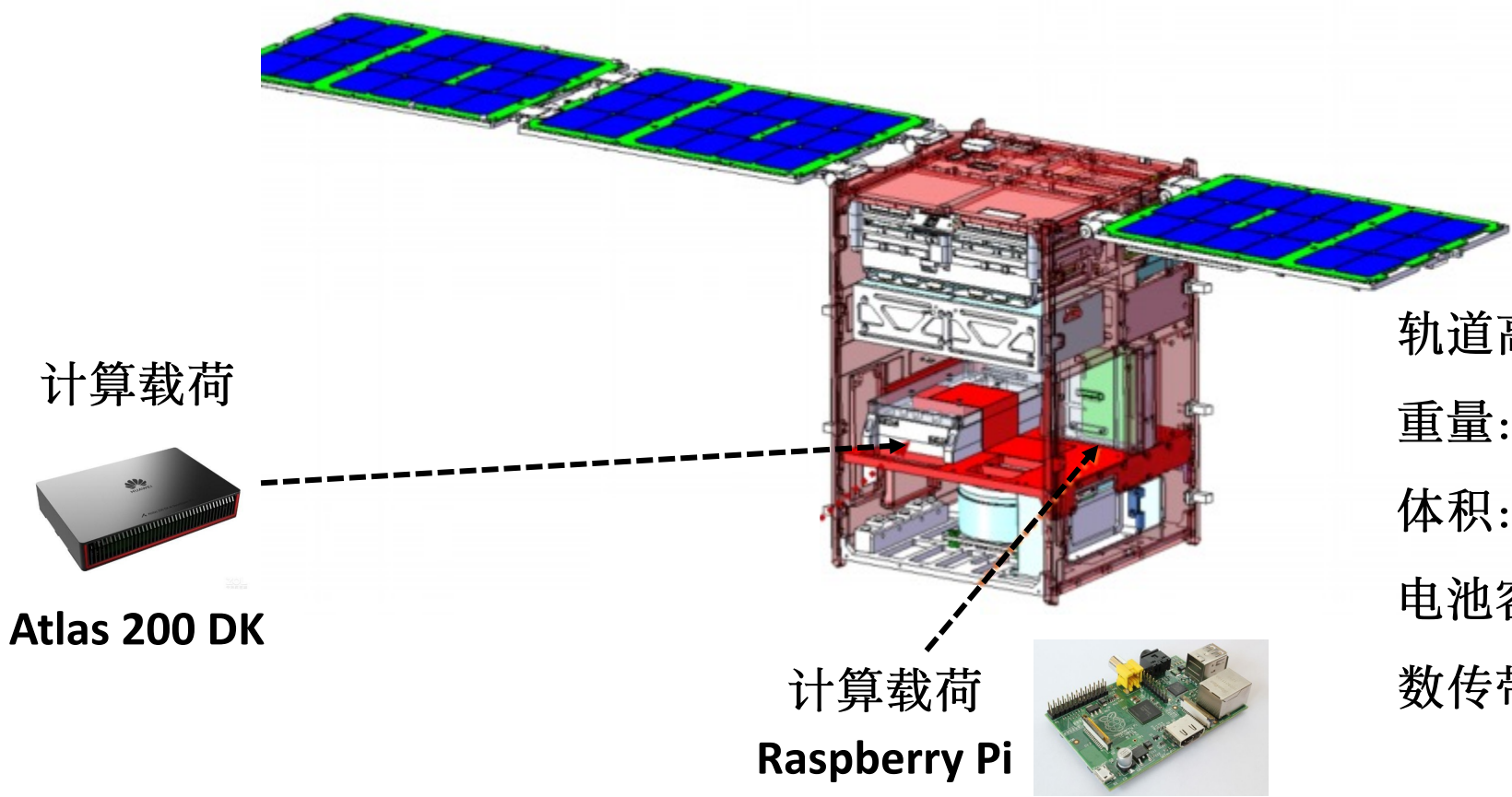
# 卫星的“器官” - 各类COTS硬件

➤ 从专用硬件向COTS硬件转变



# COTS硬件

➤以“北邮一号”为例



计算载荷



Atlas 200 DK

计算载荷

Raspberry Pi



轨道高度:  $\approx 500\text{km}$

重量: **17.44kg**

体积: **434 × 347 × 341mm**

电池容量: **236Wh**

数传带段: **1Mbps (上行)**

**100Mbps (下行)**

# 卫星COTS硬件未来之路

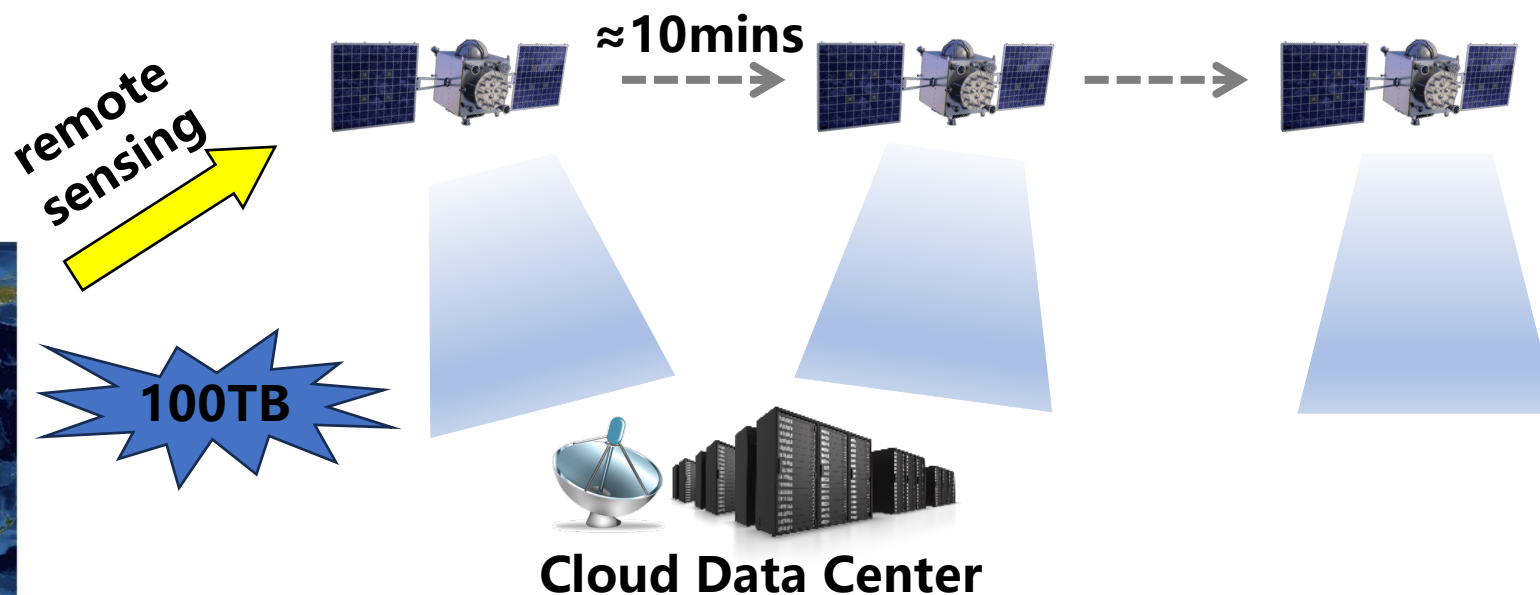
## 优势

- ✓ 降低成本
- ✓ 加快开发
- ✓ 灵活可靠
- ✓ 易于获取

## 挑战

- ✗ 适配性
- ✗ 安全性
- ✗ 可靠性

## ➤ 基于卫星COTS的新机遇





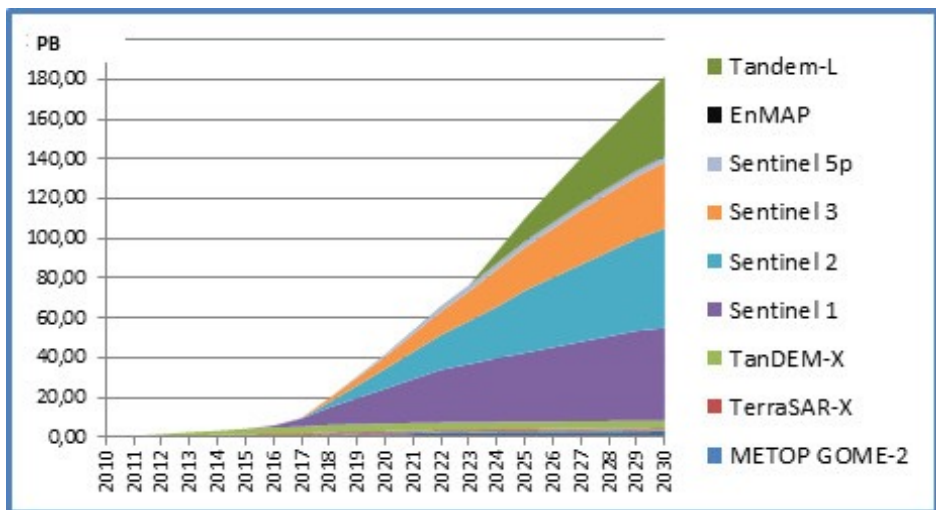


# 大纲

- 卫星技术发展
- 卫星计算需求
  - 遥感数据处理
  - 边缘计算拓展
- 卫星计算挑战
- 卫星计算尝试

# 卫星计算应用1：遥感数据处理

- 在轨遥感数据处理优势
  - 提升遥感观测能力 (capacity)
  - 提升遥感任务时效 (timeliness)



German Satellite Data Archive (D-SDA)  
于2022年新增遥感数据超6000PB

how much data do one satellite generate per day

The amount of data generated by a single satellite per day can vary widely depending on its purpose, design, and the technology used.

For example, remote sensing satellites, such as those used for weather monitoring and earth observation, can generate large amounts of data. The European Space Agency's Sentinel-2 satellite, which captures multispectral images of Earth's surface, generates about 1.6 TB of data per day.

Communication satellites, on the other hand, don't generate much data themselves, but rather they transmit and receive data between different points on Earth.

Satellites used for scientific research, like the Hubble Space Telescope, can also generate significant amounts of data. The Hubble Space Telescope generates about 120 GB of data per week, which translates to roughly 17 GB per day.

It's also important to note that not all data generated by a satellite is transmitted back to Earth. Many satellites have onboard processing systems that compress the data before transmission to save bandwidth.

Please note that these are just general examples and actual data generation can vary. Advances in technology may also increase the capacity for data generation and transmission for newer satellites.

每天产生  
1.6TB数据

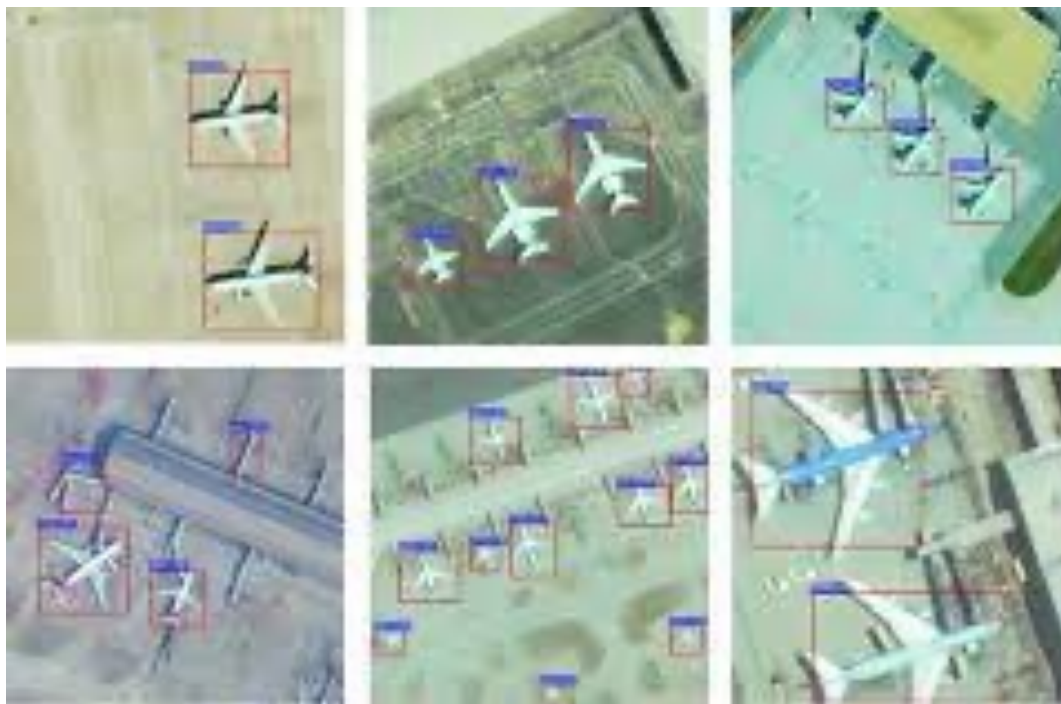
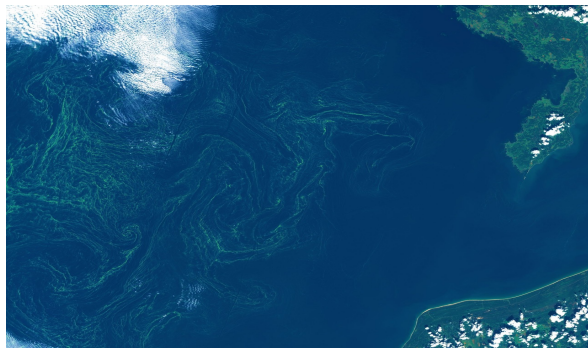
卫星计算有  
助于降低传  
输数据量

更精准、更广阔、更频繁地对地面进行观测

[1] [https://www.dlr.de/eoc/en/desktopdefault.aspx/tabid-12632/22039\\_read-51751](https://www.dlr.de/eoc/en/desktopdefault.aspx/tabid-12632/22039_read-51751)

# 卫星计算应用1：遥感数据处理

- 在轨遥感数据处理优势
  1. 提升遥感观测能力 (capacity)
  2. 提升遥感任务时效 (timeliness)

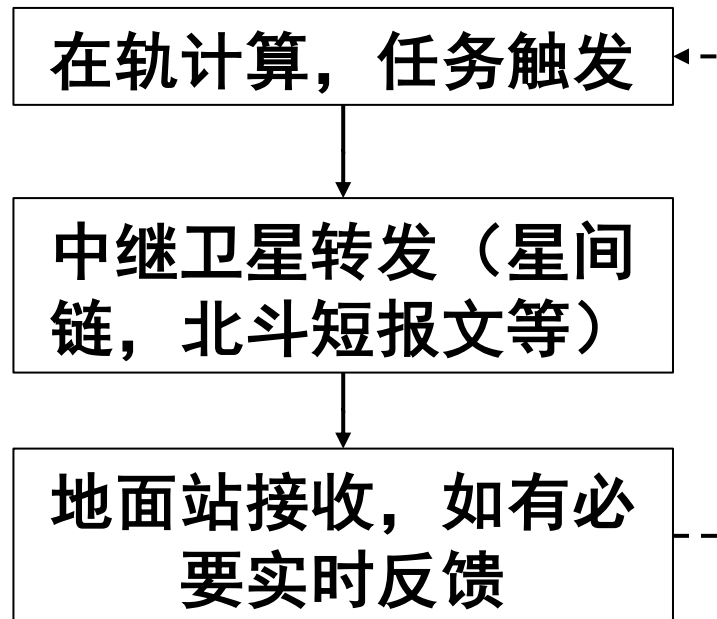


## 星上数据处理类型

- 数据过滤/筛选
- 任务驱动的数据压缩
- 机器学习推理
- 等等

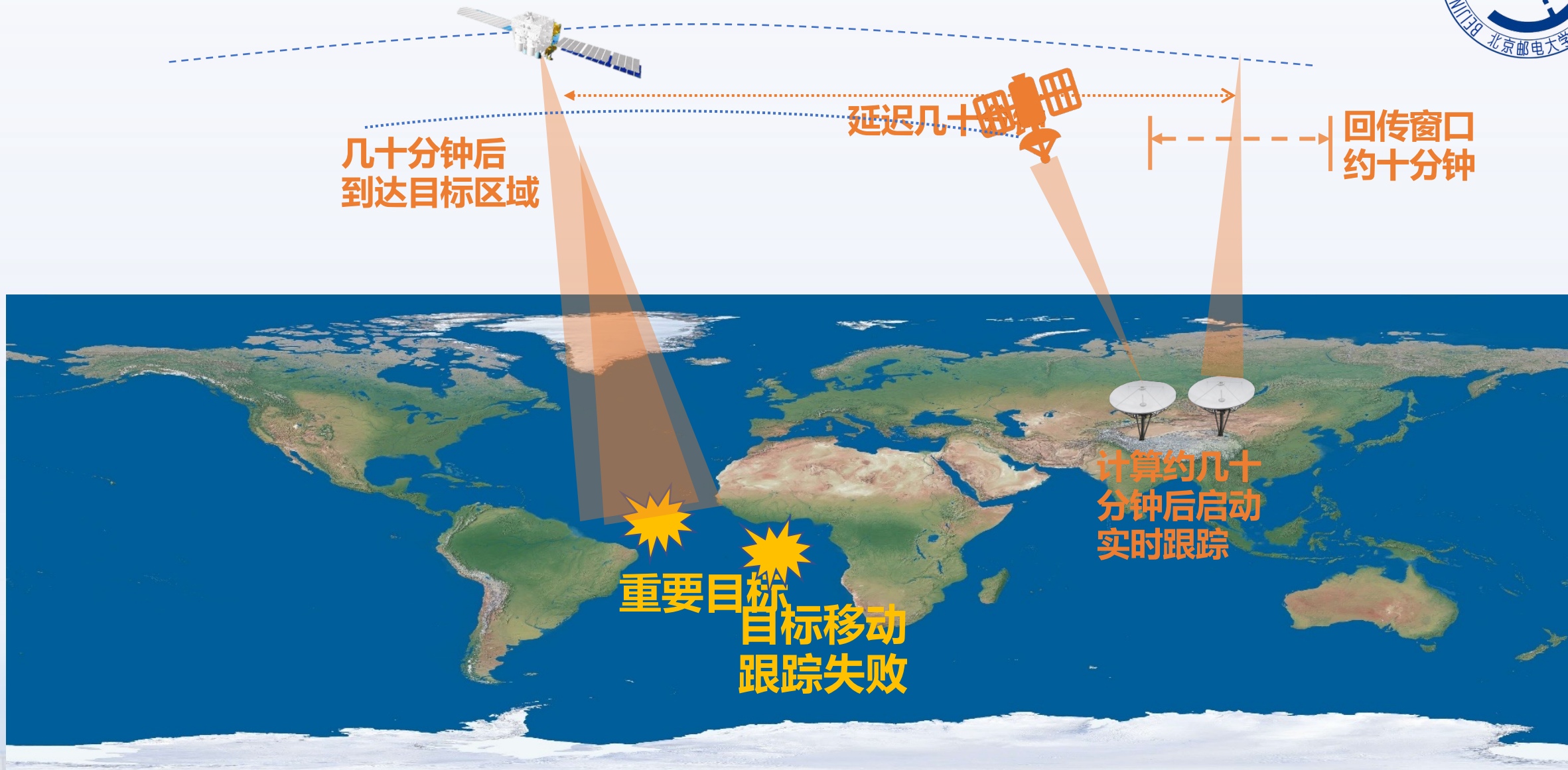
# 卫星计算应用1：遥感数据处理

- 在轨遥感数据处理优势
  1. 提升遥感观测能力 (capacity)
  2. 提升遥感任务时效 (timeliness)

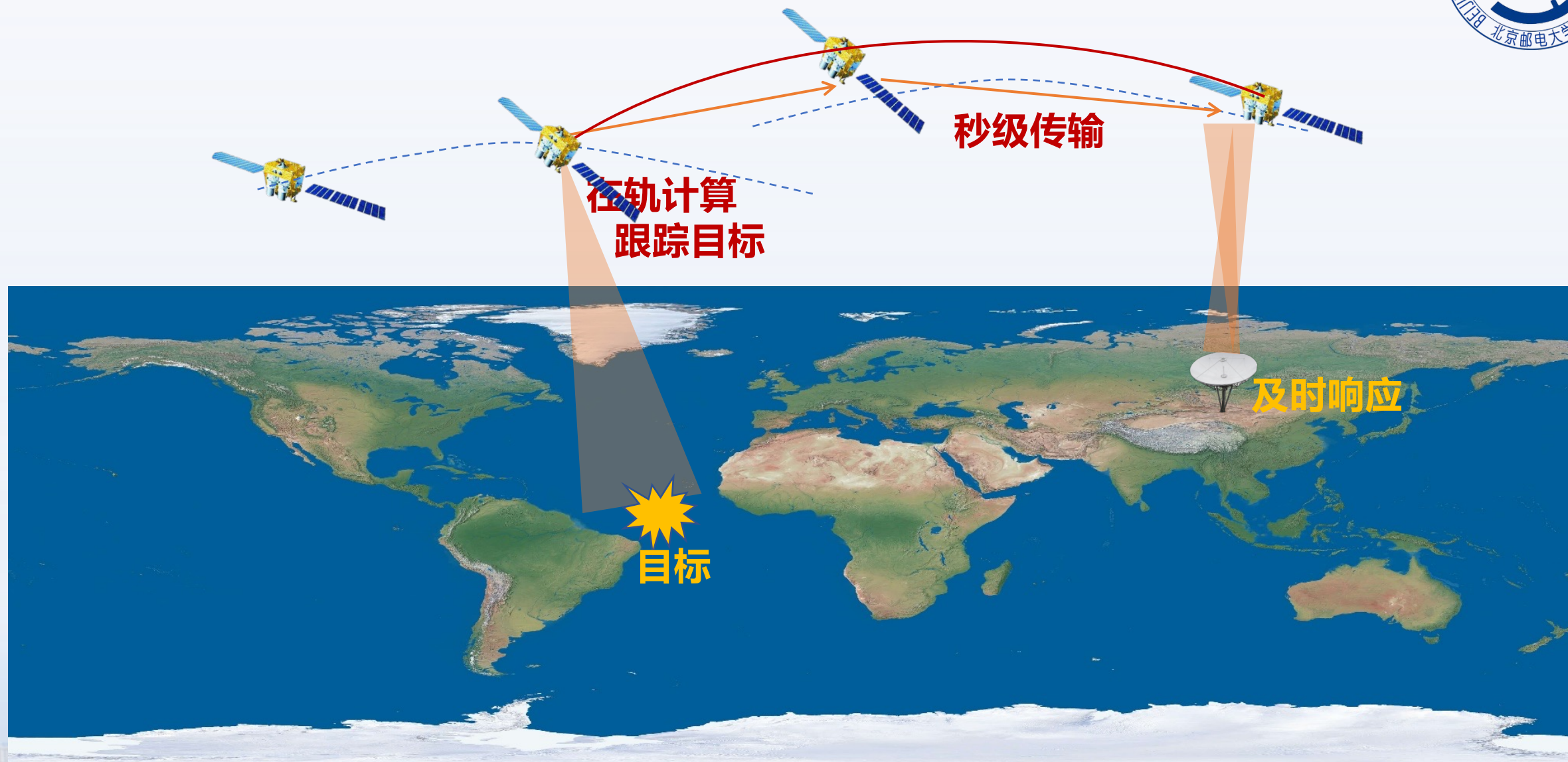


应用：目标跟踪，灾害检测（火灾、洪灾）等等

# 利用卫星计算提升任务时效性



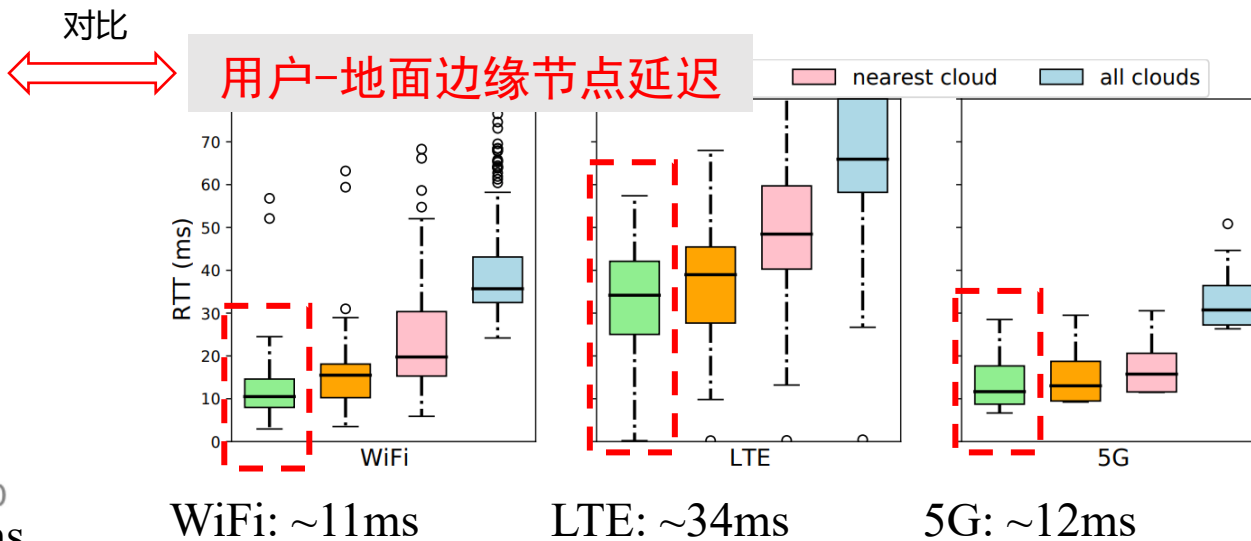
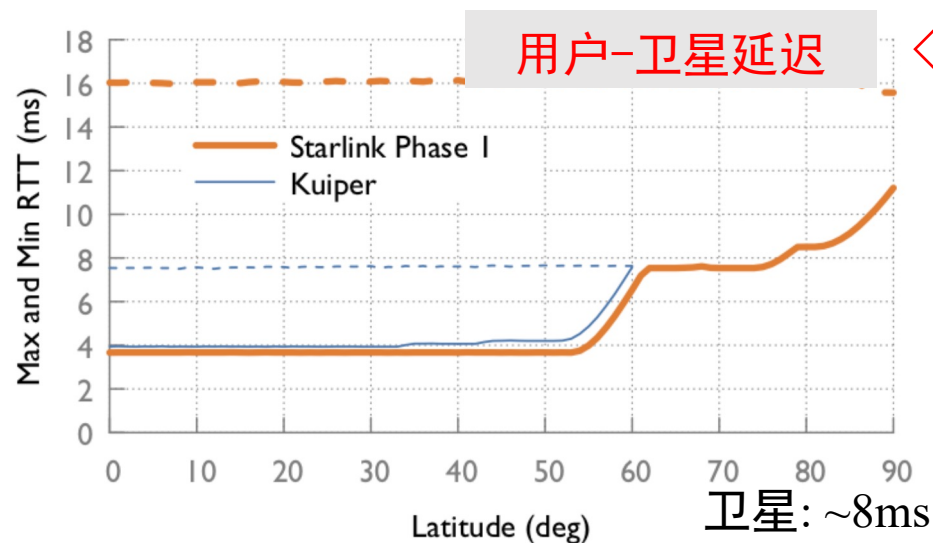
# 利用卫星计算提升任务时效性



# 卫星计算应用2：地面边缘计算拓展

- 卫星边缘计算

- 应用：CDN，跨域交互等



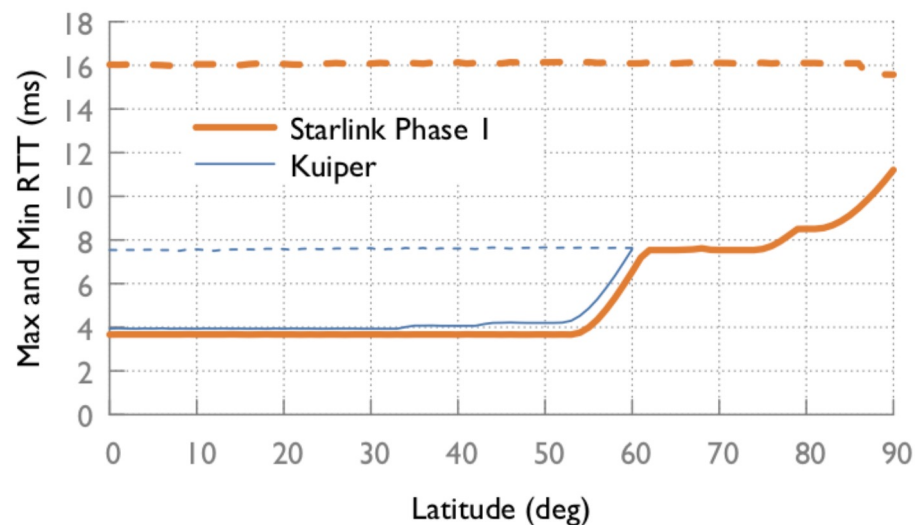
卫星计算能为终端用户提供泛在、低时延（平均8ms，最高16ms）的计算卸载服务

[HotNets'20] In-orbit Computing: An Outlandish thought Experiment?  
 [IMC'21] From Cloud to Edge: A First Look at Public Edge Platforms

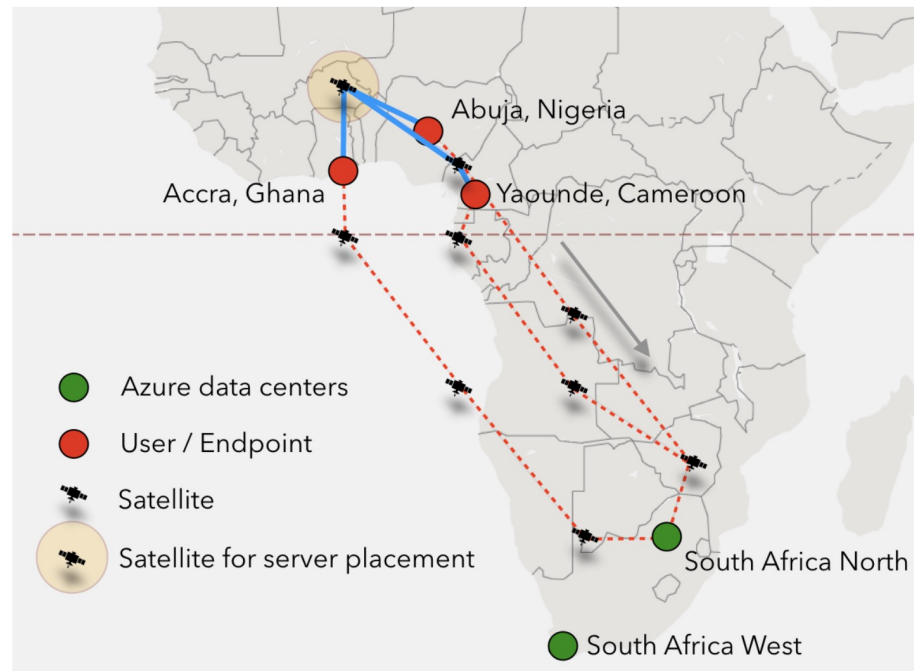
# 卫星计算应用2：地面边缘计算拓展

- 卫星边缘计算

- 应用：CDN，跨域交互等



卫星计算能为终端用户提供泛在、低时延（平均8ms，最高16ms）的计算卸载服务



选择合适卫星作为中继节点，可以大幅降低跨域用户交互时延

[HotNets'20] In-orbit Computing: An Outlandish thought Experiment?



# 卫星计算应用2：地面边缘计算拓展

- IoT数据采集与处理

- 智慧农业

1. 土壤质量

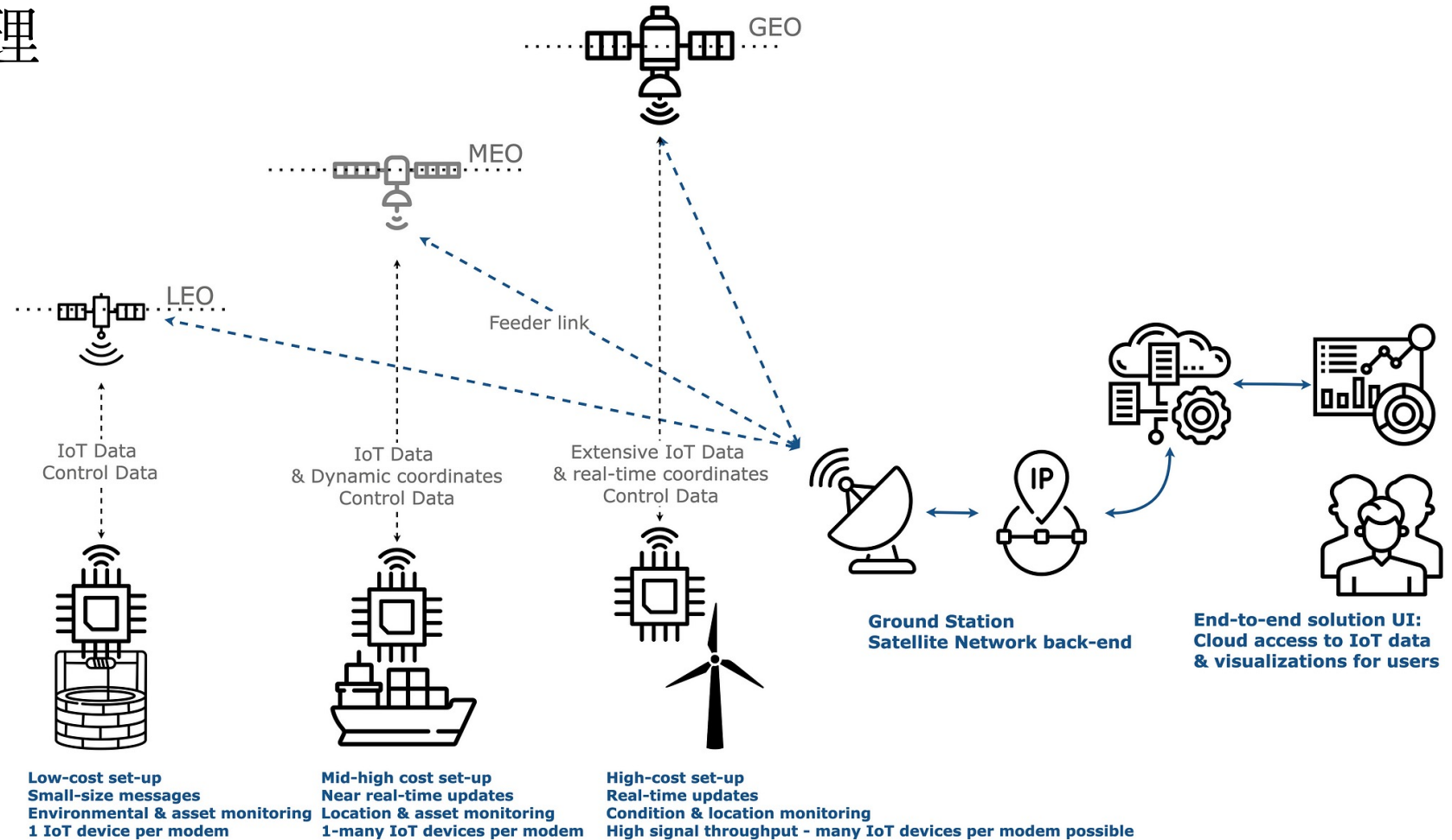
2. 水资源

- 环境检测

1. 野生动物迁徙

- 石油天然气

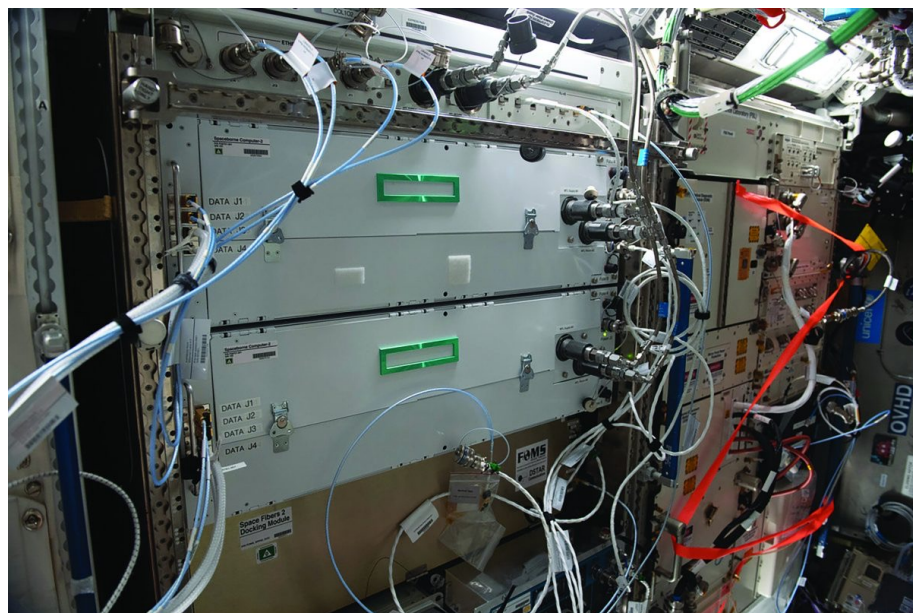
1. 管道状态监测



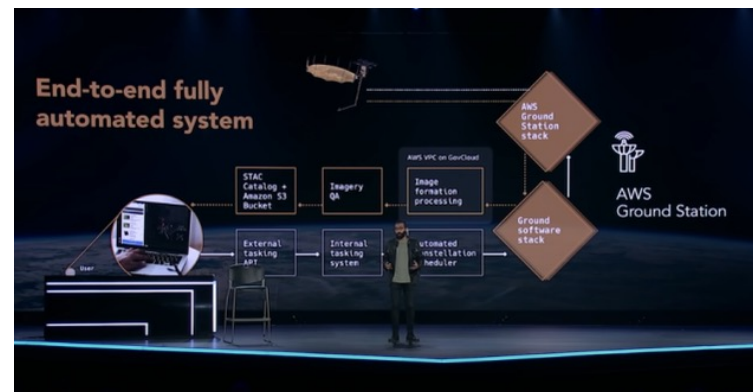
<https://www.ietfforall.com/types-of-satellite-networks-used-in-iot-solutions>

# 卫星计算应用3：科学实验和太空云

- 优势：低碳、真空



Hewlett Packard Enterprise's Spaceborne Computer-2, sent to the International Space Station in February 2021, is linked to Microsoft's Azure cloud through NASA and HPE ground stations. Credit: NASA



阿联酋的“希望号”火星探测器环绕火星探测气候数据，该探测器利用AWS Ground Station做一些数据的预处理，然后由AWS Ground Station把探测到的数据传到地球，地球上的科研人员在不到20分钟时间就能拿到来自火星的数据。

[1] <https://www.datacenterknowledge.com/hardware/space-final-frontier-data-centers>



# 总结-为什么我们需要卫星计算

Anywhere with data or  
users needs computing

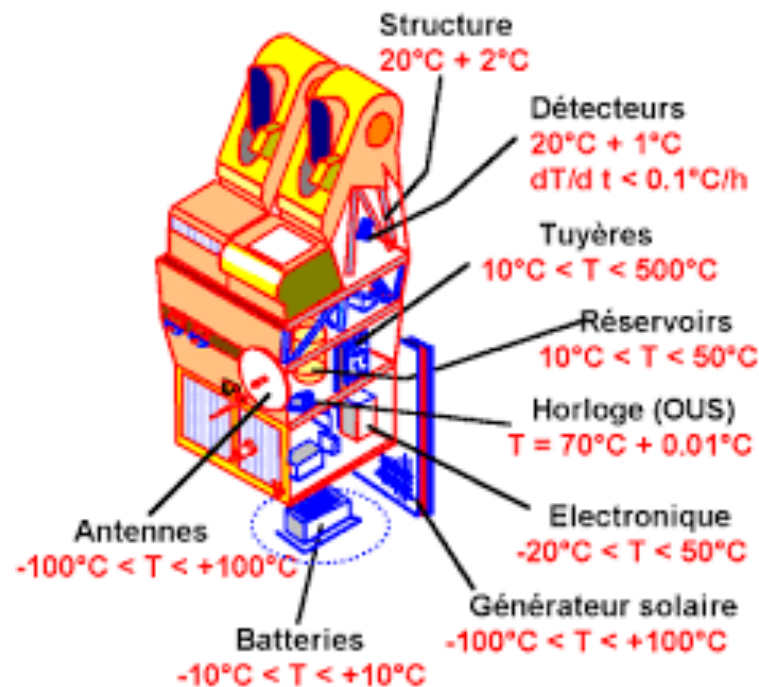


# 大纲

- 卫星技术发展
- 卫星计算需求
- **卫星计算挑战**
  - 资源受限
  - 网络协同
  - 健壮可靠
  - 架构融合
- 卫星计算尝试

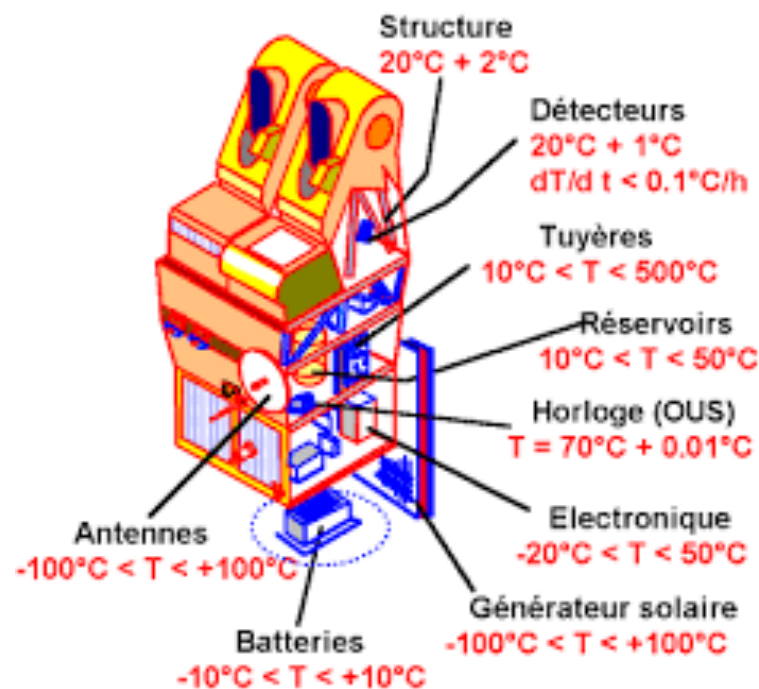
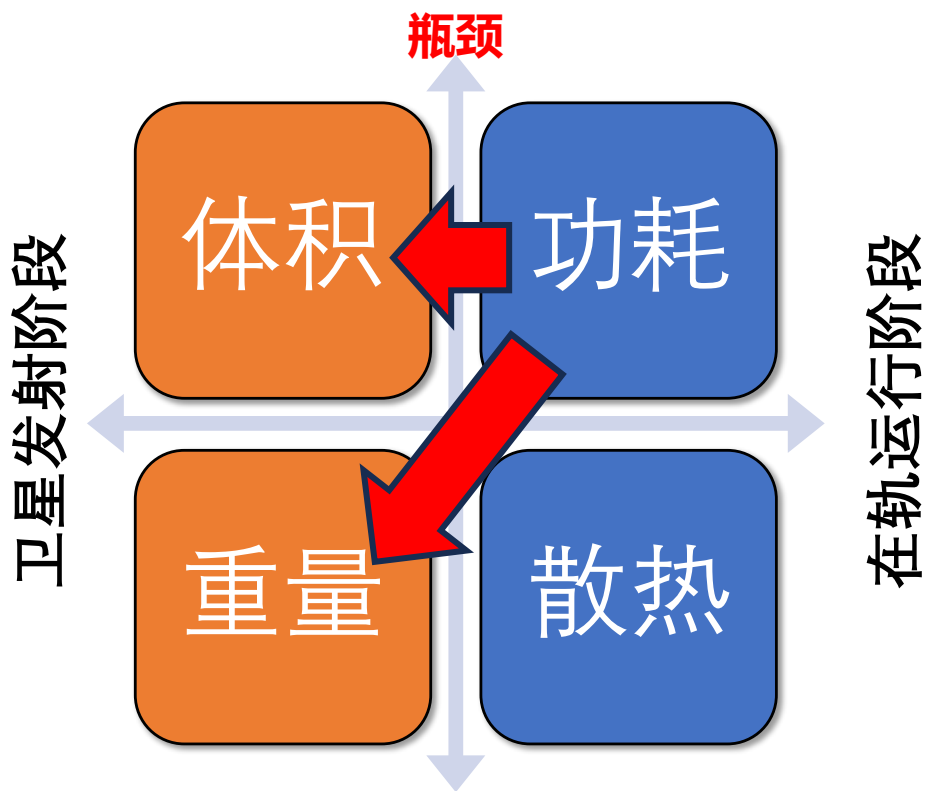
# 卫星计算挑战-1：资源受限

- 卫星是资源高度受限的嵌入式系统

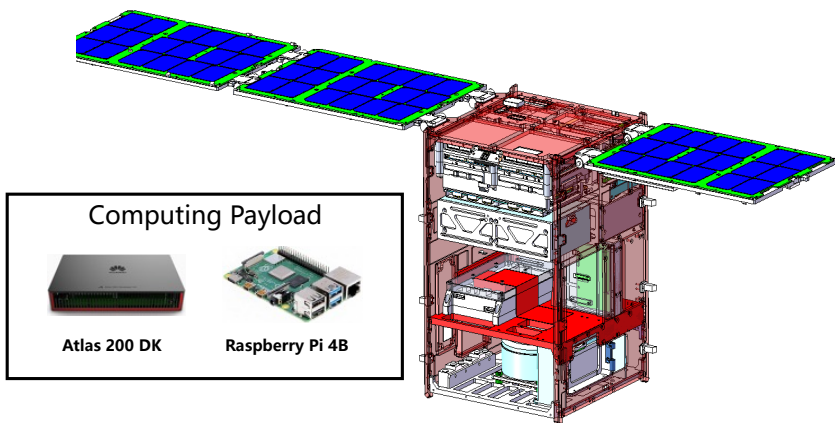


# 卫星计算挑战-1：资源受限

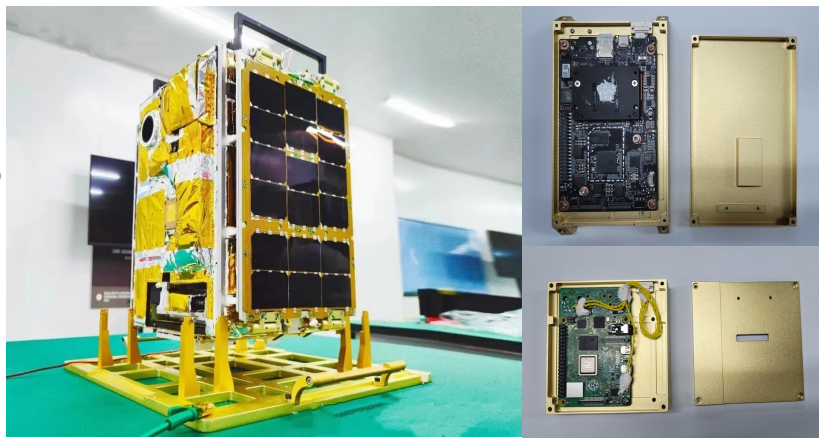
- 卫星是资源高度受限的嵌入式系统



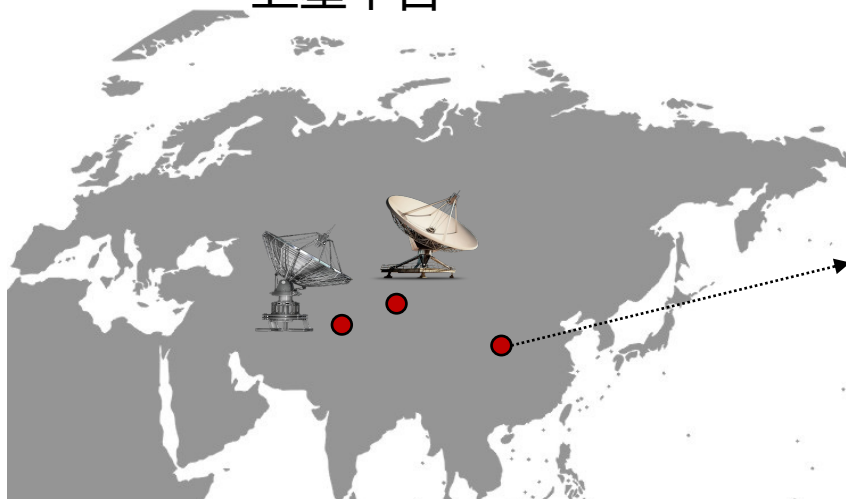
# 卫星计算挑战-1：资源受限



卫星平台



载荷



北邮一号

首个大规模卫星计算数据集

2023-03-22 to

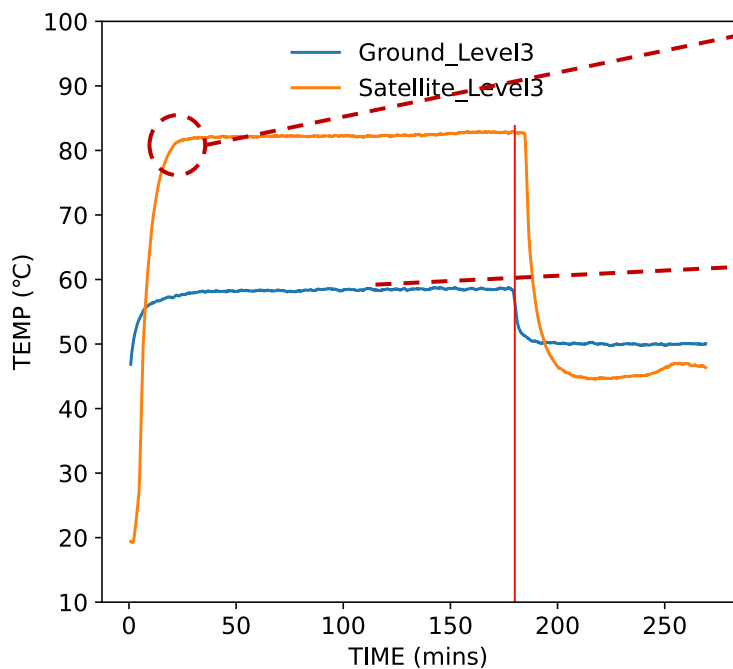
2023-07-10

15.61GB

超1千万数据行

测控数据：长沙  
数传：喀什和铜川

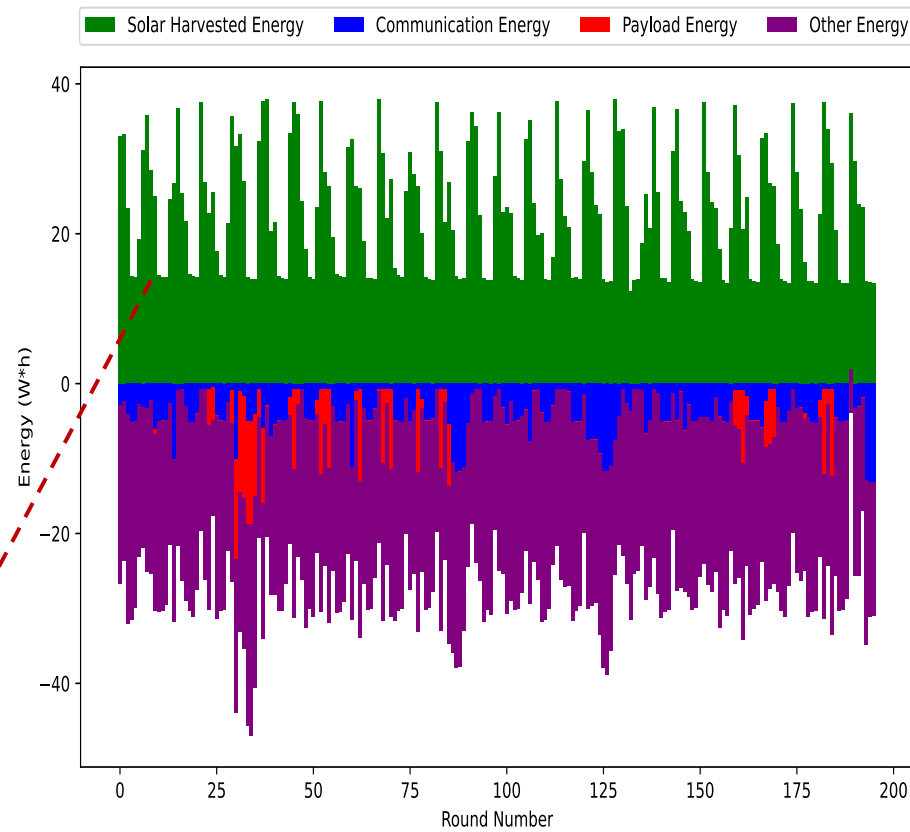
# 卫星计算挑战-1：资源受限



三核高负载下，星载树莓派快速到达最高工作温度（80-85度）

地面树莓派不受限制

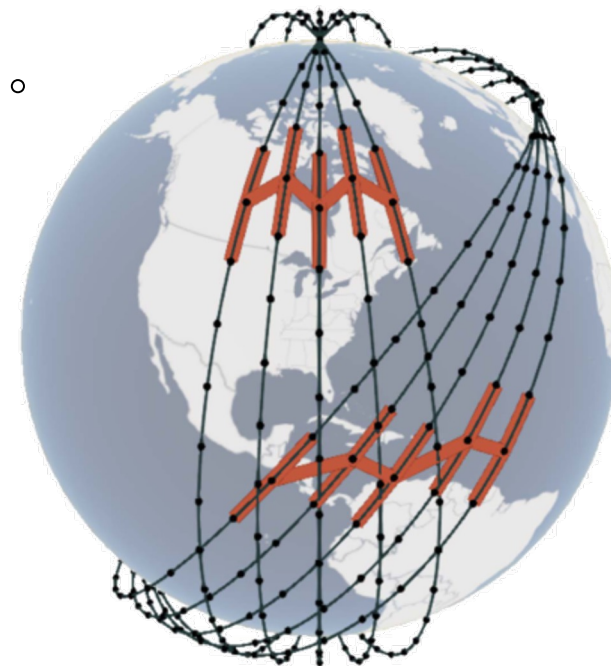
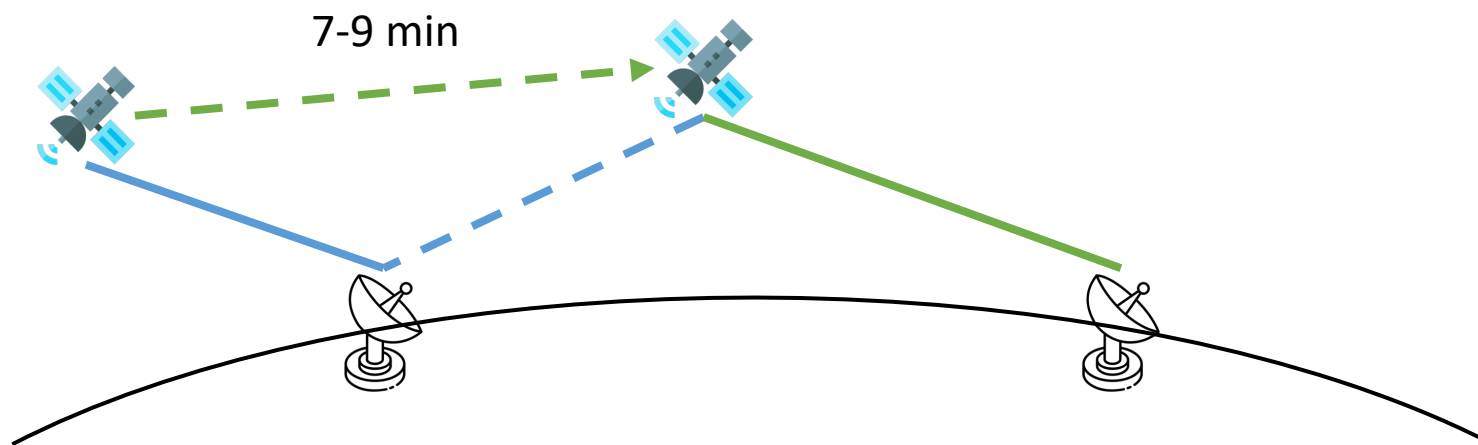
卫星单周期（90mins）内太阳能供应约15~40Wh，难以支撑高功率载荷持续工作





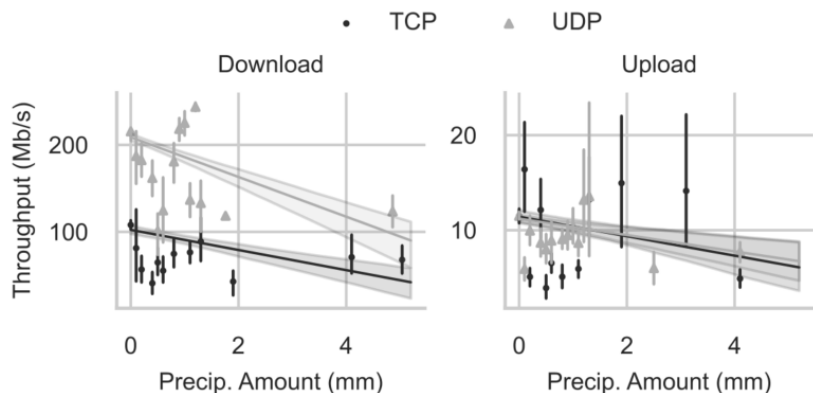
# 卫星计算挑战-2：网络协同

- 低轨卫星的高速移动性（ $27\text{KM/s}$ ），使得星座网络拓扑及路由动态变化，单次过顶仅有7-9分钟的通信窗口期。
  - 高度动态的网络拓扑要求高速收敛的路由算法。
  - 极短的通信窗口期要求低轨卫星具有离线自治能力。

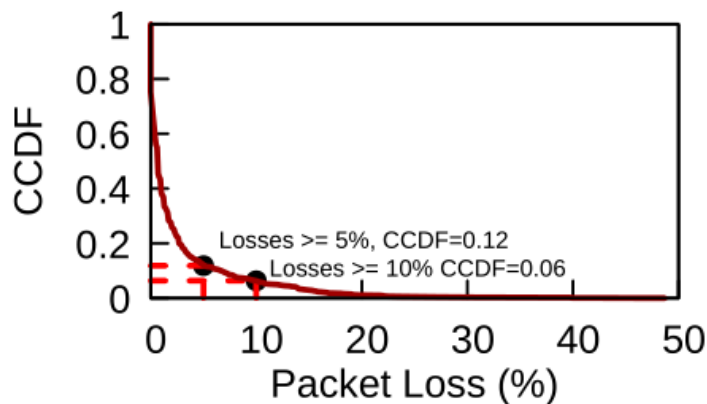


# 卫星计算挑战-2：网络协同

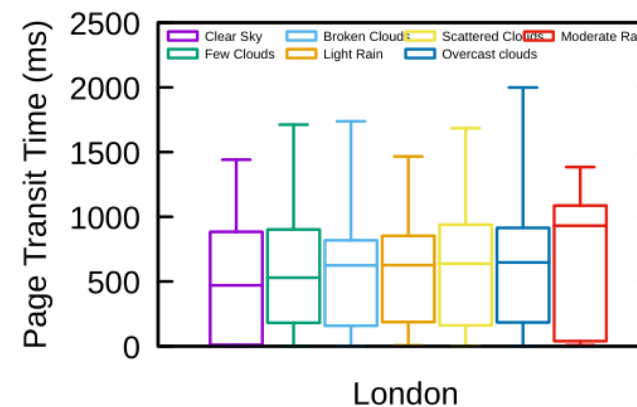
- 无线电链路稳定性差，网络参数波动大，要求应用数据交互以及云边协同过程具有高鲁棒性。



Starlink吞吐量随降雨量变化



Starlink丢包率累积分布函数

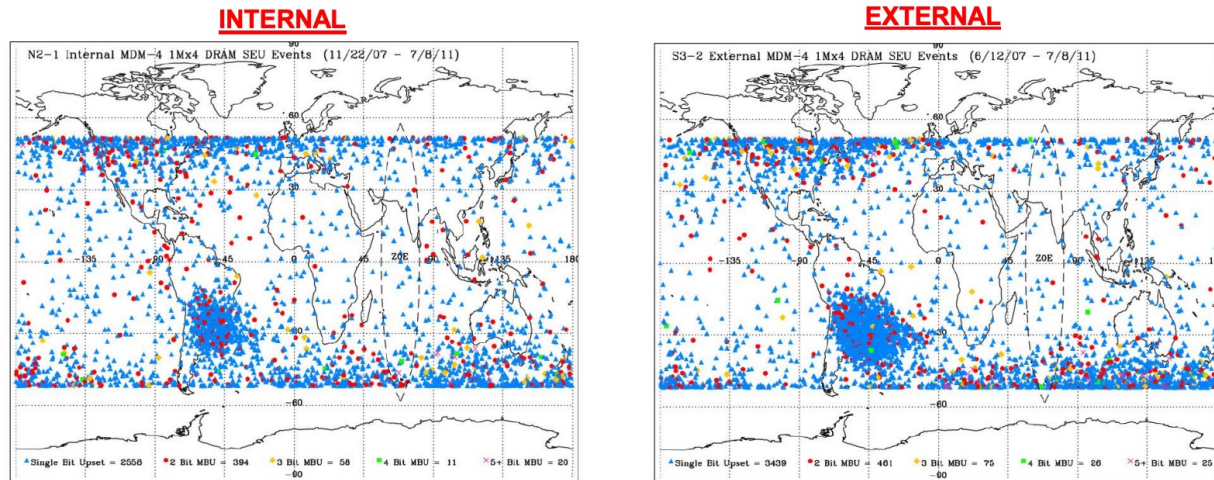


Starlink延迟随天气变化图

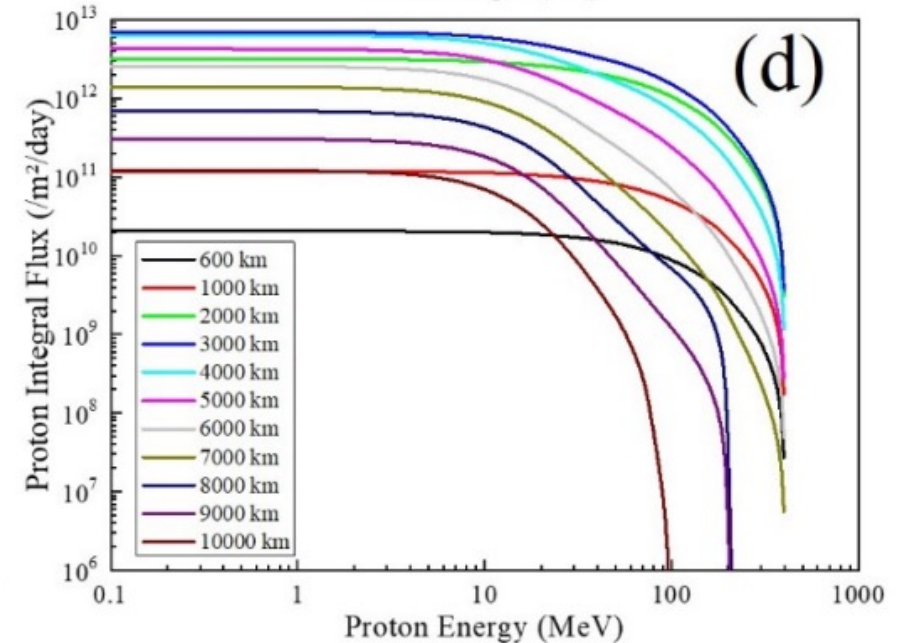
[INFOCOMM'23] Network Characteristics of LEO Satellite Constellations: A Starlink-Based Measurement from End Users  
 [IMC'22] A Browser-side View of Starlink Connectivity  
 [IMC'22] A First Look at Starlink Performance

# 卫星计算挑战-3：健壮可靠

- 单粒子效应 (single event effects)
  - 单粒子翻转、单粒子锁定、单粒子烧毁等
  - 影响因素：轨道高度、太阳活动、电子器件等



**GCR & trapped proton SEUs detected & corrected by Error Detection And Correction (EDAC) firmware in ISS computer system Dynamic Random Access Memory (DRAM). EDAC operation is part of the nominal system design.**



[1] ISS SPACE RADIATION EFFECTS AND EXPERIENCE: UTILIZATION FOR CURRENT AND FUTURE EXPLORATION PROGRAMS, 2012

[2] Influence of Orbital Parameters on SEU Rate of Low-Energy Proton in Nano-SRAM Device

# 卫星计算挑战-3：健壮可靠

- 应对策略

- 器件选择 (hardened) : 牺牲了可靠性以外的几乎所有指标



**Raspberry Pi 4B**

主频  
价格

1.4 GHz  
\$50



**RAD750**

200 MHz  
\$200,000

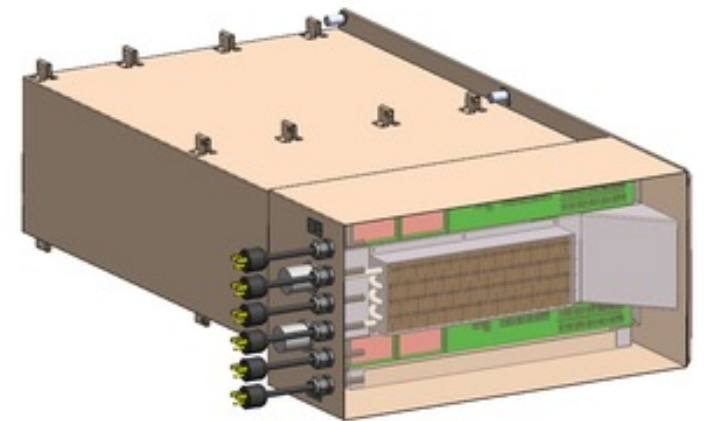
# 卫星计算挑战-3：健壮可靠

## • 应对策略

- 器件选择 (hardened)：牺牲了可靠性以外的几乎所有指标
- 系统架构：三模冗余、EDAC软硬件、FDIR系统软件
  - 案例：SpaceX; HPE Supercomputer on ISS

Linux is used for everything at SpaceX. The Falcon, Dragon, and Grasshopper vehicles use it for flight developers' desktops. SpaceX is "Linux, Linux, Linux", he said.

Rose went on to briefly describe the Dragon flight system, though he said he couldn't give too many requirements for when it gets close to the ISS. There are rules about how many faults a craft needs to be station. It uses triply redundant computers to achieve the required level of fault tolerance. The Byzanti computers do not agree. That situation could come about because of a radiation event changing memor



HPE Spaceborne Computer, 2017年发射至空间站。提供teraflop级算力，与地面误差为0.03%，仅采用软件层可靠性优化。

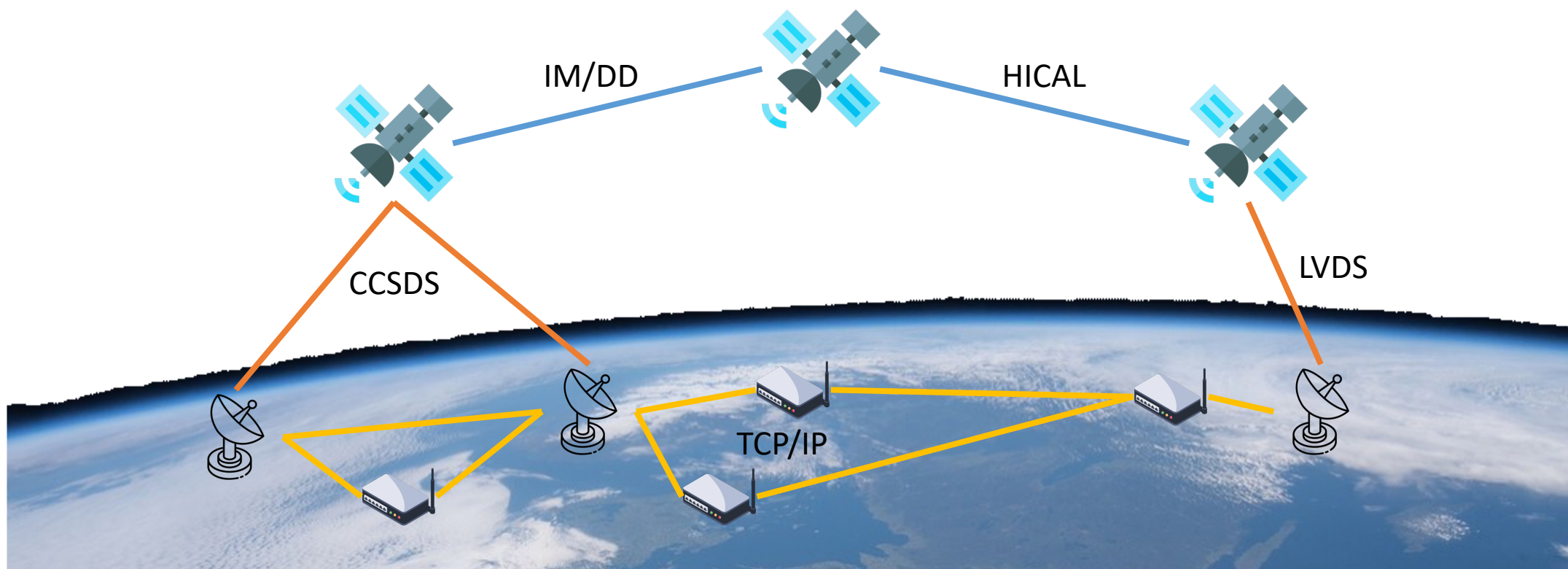
Robert Rose (SpaceX飞行软件部门主管) 于2013年 Embedded Linux Conference中的报告提及SpaceX大量使用Linux和软件层加固

[1] <https://lwn.net/Articles/540368/>

[2] [https://www.nasa.gov/sites/default/files/atoms/files/benefits-for-humanity\\_third.pdf](https://www.nasa.gov/sites/default/files/atoms/files/benefits-for-humanity_third.pdf)

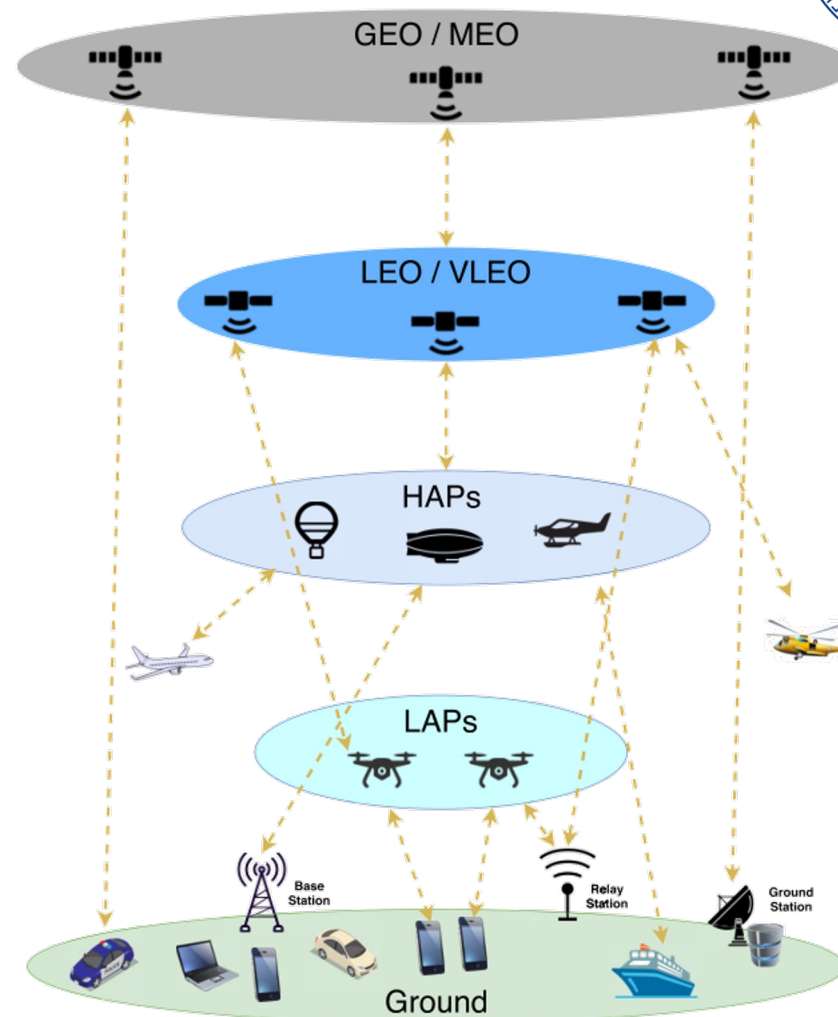
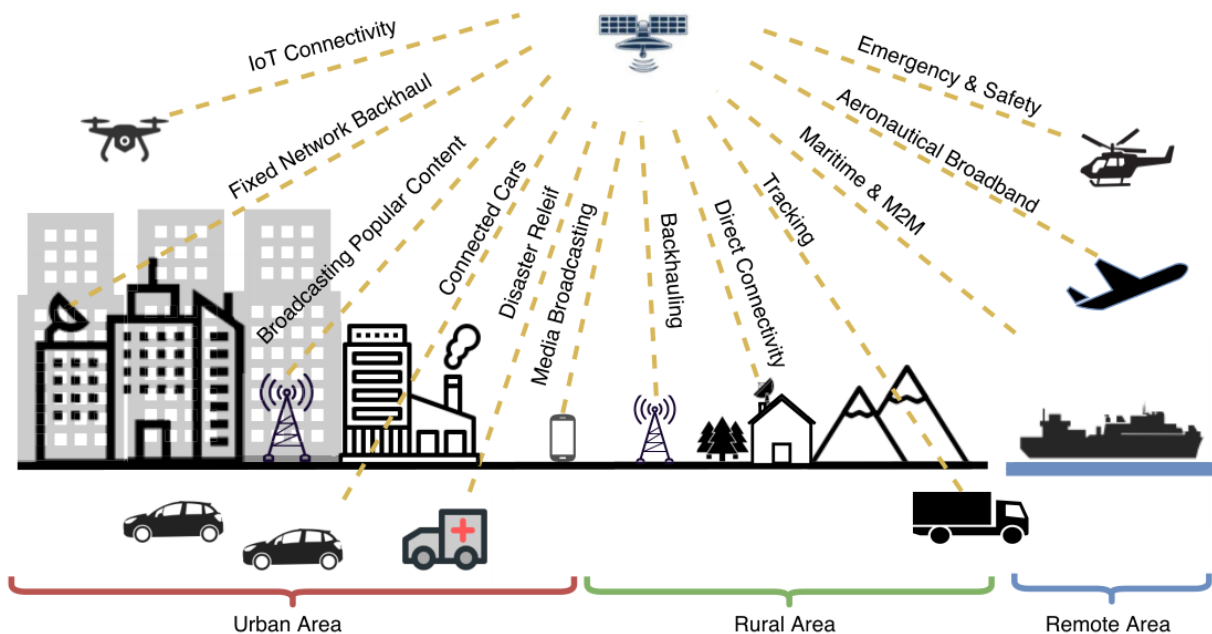
# 卫星计算挑战-4：架构融合

- 星座通信规范以及网络协议不统一，无法兼容适配现有应用



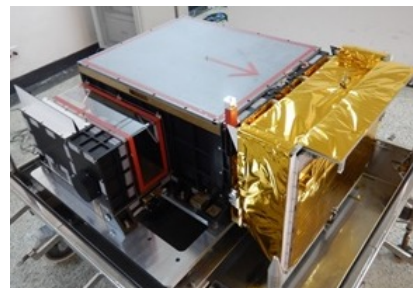
# 卫星计算挑战-4：架构融合

- 空天地一体化网络融合难
  - 5G与NTN网络融合
  - 低空、海洋、太空网络融合



# 卫星计算挑战-4：架构融合

- 卫星高度定制化，遥感、通信、导航卫星星星并存，应用部署不灵活，星座资源利用不充分。
- 星载设备架构、接口异构，资源认知难。
  - X86、ARM、RISC……
  - RS232、CAN、SPI……







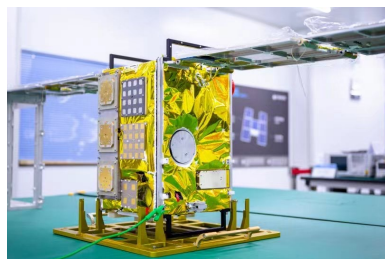
# 大纲

- 卫星技术发展
- 卫星计算需求
- 卫星计算挑战
- **卫星计算尝试**
  - 天算星座
  - 操作系统
  - 卫星云服务器

# 天算星座

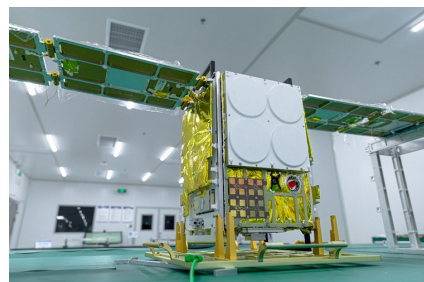
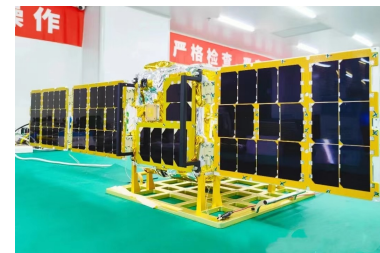
2020年06月，星际网络与智能计算实验室在深圳成立

2021年10月，北邮、天仪、华为云、北大、中移动等共建**天算星座**



**天算一号**：全球首颗具备云原生与边缘计算能力的智能卫星

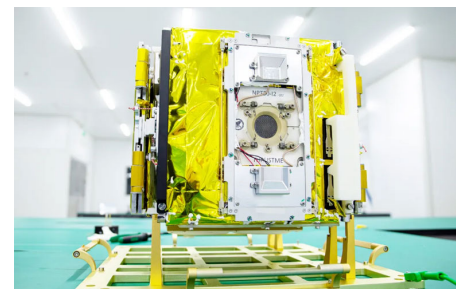
**北邮一号**：基于成熟云原生技术栈的载荷服务化、星地协同计算平台



**TY20卫星**：全球首次星载核心网部署



**天算二号**：全球首次QUIC协议在轨试验验证



**天算三号**：自主研发双内核卫星操作系统

# 天算星座

**定位：**面向国家需求，瞄准国际前沿，立足产学研用，坚持开放开源，争创国际领先，服务人类社会

## 卫星发展态势

- 卫星通用化
- 卫星智能化
- 卫星软件化

## 计算演进趋势

- 新型网络计算
- 融合服务计算
- 分布式AI计算

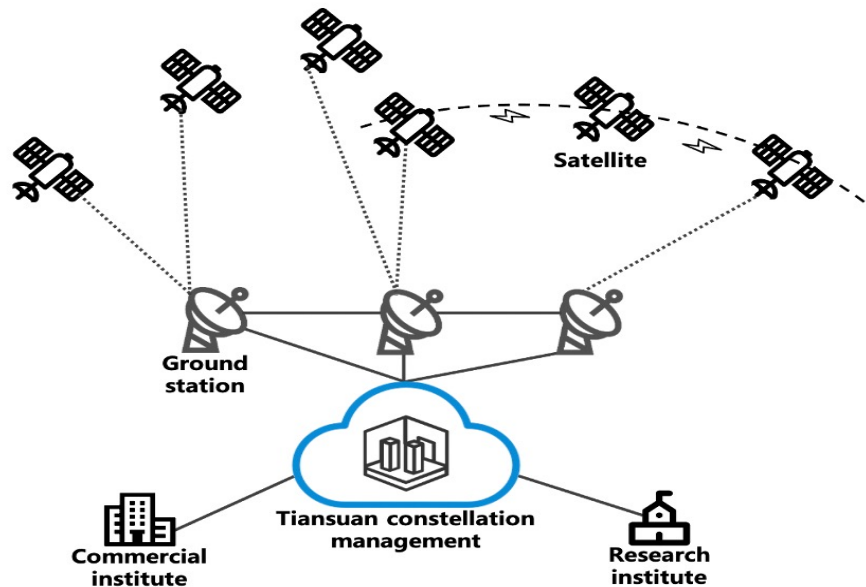
## 重点突破任务

- 6G核心网系统
- 卫星数联网
- 卫星操作系统
- 云原生卫星平台
- 器件与载荷搭载
- 星载服务能力开放

空天  
计算  
在轨  
试验  
开放  
开源  
平台

# 天算星座

主页: <http://www.tiansuan.org.cn/>



Communication

Computing

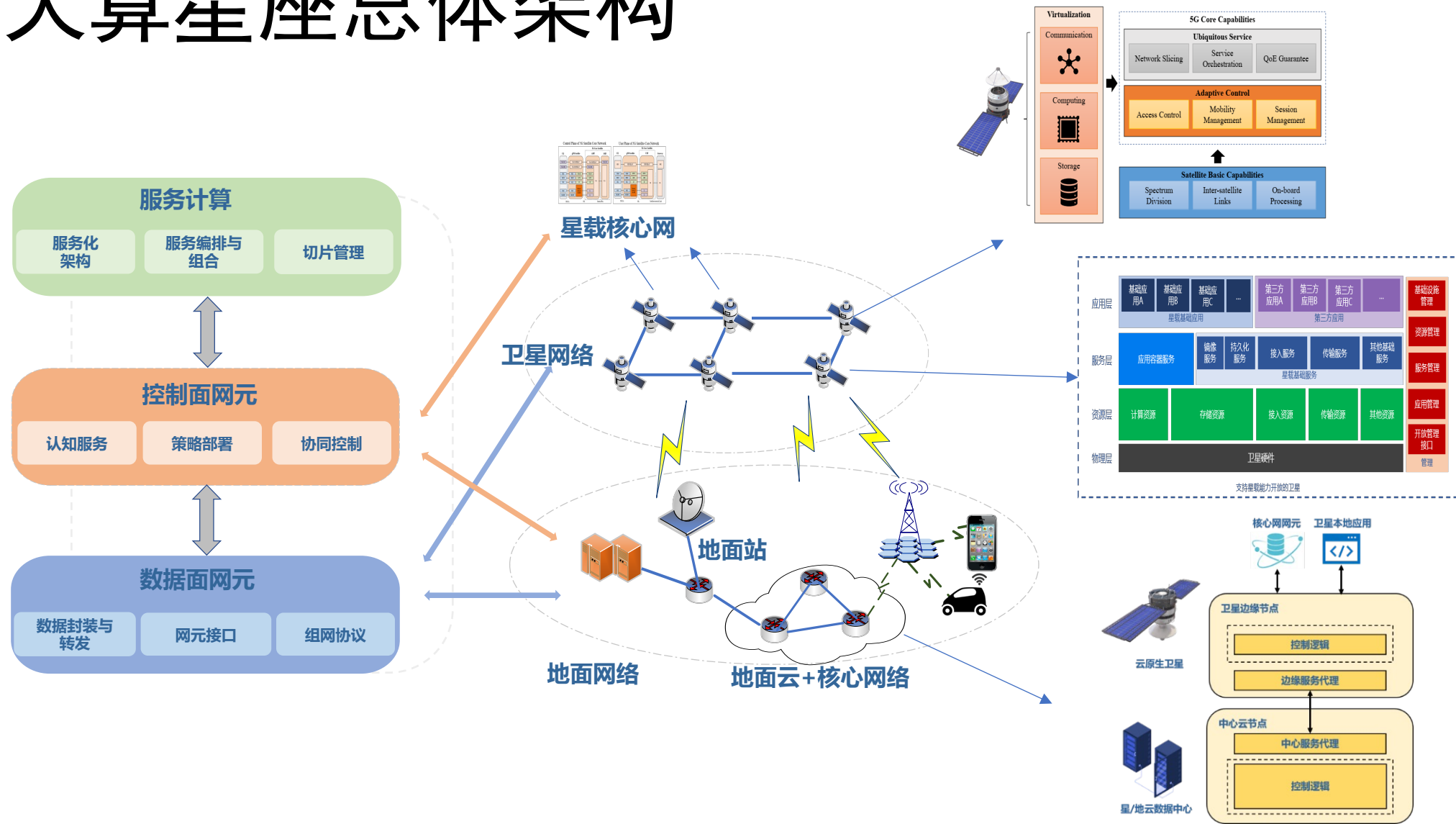
Satellite operating system

Security and reliability

Hardware testing

- Shanguang Wang, Qing Li, Mengwei Xu, Xiao Ma, Ao Zhou, Qibo Sun, Tiansuan Constellation: An Open Research Platform, Proc. IEEE EDGE 2021, Invited Paper.

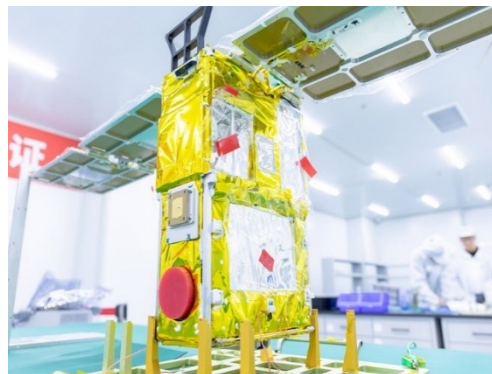
# 天算星座总体架构



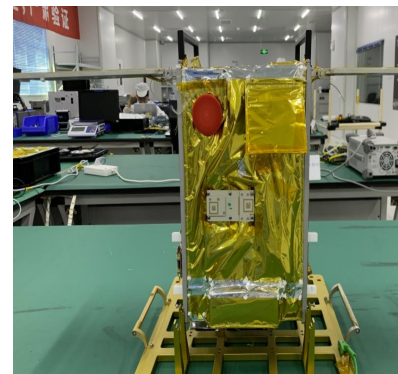
# 天算星座现状和规划



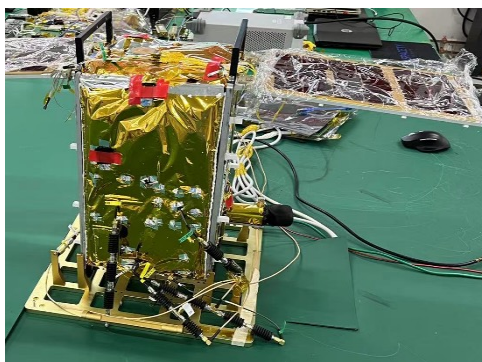
2021年12月7日 宝酝号



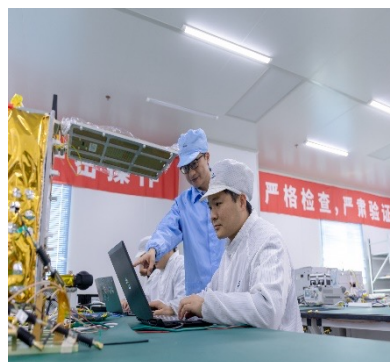
2022年2月27日创星雷神号



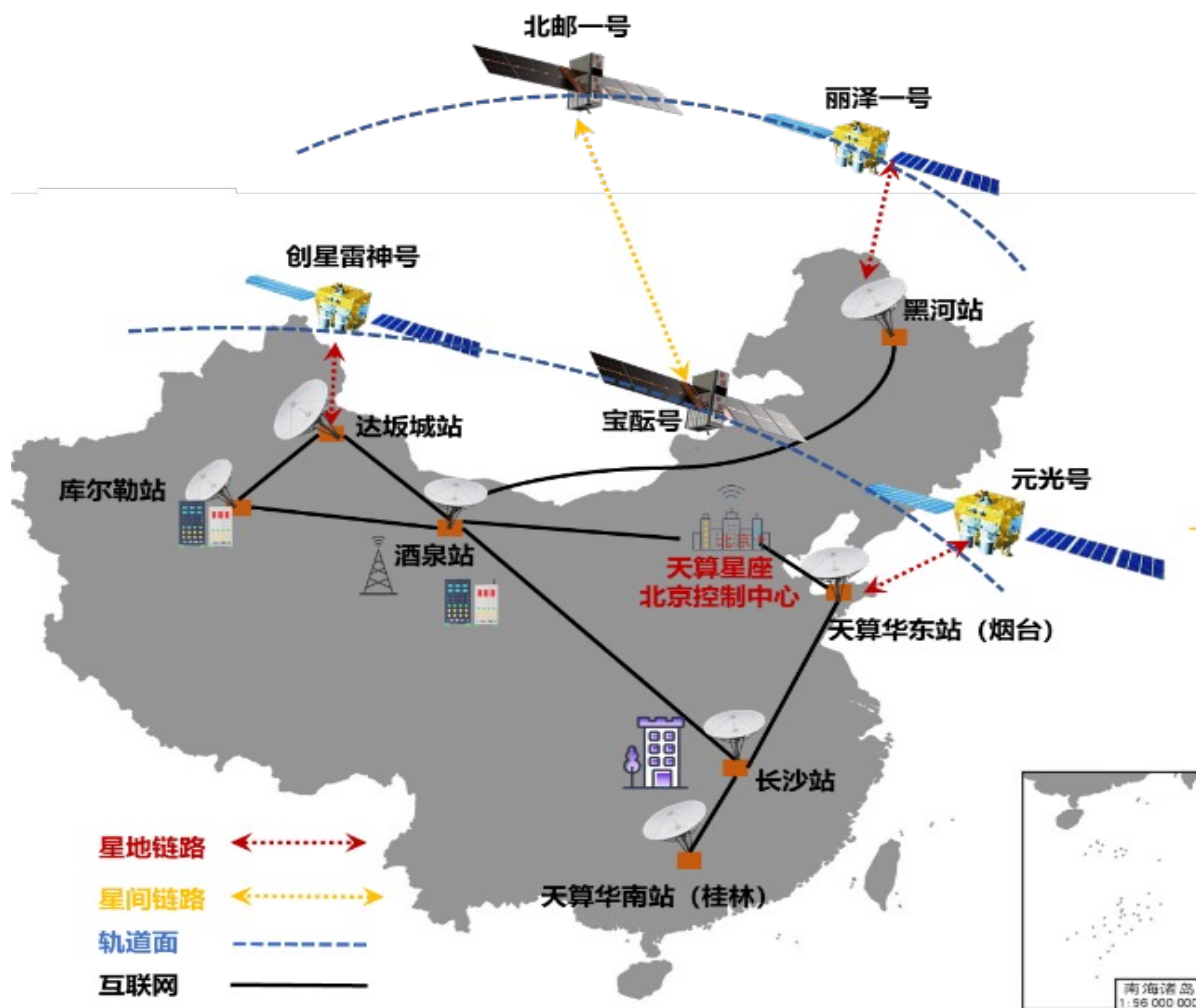
2022年12月14日 望齐州号



2023年1月16日 北邮一号



# 天算星座现状和规划：5星7站



# 天算星座试验

■ 平台能力让用户无感部署

■ <https://github.com/TiansuanConstellation/TiansuanOpenCodePlatform>

GitHub - TiansuanConstellation: x +  
github.com/TiansuanConstellation/TiansuanOpenCodePlatform

README.md

research-application model.

Tiansuan OpenCode Platform is a step forward to unit the forces of open source communities. We welcome creative members to join and together with us contribute novel solutions to satellite computing.

### Advantages

At present, Tiansuan Constellation has several satellites in orbit. We have established cooperative relations with numerous universities including Tsinghua, Peking, Beihang University and so on, and successfully carried many scientific research projects to the space.

We provide unified hardware equipment, common access methods and strong technical support for the thoroughly customized in-orbit research and experimental needs.

### How to contribute

If you would like to be one of us, please follow [this link](#). Tiansuan OpenCode Platform welcomes any organization or individual to contribute codes and build the Tiansuan Constellation together.

At the same time, Tiansuan Constellation also welcomes individuals, organizations and insitutions dedicated to sateliite research to use our platform for aerospace computing,

Tiansuan OpenCode Platform x +  
不安全 | tiansuan.org.cn/login

天仪研究院

### Platform-Login

username

password

Login



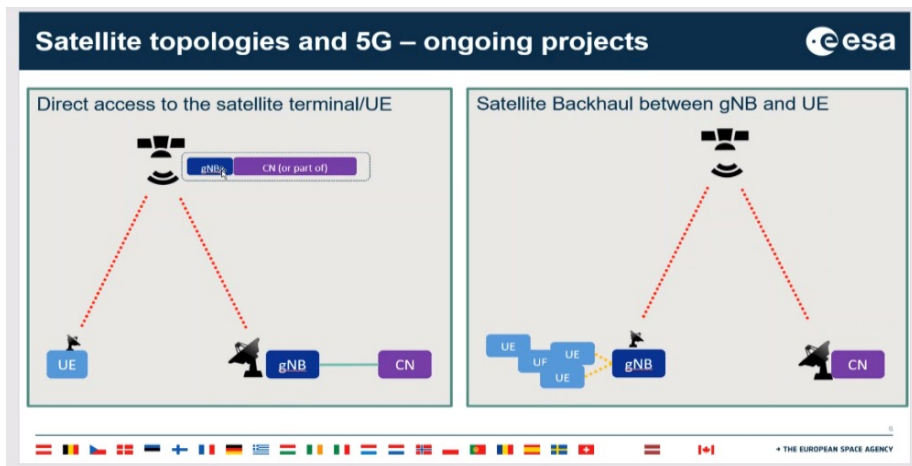
# 天算星座活动



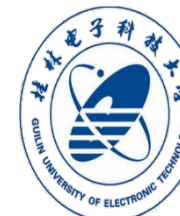
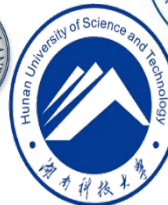
# 天算星座生态



- 欧洲航天局通信综合应用部 (ESA, TIA) 开展星载KubeEdge、星载核心网试验情况与天算星座开展合作
- 米兰大学签订合作协议
- 国内外20多个单位参与天算星座



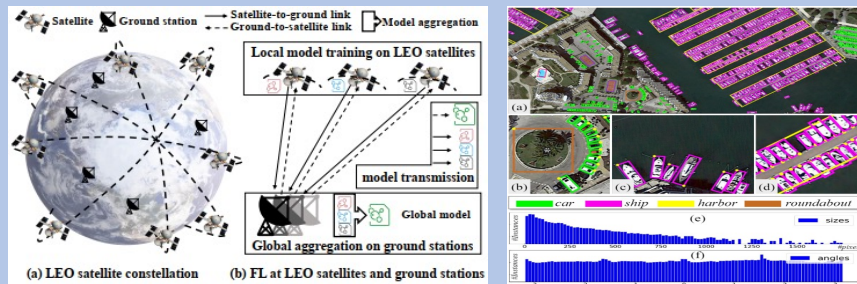
中国移动  
China Mobile



# 天算星座技术探索

## 应用

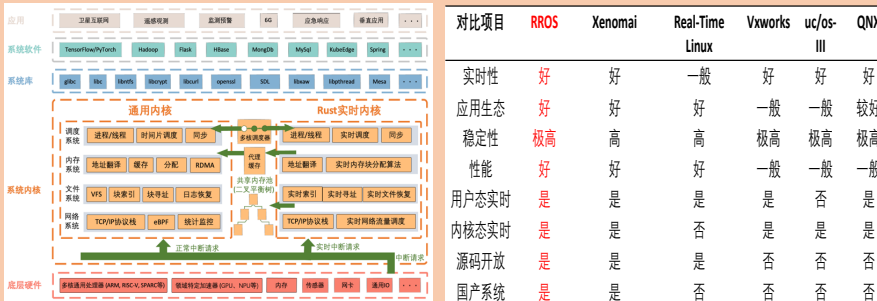
- 星地协同推理
- 卫星联邦学习
- 遥感数据压缩
- ○ ○



负责人  
王尚广

## 系统

- 卫星操作系统
- 星上虚拟化
- 云原生卫星
- 空天服务架构
- ○ ○



孙其博



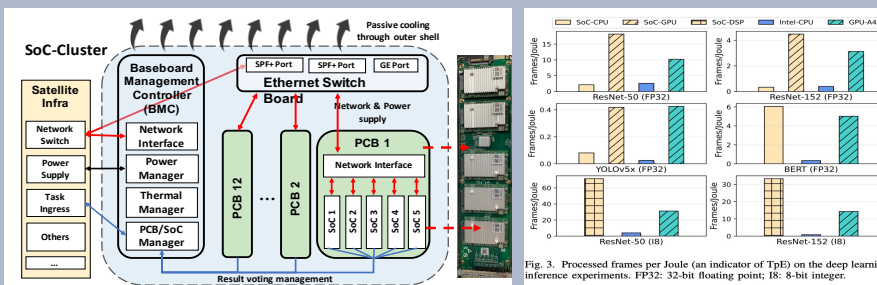
周傲



马骁

## 硬件

- 星载服务器架构
- 半实物卫星仿真
- 地面站服务共享
- ○ ○



张乙然



王鹏飞



徐梦炜

# CCF YOCSEF卫星计算工作组

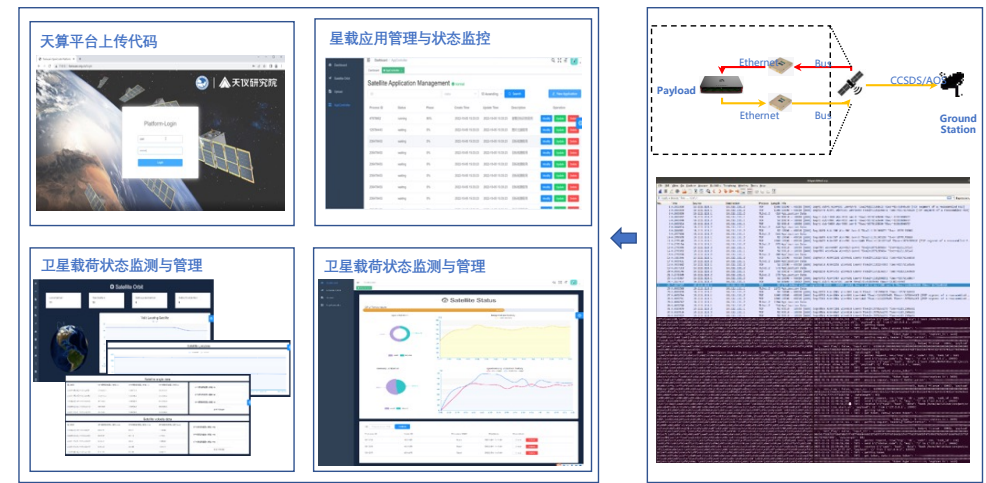
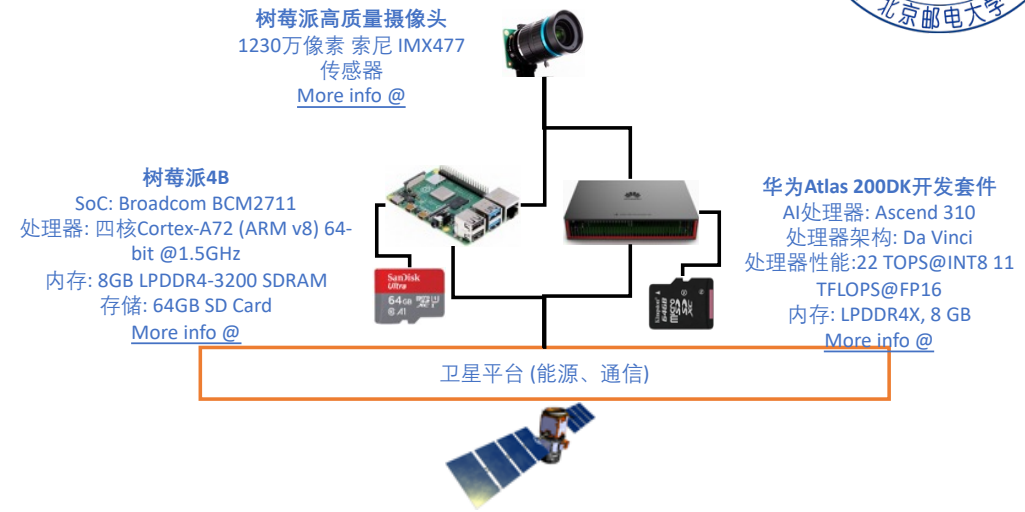
**提供4颗卫星：北邮一号、宝酝号、北邮二号和三号**

**面向科研院所**    **科研需求**  
**合作进步**

**面向普通单位**    **新器件、新载荷试验**  
**新方向、新技术**

**面向卫星企业**    **生态重要补充，助力行业发展**  
**减少拓展成本，积少成多**

**目标：CCF 卫星计算分会**





# 大纲

- 卫星技术发展
- 卫星计算需求
- 卫星计算挑战
- **卫星计算尝试**
  - 天算星座
  - **操作系统**
  - 卫星云服务器

# 背景

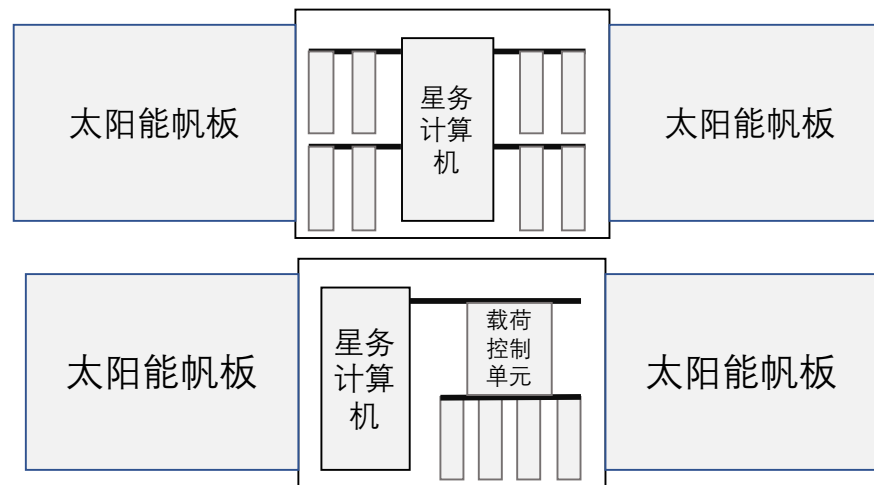
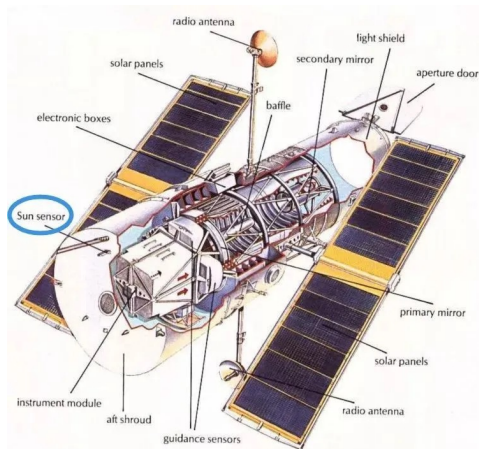
- 低轨卫星任务复杂化

- 传统卫星任务对实时性和稳定性有较高要求

- 星间/星地网络流量处理和转发，卫星姿态和轨道控制，星载传感器数据实时反馈处理

- 新兴卫星任务对通用性和性能有新的需求，需要良好的软件生态

- 遥感图像处理（编解码、物体识别等），太空科学实验





# 背景

- 低轨卫星任务复杂化

- 传统卫星任务对实时性和稳定性有较高要求

- 星间/星地网络流量处理和转发，卫星姿态和轨道控制，星载传感器数据实时反馈处理

- 新兴卫星任务对通用性和性能有新的需求，需要良好的软件生态

- 遥感图像处理（编解码、物体识别等），太空科学实验

- 案例-SpaceX 猎鹰火箭

- SpaceX猎鹰9号火箭采用了更先进的姿态控制算法，虽然已经通过算法优化将非凸优化问题转化为凸优化问题，利用凸优化的工具包求解，但是仍然需要100ms必须求解一次结果，这对于传统的实时操作系统性能提出了更高的要求；
- 卫星场景下，为了实现更精密的控制，高复杂度的算法将会逐渐增多，这类任务要求操作系统即有良好的通用性和性能，还需要在规定时间内完成求解；

# 动机

- 传统卫星操作系统难以同时满足通用化和实时性的需求



软件生态丰富，开发便捷灵敏



实时性保证



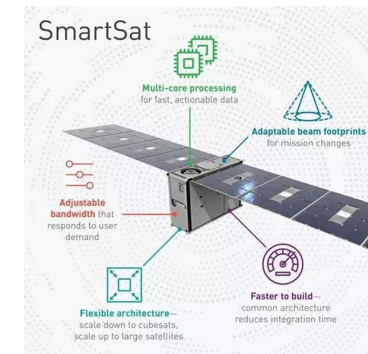


# 动机

- 已有的解决思路存在缺陷
  - Real-Time Linux
    - 对Linux内核采用Preempt-RT补丁
    - 缺点：只能做到软实时
  - POSIX兼容的实时内核
    - 编写一个兼容POSIX接口的实时内核
    - 缺点：对大型的项目兼容性不够好
  - Amp/虚拟机运行实时内核与Linux
    - 利用Amp/type-1的Hypervisor同时运行Linux kernel和RTOS
    - 缺点：两个内核之间隔离相对较重，通信/性能受到限制；资源分配静态

# 动机

- 已有的解决思路存在缺陷
  - SpaceX猎鹰火箭，星链卫星，dragon飞船，星链均采用COTS器件作为计算硬件，利用三模冗余提升可靠性，采用Preempt-Linux作为主要的操作系统<sup>[1]</sup>。
  - 天智卫星<sup>[2]</sup>计划采用Linux作为主要的操作系统
  - Smartsat<sup>[3]</sup>星座采用Hypervisor技术同时运行Linux和实时操作系统



[1] ELC: SpaceX lessons learned <https://lwn.net/Articles/540368/>

[2] <https://news.lockheedmartin.com/2019-03-20-Lockheed-Martins-First-Smart-Satellites-are-Tiny-with-Big-Missions>

[3] <https://max.book118.com/html/2020/1220/7152012022003033.shtm>

# RROS: 首个基于Rust的双内核OS

- RROS针对实时性和通用性难以兼顾的解决方案: **双内核**

- 整体架构

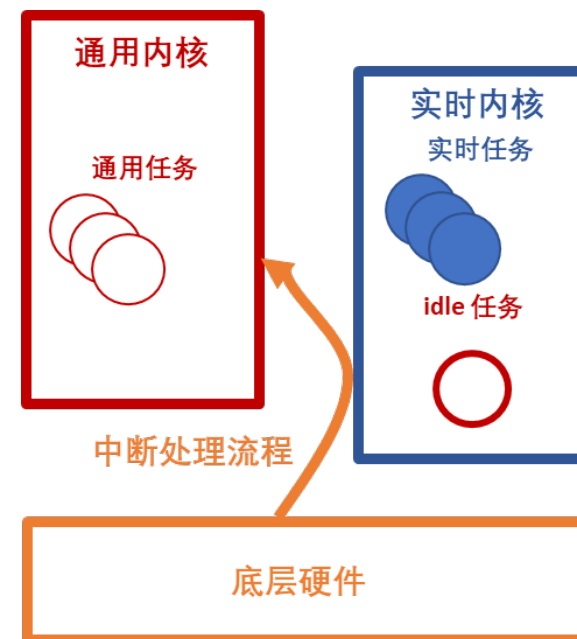
- 通用Linux内核提供通用性的支持
- 实时内核提供硬实时保证
- 采用中断虚拟化隔离内核, 减少开销

- 中断处理流程

- 底层硬件产生的中断首先发送到实时内核
- 若实时内核无对应中断处理函数再向通用内核转发

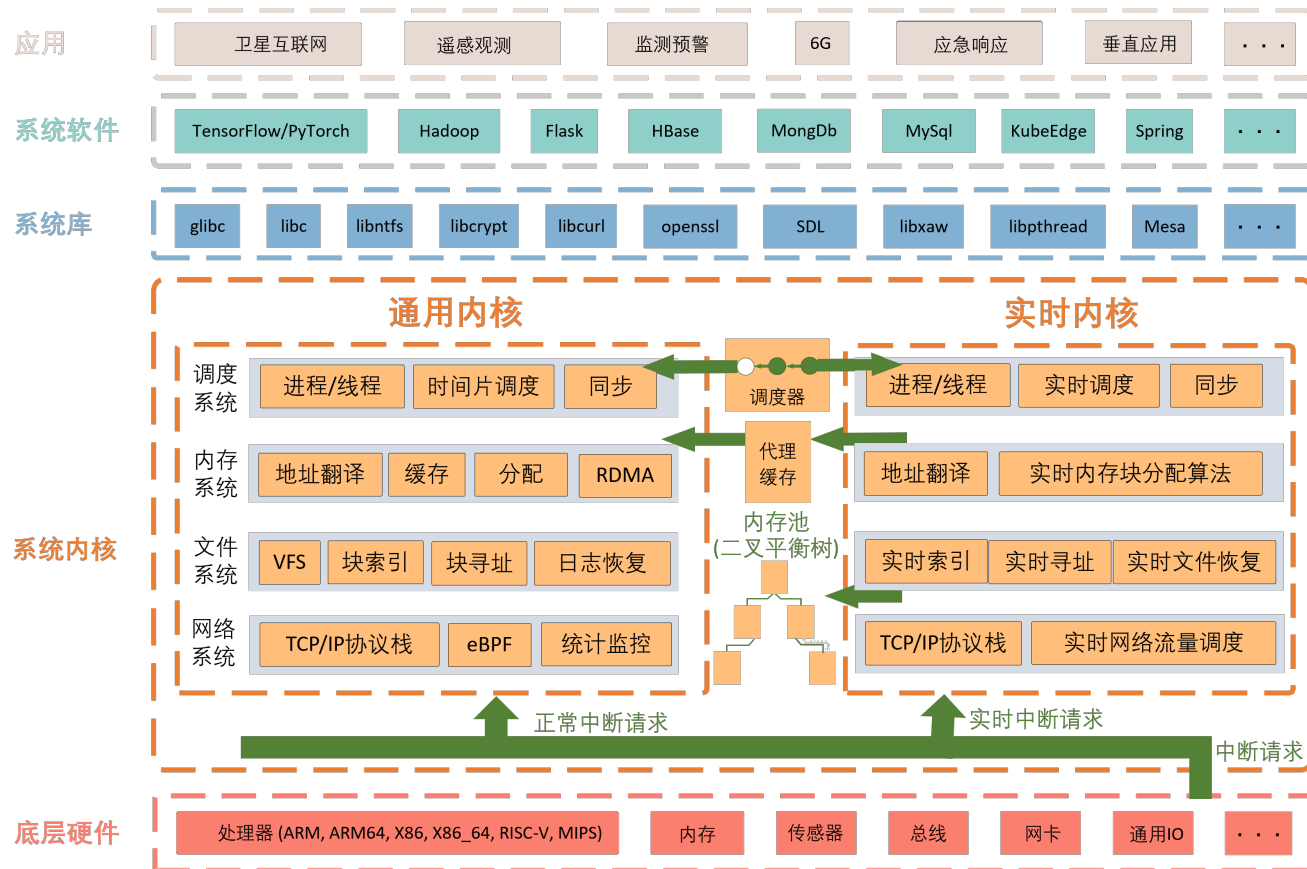
- 任务处理流程

- 如果实时内核有实时任务, 则优先处理实时任务
- 如果所有的实时任务都被处理完, 则处理idle任务
- 实时内核的idle任务即代表通用内核, 此时处理通用内核的通用任务



# RROS: 首个基于Rust的双内核OS

- RROS架构: 通用型内核Linux (基于C) + 实时内核 (基于Rust)



## Rust语言优势

- 内存安全
- 高性能
- 并发编程
- 生态完善

The EVL Core: <https://evlproject.org/core/>



# RROS: 首个基于Rust的双内核OS

- RROS针对部署场景受限的解决方案: 可伸缩性架构
  - 针对不同开发板的资源差异, 可以灵活调整Linux内核的大小
    - 对于资源丰富场景: 通用Linux
    - 对于资源一般场景: 剪裁后的Linux
    - 对于资源紧张场景: ucLinux

RROS架构	资源需求	内核大小	目标硬件
Linux+实时内核	高	15~30MB	Zynq, NVIDIA Jetson, etc
精简Linux+实时内核	中	3~7MB	RPI, HiFive Unmatched, etc
ucLinux+实时内核	低	400~700KB	STM32, 飞凌RT1052, etc



# RROS: 首个基于Rust的双内核OS

- 优势

- 实时性强

- 内核达到硬实时要求，并且兼容现有的rtos的接口，应用的迁移、维护和开发更为简单

- 通用性好

- 借助了Linux丰富的生态，可以直接部署tensorflow/k8s等复杂应用

- 稳定性高

- 基于Rust开发增强了内核的稳定性

- 隔离较轻

- 实时内核和Linux内核的交互简单，隔离相对较小

- 适配性好

- 采用可伸缩架构适配多种场景，同时提供多种实时接口便于迁移应用



# RROS: 首个基于Rust的双内核OS

- 实时操作系统接口

实时操作系统子系统	支持能力
线程	1. 支持fifo/rr/tp等多种实时调度算法 2. 提供优先级继承，天花板算法解决优先级反转问题
内存	支持动态分配内存的实时算法，保证灵活性和系统的实时性
文件	使用实时文件系统，提升数据保存的实时性
网络	适配udp协议栈，可以对udp报文实时处理
驱动	可以通过gpio等接口实时驱动外部设备，保证结果的正确、实时性



# RROS: 首个基于Rust的双内核OS

- 生态建设

- 对于Linux应用可以在无修改的情况下迁移适配
- 对于实时应用, 提供freertos, uc/os, vxworks等实时接口
- 适配x86/x86\_64/arm/arm64/Risc-V等多种架构
- 适配树莓派, 瑞芯微不同型号等多种开发板
- 内核适配范围广, 支持内核剪裁到150kb, 可以在2MB存储, 256KB内存的MCU上运行





# RROS: 首个基于Rust的双内核OS

- 前期性能测试

- 硬件平台: 树莓派4b+, 软件平台: Linux kernel 5.13
- 测试应用: Tensorflow 2.12, Kubernetes 1.22.15

对比项目	RROS	PREEMPT-Linux
Timer Latency	5ms	30ms
Network ping Latency	2ms	5ms
TensorFlow	Mobilenet v2	0.25s/img
	Resnet101	1.45s/img
	Vgg11	1.27s/img
	Vit_b_16	1.48s/img
Kubeedge Cloudcore setup	13s	12s

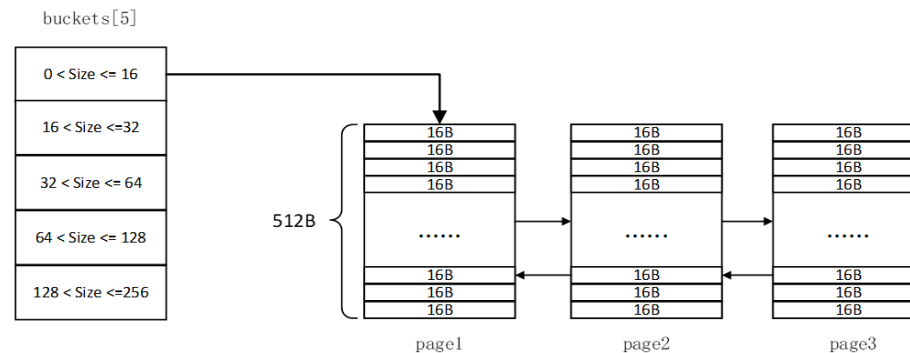


# RROS：首个基于Rust的双内核OS

对比项目	RROS	Xenomai	Real-Time Linux	Vxworks	uc/os-III	QNX	Free RTOS	NuttX	HarmonyOS
实时性	好	好	一般	好	好	好	好	好	一般
应用生态	好	好	好	一般	一般	较好	较好	好	较好
稳定性	极高	高	高	极高	极高	极高	高	高	高
性能	好	好	好	一般	一般	一般	好	一般	一般
用户态实时	是	是	是	是	否	是	是	是	是
内核态实时	是	是	否	是	是	是	是	是	是
源码开放	是	是	是	否	否	否	是	是	是
国产系统	是	否	否	否	否	否	否	否	是
适配性	好	一般	一般	一般	一般	一般	一般	好	好

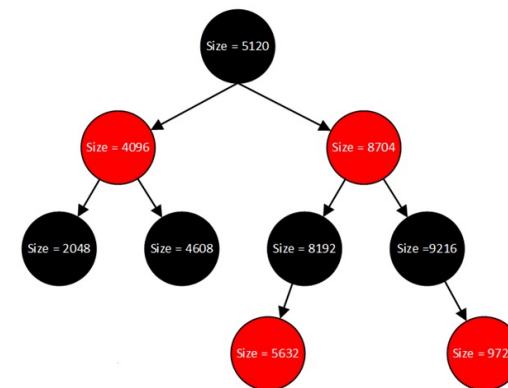
# RROS核心技术：以内存管理为例

- 问题提出
  - Linux内存管理系统：不实时，动态分配
  - 传统实时内存管理系统：实时，静态分配
- RROS针对动态内存管理不实时的解决方案：



小内存管理采用快速缓冲池 分配时间复杂度为 $O(1)$

不实时来源	解决方案
内存分配算法时间复杂度高	高效分配算法+预留空间/延迟释放
实时应用和通用应用不隔离	双内存池隔离实时和通用应用
Hard/Soft Page Fault	内核vmalloc分配
TLB/Cache miss	实时内核缓存动态保活

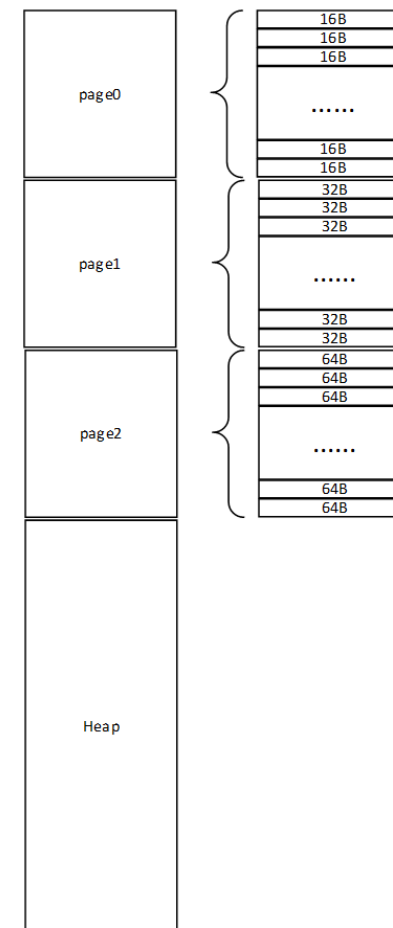


大内存管理直接从红黑树中查找 分配时间复杂度为 $O(\log n)$

# RROS核心技术：以内存管理为例

- 问题提出
  - Linux内存管理系统：不实时，动态分配
  - 传统实时内存管理系统：实时，静态分配
- RROS针对动态内存管理不实时的解决方案：

不实时来源	解决方案
内存分配算法时间复杂度高	高效分配算法+预留空间/延迟释放
实时应用和通用应用不隔离	双内存池隔离实时和通用应用
Hard/Soft Page Fault	内核vmalloc分配
TLB/Cache miss	实时内核缓存动态保活

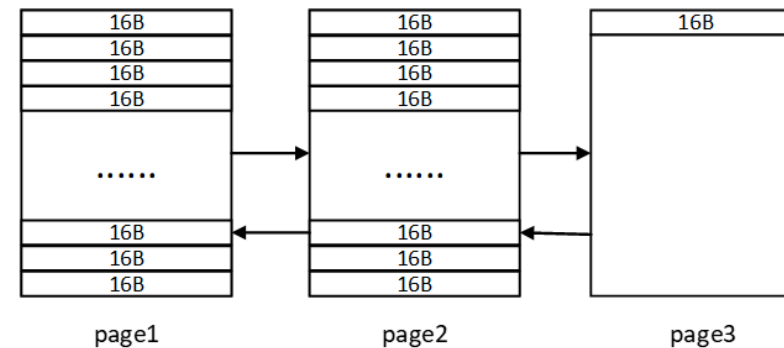


小内存管理分配时采用预留空间方法 提升实时性

# RROS核心技术：以内存管理为例

- 问题提出
  - Linux内存管理系统：不实时，动态分配
  - 传统实时内存管理系统：实时，静态分配
- RROS针对动态内存管理不实时的解决方案：

不实时来源	解决方案
内存分配算法时间复杂度高	高效分配算法+预留空间/延迟释放
实时应用和通用应用不隔离	双内存池隔离实时和通用应用
Hard/Soft Page Fault	内核vmalloc分配
TLB/Cache miss	实时内核缓存动态保活

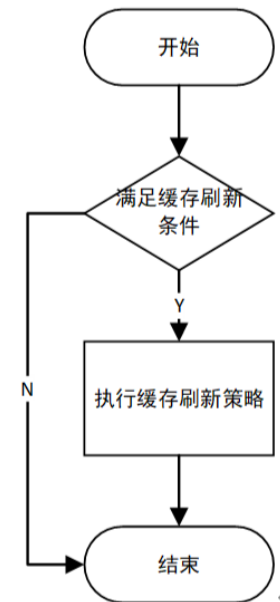


小内存管理释放时采用**延迟释放**方法 提升实时性

# RROS核心技术：以内存管理为例

- 问题提出
  - Linux内存管理系统：不实时，动态分配
  - 传统实时内存管理系统：实时，静态分配
- RROS针对动态内存管理不实时的解决方案：

不实时来源	解决方案
内存分配算法时间复杂度高	高效分配算法+预留空间/延迟释放
实时应用和通用应用不隔离	双内存池隔离实时和通用应用
Hard/Soft Page Fault	内核vmalloc分配
TLB/Cache miss	实时内核缓存动态保活



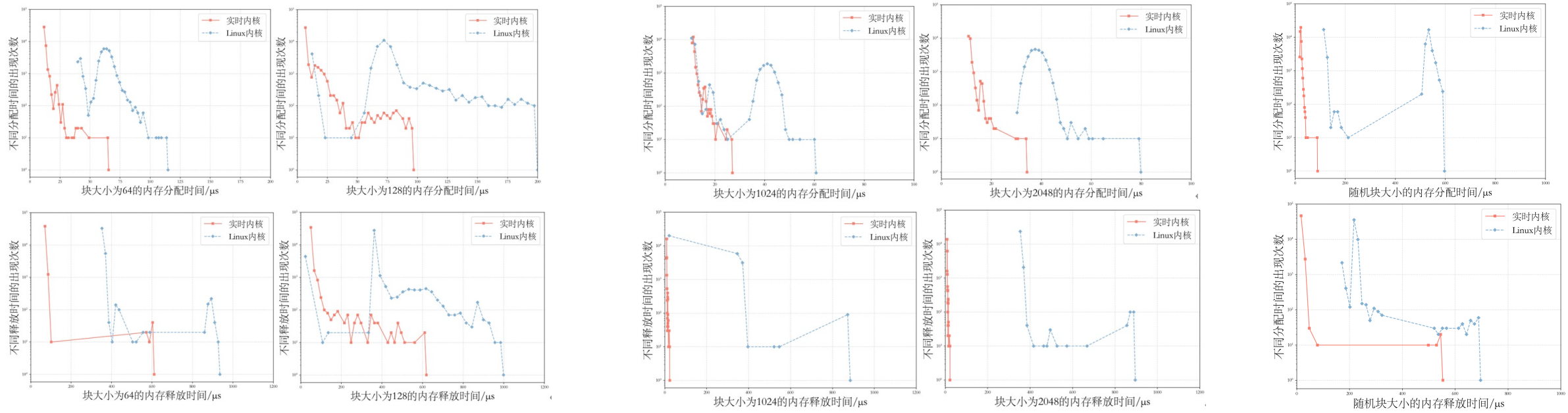
在通用内核执行任务时 对实时内核缓存保活

1. 对TLB刷新全部缓存，对Cache刷新部分缓存
2. 定期访问页表项提升计数
3. 编译期插入预读取指令 `__builtin_prefetch`，使数据常驻Cache

# RROS核心技术：以内存管理为例

- RROS动态内存管理方案性能测试
  - 小内存分配/释放性能测试
  - 大内存分配/释放性能测试
  - 随机大小分配/释放性能测试

策略	优化前 $T_{max}/\mu s$	优化后 $T_{max}/\mu s$	优化降低
预留空间	6.12	1.67	72.71%
延迟释放	548.41	287.74	47.54%
TLB优化	32.12	12.61	60.75%
Cache优化	28.15	8.12	71.16%





# 大纲

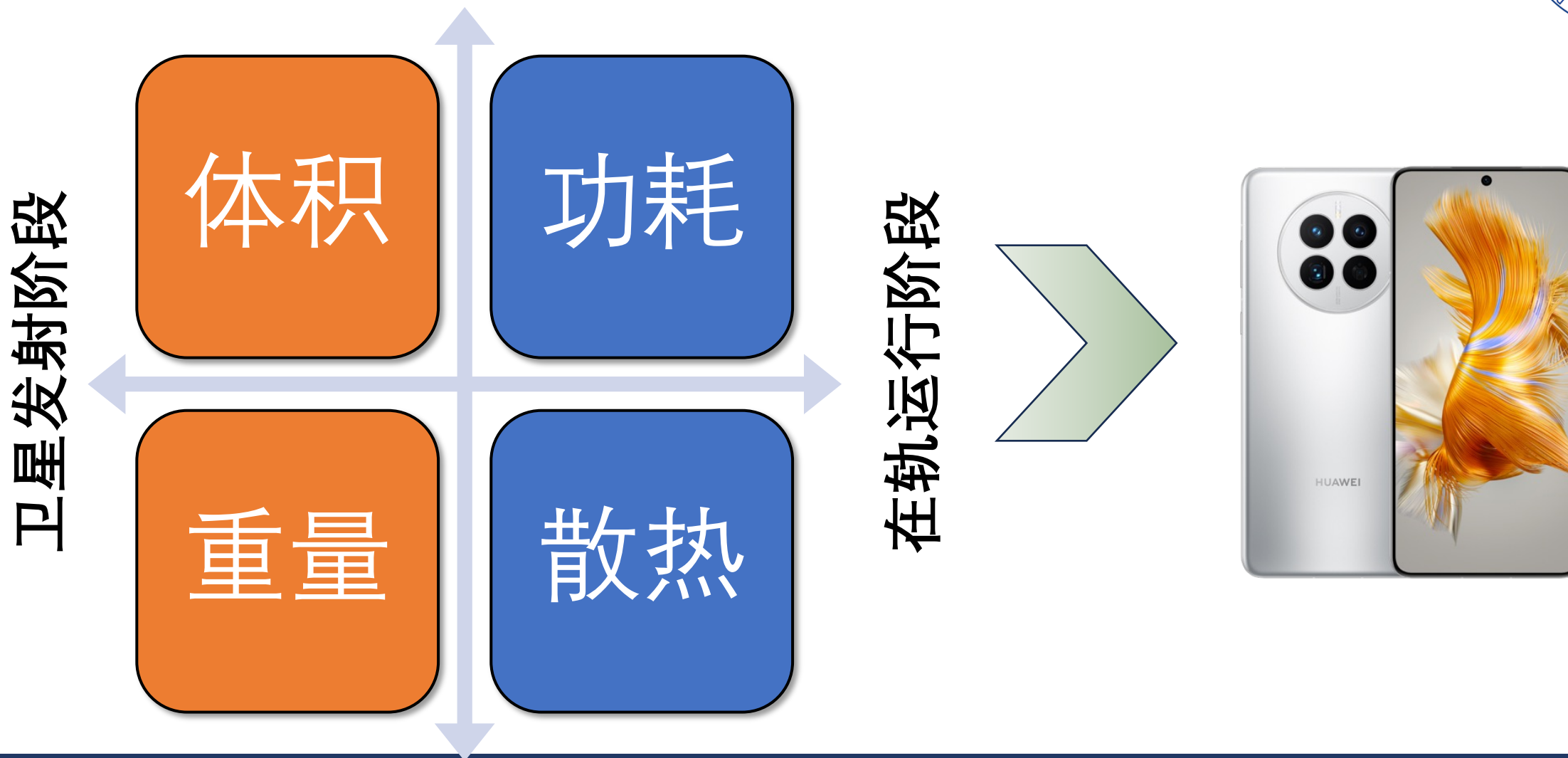
- 卫星技术发展
- 卫星计算需求
- 卫星计算挑战
- **卫星计算尝试**
  - 天算星座
  - 操作系统
  - **卫星云服务器**



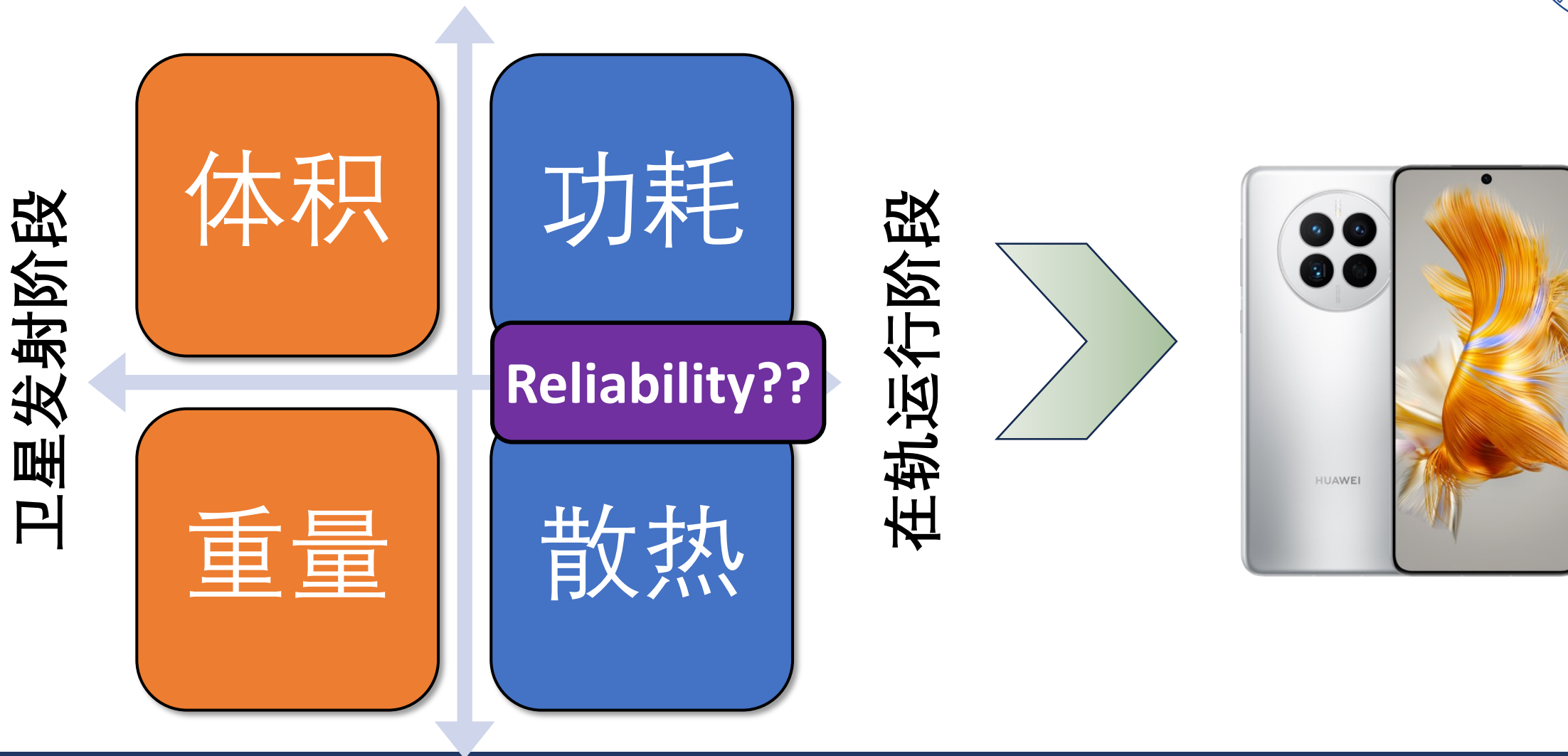
# 卫星计算制约因素



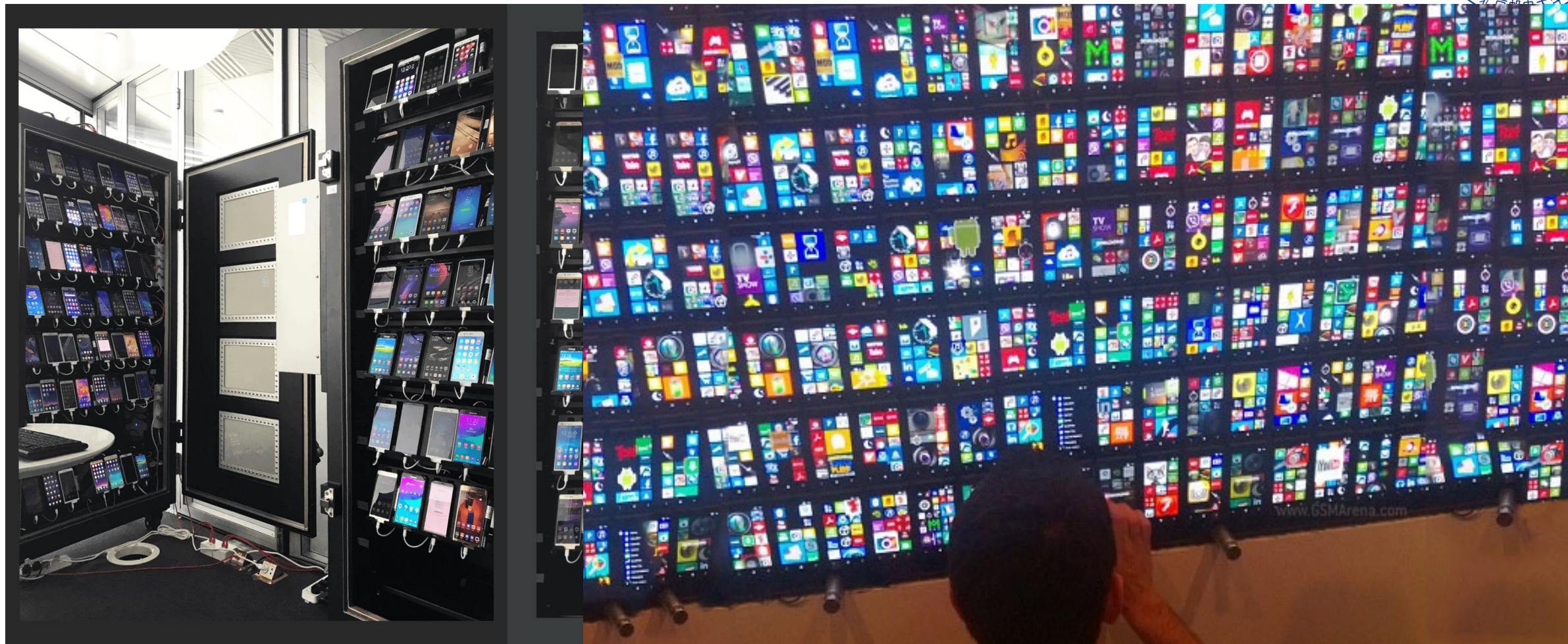
# 卫星计算制约因素



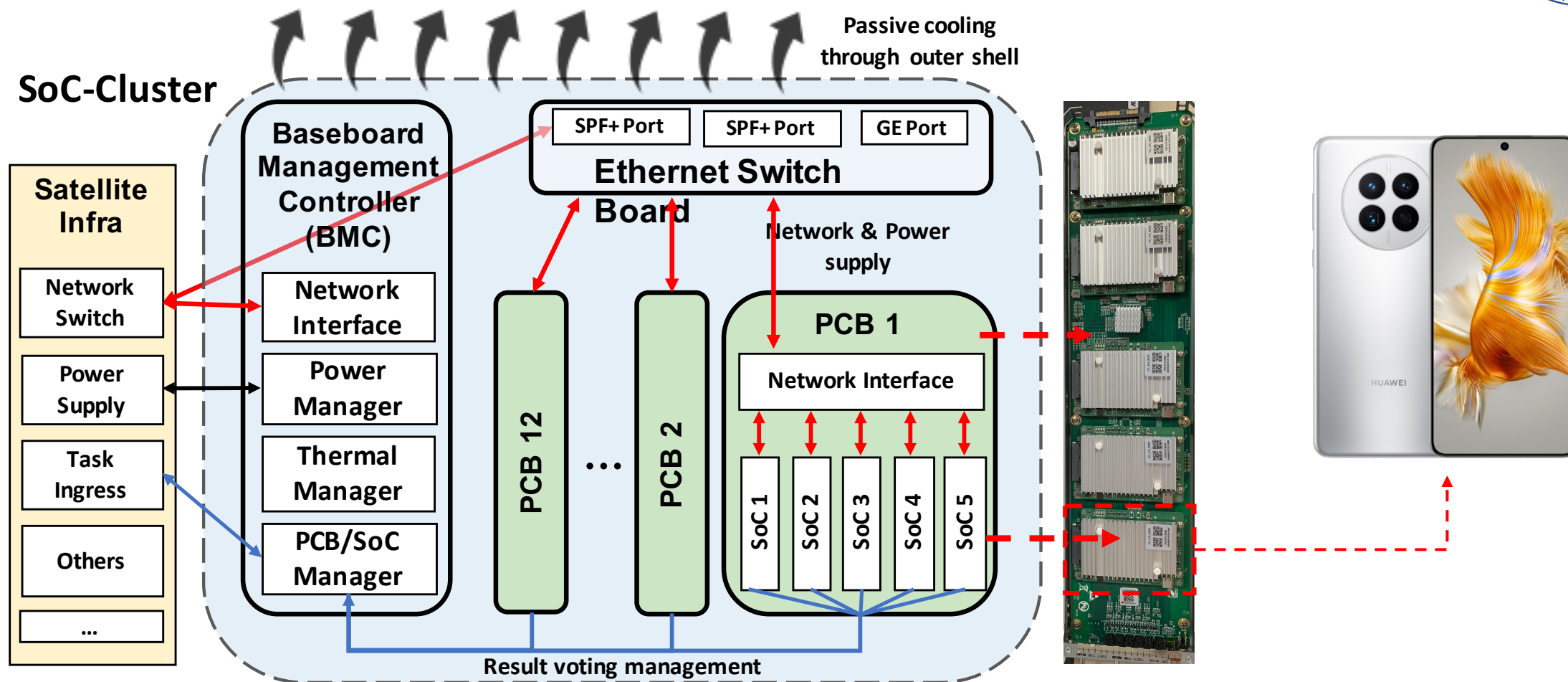
# 卫星计算制约因素



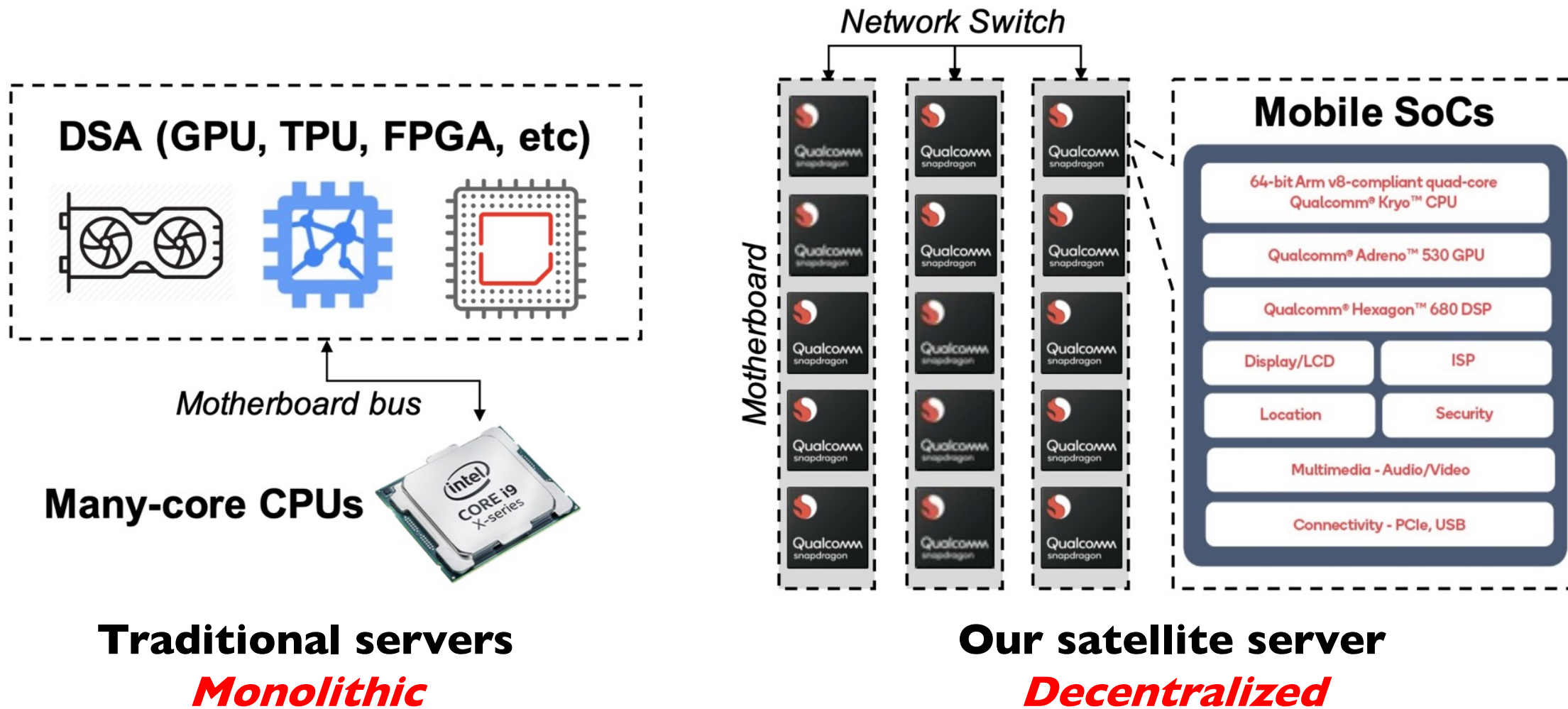
# 卫星计算：“手机云”？



# 卫星计算：ARM SoC阵列



# 卫星计算：ARM SoC阵列



# 理论性能对比

	Throughput per Energy (TpE)			Throughput per Volume (TpV)			Throughput per Weight (TpW)		
	Power (watt)	GFLOPs per watt (FP32)	GINOPs per watt (INT8)	Volume (U)	GFLOPs per U (FP32)	GINOPs per U (INT8)	Weight (kg)	GFLOPs per kg (FP32)	GINOPs per kg (INT8)
<i>Xeon 40-core CPU Server</i>	276.3	0.8	0.5	1	208.3	130.4	18.8	11.1	6.9
<i>NVIDIA A40 GPU Server</i>	2,000.0	149.6	1,197.2	4	74,800	598,600	57.9	5,165.8	41,339.8
<i>PowerEdge R350</i>	95.0	0.5	0.9	1	49.3	85.4	13.6	3.6	6.3
<i>PowerEdge R550</i>	330.0	0.5	0.9	2	83.0	151.3	20.4	8.1	14.8
<i>PowerEdge R750xs</i>	370.0	0.6	1.0	2	104.6	182.5	21.9	9.5	16.6
<i>SoC-Cluster (Kryo CPU)</i>	672.0	1.3	0.2	2	437.4	76.5	27.0	32.4	5.7
<i>SoC-Cluster (Adreno GPU)</i>	387.0	193.8	X	2	37,500	X	27.0	2,777.8	X
<i>SoC-Cluster (Hexagon DSP)</i>	345.5	X	2,604.9	2	X	450,000	27.0	X	33,333.3

TABLE I

THEORETICAL COMPARISON BETWEEN SoC-CLUSTER AND CONVENTIONAL COTS EDGE SERVERS. “X” MEANS THAT THIS NUMERICAL OPERATION IS NOT SUPPORTED BY THE HARDWARE.

- 与传统服务器相比，SoC阵列服务器在单位体积/重量/功耗下可以提供更多的算力

# 应用性能对比

- 视频编解码
  - Earth imagery pre-processing
  - Software: Ffmpeg & LiTr[1].
  - Datasets: 6 videos randomly picked from vbench
- 深度学习推理
  - Object detection, segmentation, etc..
  - Software: TVM@Intel CPU; TensorRT@NVIDIA GPU; TFLite@SoC
  - Models: ResNet-50, ResNet152, YOLOv5x, BERT
- Alternative hardware
  - 40-core Intel Xeon Gold 5218R
  - NVIDIA A40 GPU.





# 应用性能对比

- 视频编解码

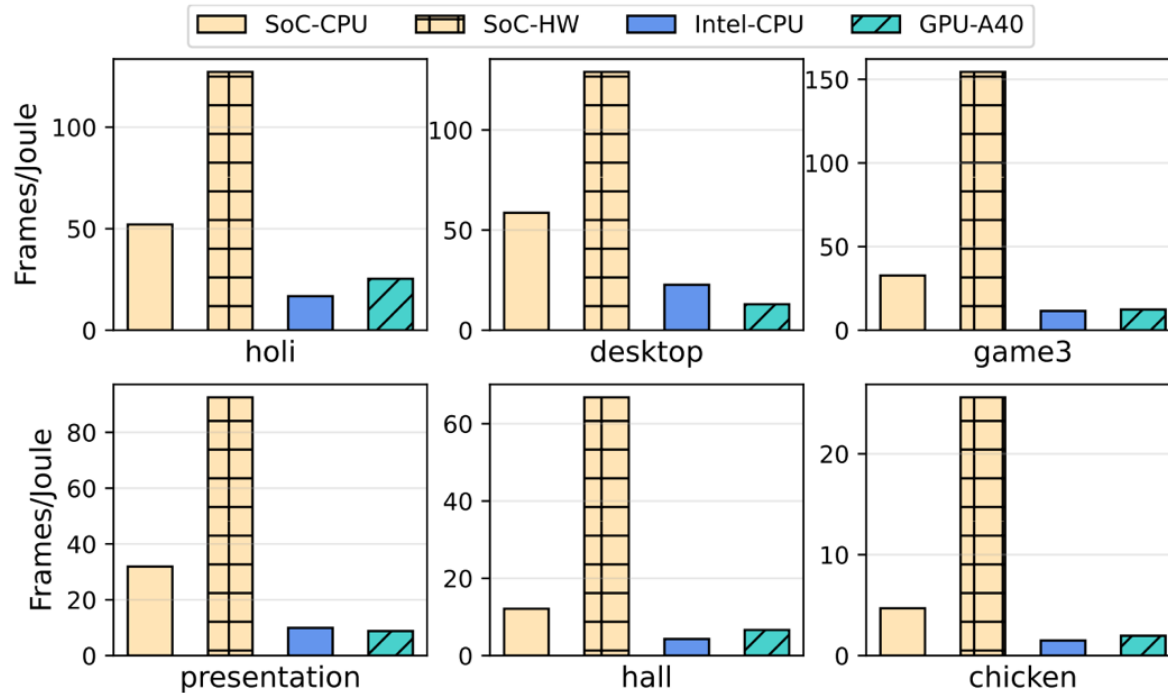


Fig. 4. Processed frames of per Joule (an indicator of TpE) on the video processing experiments. The 6 videos are randomly selected from a popular video benchmark [29].

- Our SoC servers can transcode 26–154 frames per Joule, which is  $5.7\times$ – $17.1\times$  higher than Intel CPU and  $5.0\times$ – $13.0\times$  higher than NVIDIA A40 GPU
- Brings benefit even without using its hardware codec

# 应用性能对比

- 深度学习推理

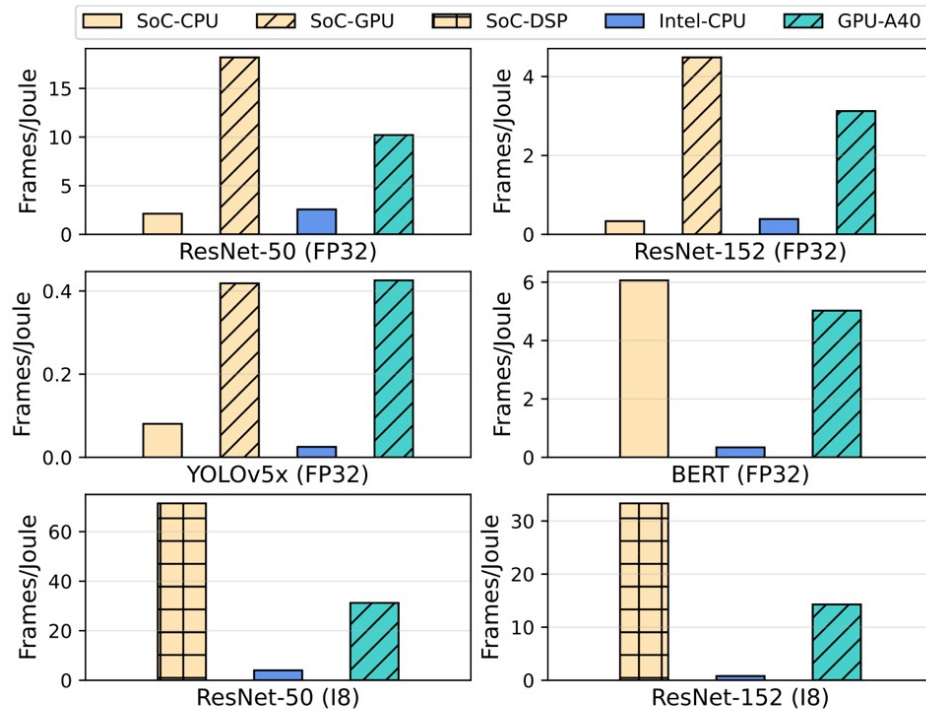


Fig. 3. Processed frames per Joule (an indicator of TpE) on the deep learning inference experiments. FP32: 32-bit floating point; I8: 8-bit integer.

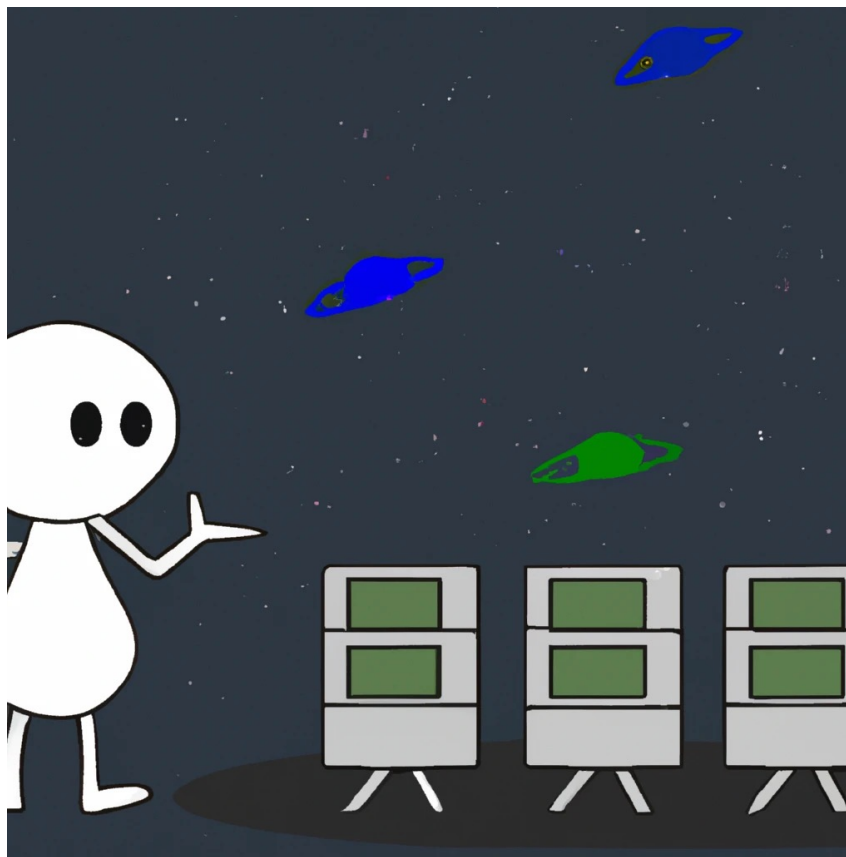
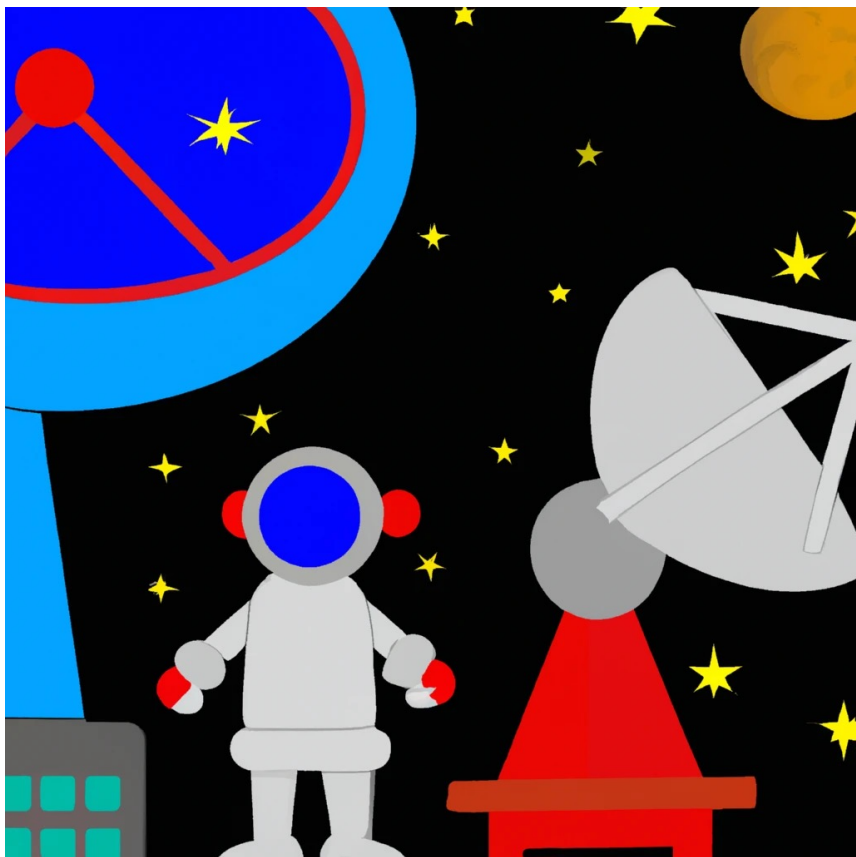
- Running prediction with ResNet-50 model (FP32), SoC GPU can process 18.2 samples per Joule, which is 7 $\times$  and 1.8 $\times$  higher than Intel CPU and NVIDIA GPU, respectively.
- The energy efficiency of SoC DSP is even more significant, i.e., 2.3 $\times$  higher than NVIDIA A40 GPU (with batch size 64).



# 太空服务器对比

Name	Launched Time	Hardware Platform	Process	General-purpose cores	DSA capacity	Other specs
HPE Spaceborn Computer-2	2021.02 (ISS)	2x HPE Converged EL4000 Edge system 2x HPE ProLiant DL360 server	14nm	64*2 cores + 28*2 cores	-	2*1U, 2*14KG, 2*800W; 2*1U, 2*17KG, 2*800W
北邮一号	2023.01	2x RPI-4B 2x Atlas 200DK	28nm,	8 cores + 10 cores	26GFLOPS + 16TFLOPS/32TINOPS	Small enough, ~3KG, 13W + 16W
天智一号						<27KG
星测未来						
RUAG Space	xxxx	Lynx Single Board Computer ARM processor with > 30000 DMIPS		4 cores for single Board	-	25W
Exo-Space FeatherEdge	2023	Quad Cortex-A53 CPU	~10nm	4 cores	4 TOPS	
Our SoC Server	2023	50x SoC (Snapdragon 865 & <a href="#">Rockchip RK3588</a> )	5nm – 8nm	400 cores	300TOPS (QS865: 62.5TFLOPS + 750TINOPS)	2U, 27KG, ~560W peak power

# 总结



把卫星变成带翅膀的PC：既要“人算”，又要“天算”

Image credits to: DALL-E of OpenAI