北京航空航天大學

Beijing University of Aeronautics and Astronautics

分布式OS概论

胡 凯

北航分布与移动计算实验室

hukai@buaa.edu.cn

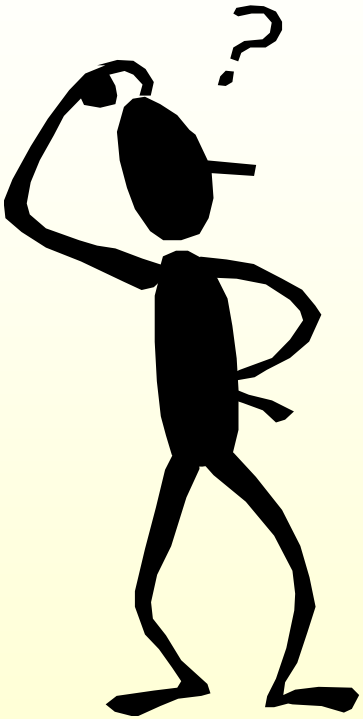
G1041 tel:82339460

计算机学院

课程目标

- 掌握分布式OS的基本原理
  - 计算模式，体系结构，方法，算法，关键问题
- 提高分布式研究与应用能力
  - 项目、实例、实验、研讨

培养多点创新思维和前沿意识



- **学：思索式**

1）过去、现在和未来

2）philosophy，trade off

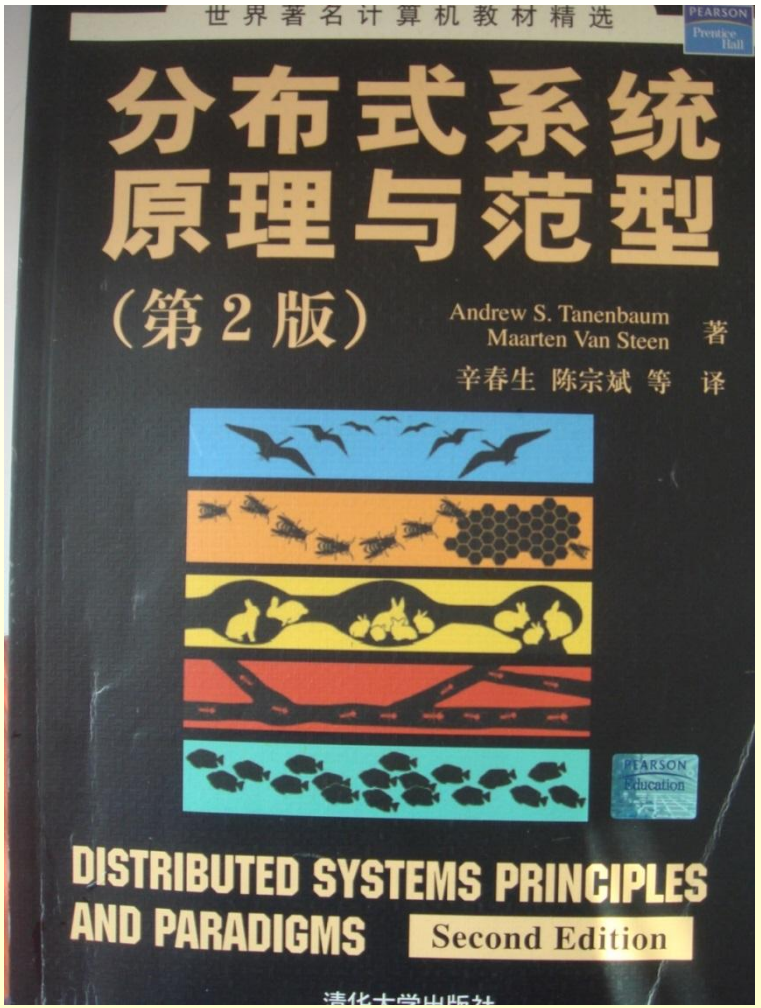
3）有的放矢的思考：

- **知道就行的知识点，**
- **需要较深入的知识点，**
- **感兴趣的研究点，**

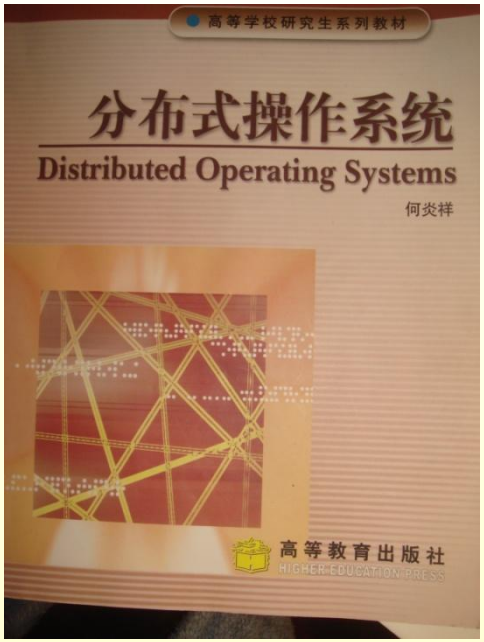
广度和深度具有辨正关系！

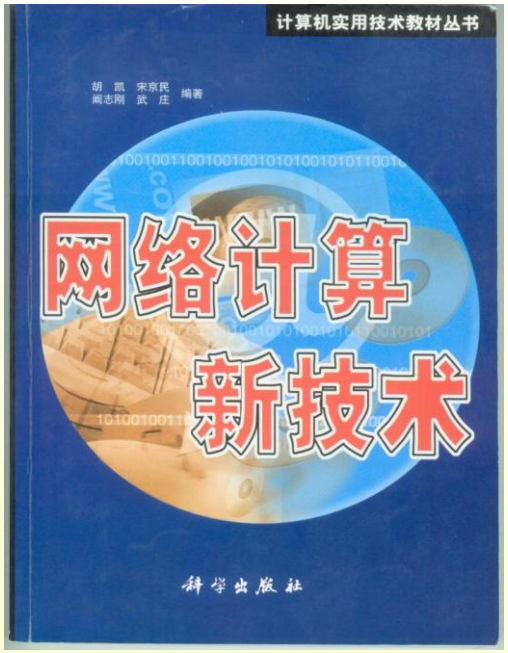
教材

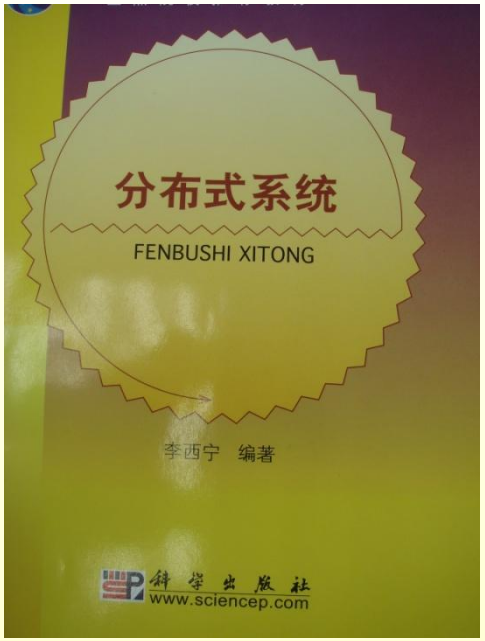

普通高等教育“十一五”国家级规划教材
重点大学计算机专业系列教材
分布式计算系统导论
——原理与组成
胡建平 胡凯 编著
清华大学出版社


世界著名计算机教材精选
PEARSON
Prentice Hall
分布式系统
原理与范型
(第2版)
Andrew S. Tanenbaum
Maarten Van Steen 著
辛春生 陈宗斌 等 译
PEARSON
Education
DISTRIBUTED SYSTEMS PRINCIPLES
AND PARADIGMS
Second Edition

参考书


高等学校研究生系列教材
分布式操作系统
Distributed Operating Systems
何炎祥
高等教育出版社
HIGHER EDUCATION PRESS


计算机实用技术教材丛书
胡 凯 宋京民
阚志刚 武 庄 编著
网络计算
新技术
科学出版社


分布式系统
FENBUSHI XITONG
李西宁 编著
科学出版社
www.sciencep.com

参考资料(1)

- Andrew s. Tanenbaum, Distibuted Operating system, 北京：清华大学出版社，影印版 1997.8 ，（注：分布式系统代表作）
- Andrzej Goscinski, Distributed Operating System The Logical Design, Addison-Wesley Publisher Ltd, 1991（分布式系统经典作）
- Lou Baker,Bradley J.Smith, Parallel Programming, McGraw-Hill ,1996
- Scheduling and Load Balancing in Parallel and Distributed System I ,II, III 卷
- Gregory F.Pfiste, In Search of Cluster, Second Edition, Prentice Hall PTR Upper Saddle River Nj, 1998.（Cluster的代表作）
- David E. Culler etc, Parallel Computer Architecture 机械工业出版社 1999
- M.L.Liu, Distributed Computing,2004,清华大学出版社
- Nancy A.Lynch, Distributed Algorithms 机械工业出版社

参考资料(2)

- Glen Bruce等，李如豹等译，分布式计算的安全原理，机械工业出版社。
- 李西宁，分布式系统，科学出版社,2006.12
- Hagit Attiya等 分布式计算（第二版），电子工业出版社，2008.4
- Nancy等，分布式算法，机械工业出版社，2004.1
- 何炎祥 分布式操作系统，高等教育出版社，2005.1

课程研讨

目标：开阔视野，启迪思路，增强动手能力

要求：具体、深入到研究点

提交：设计程序或技术报告到课程FTP
（课程结束前提交）

研讨：PPT演讲与系统设计

课程下载FTP： 219.224.169.136 ； port:21

课件下载（软件，文档，课件等）：

用户名，口令：cluster

作业上载：

用户名，口令：cgjobs

上载文件格式：压缩包

文件名：姓名 + 学号

课程考核

课时、学分：32学时（1～16周）

- 先修课程：操作系统，计算机网络
- 授课方法：课堂讲授＋互动
- 考核方式：平时＋研讨＋考试
  - 平时（15%）
  - 研讨（25%）
  - 考试（60%）

转变：互动讨论＋动手

无故缺课三次以上（含三次）取消考试资格

请假条需要导师签字

Any Question?

Thinking

- 什么是分布式系统（OS）？
- 为什么会发展出这样的模式？
- 它的特征或优势是什么？
- 它主要内容是什么？
- 它面临的问题是什么？
- 它下一步发展的方向是什么？
- 你要关心的细节是什么？


FIGHTER FORCE TODAY AND IN 2025
A-10/OA-10
F-15A/B/C/D
F-15E
F-16A/B
F-16C/D
F-117
F/A-18A/B
F/A-18C/D
F/A-18E/F
F-14D
F/A-18C/D
AV-8B
Harrier GR-7
JSF CTOL
F-22
F/A-18E/F
JSF CV
JSF STOVL
JSF STOVL
Joint-ness Is the Key To Affordability

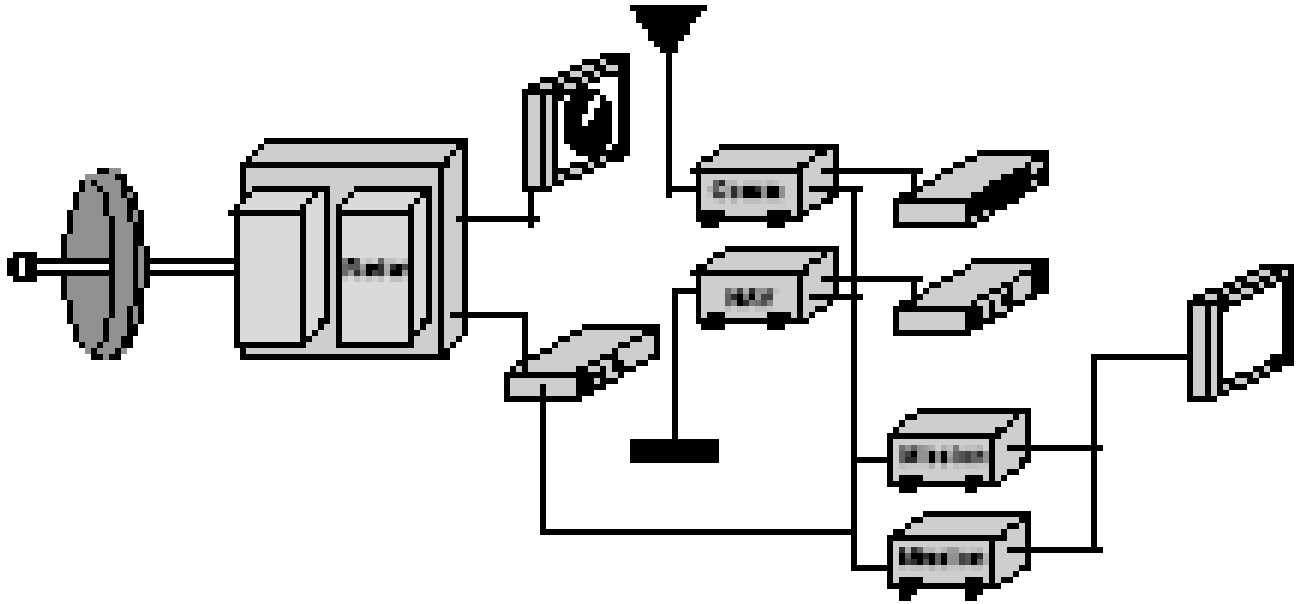
联邦式系统结构

由1553B总线互连,

每个黑盒子都有其A/D变换器,处理器等,是一个独立的数据处理单元

- 综合程度低,主要是综合显示
- 使用情况: F-16,F-18,歼-10 等
- 实例: F-16; F-18

联邦式系统结构

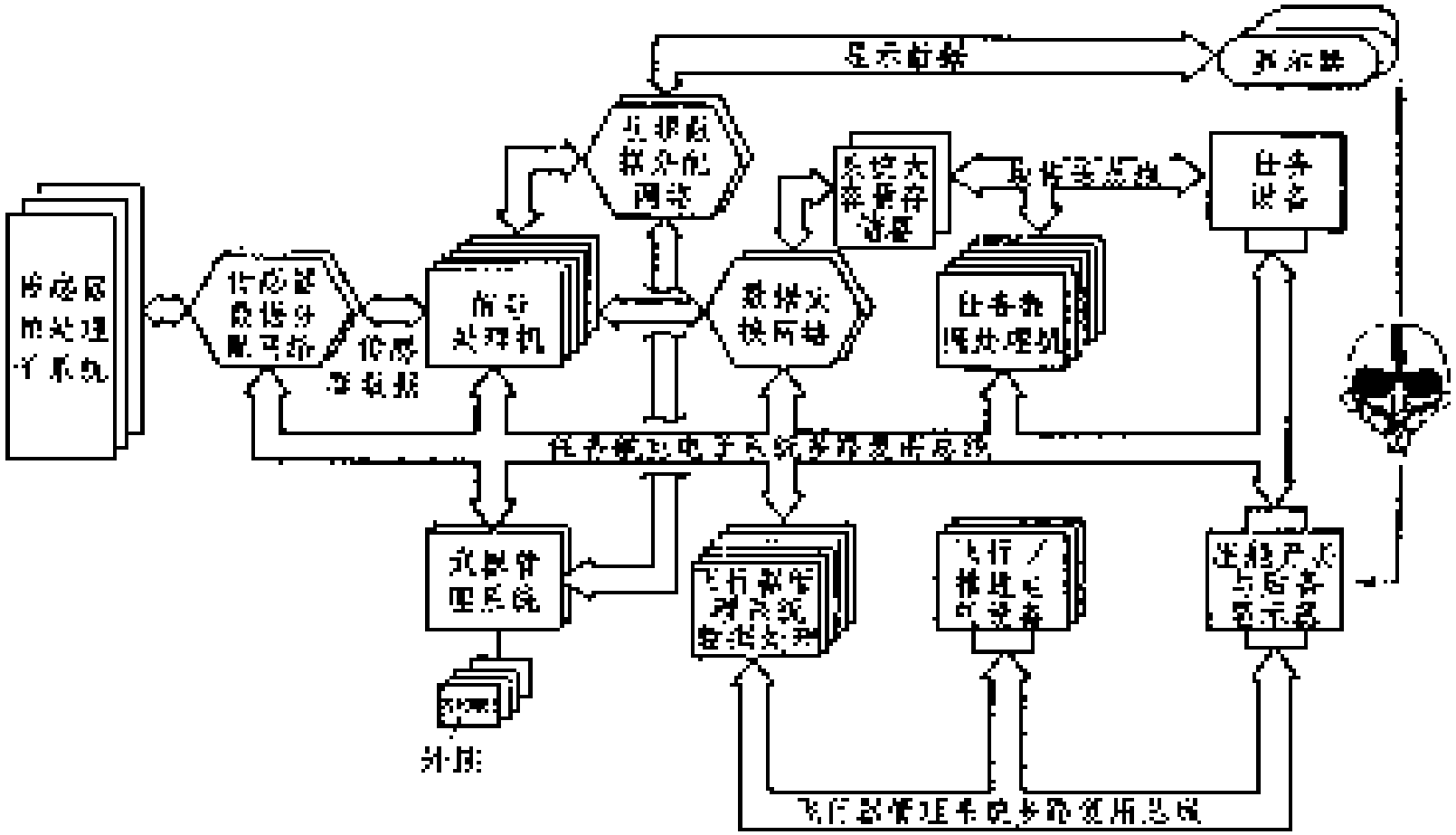


Federated Avionics
(60's - 70's)

综合化航电系统结构

- Pave Pillar (宝石柱)计划 (1980年代初)
  - 功能区划分: 传感器区,任务处理区,外挂物管理区,飞机控制区和控制显示区
  - 互连技术: 传感器光电交换网,显示视频交换网,高速数据传输总线HSDB
  - 模块化设计: 现场可更换模块结构,实现二级维护

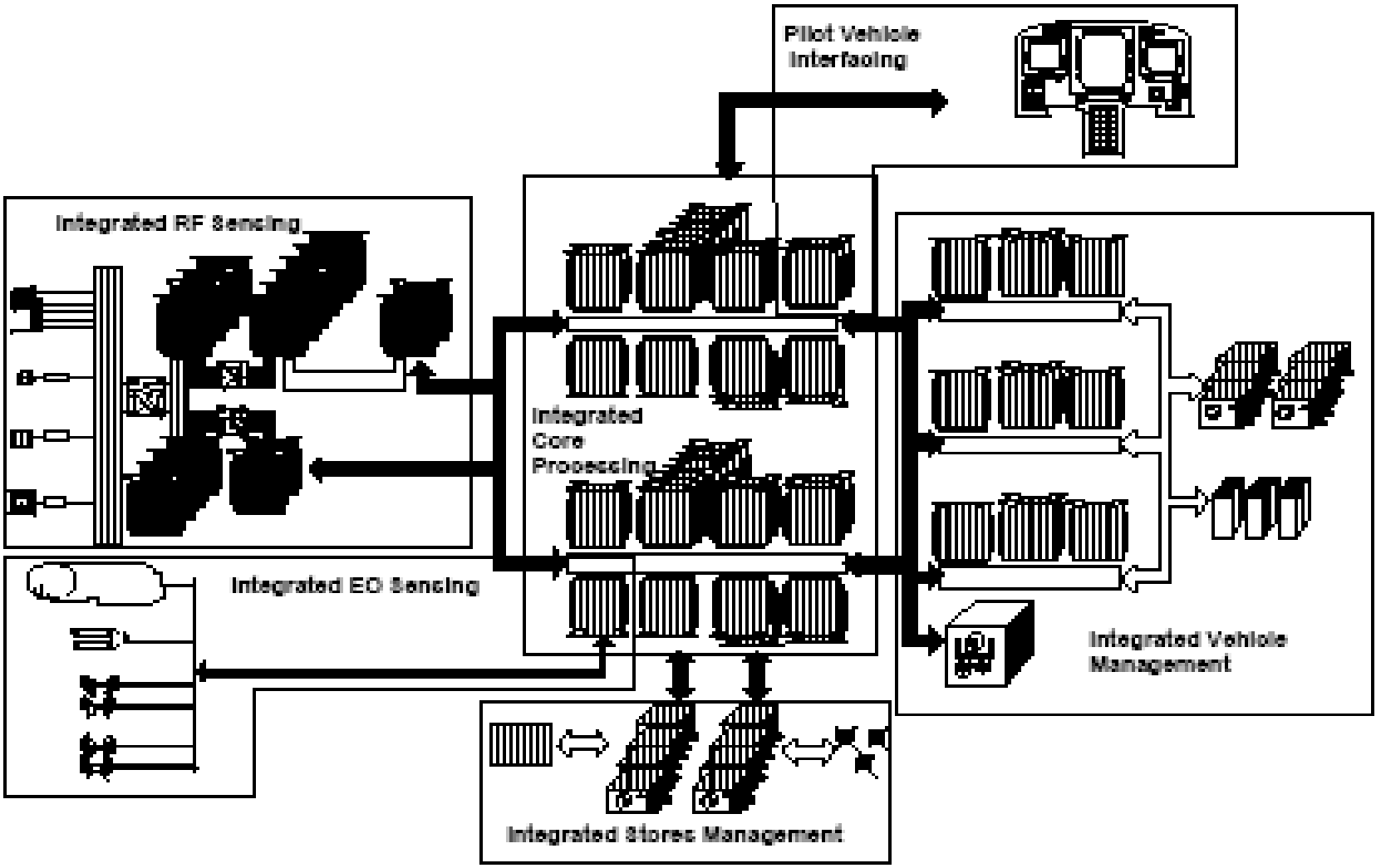
综合化系统结构



宝石台系统结构

- Pave Pace(宝石台)计划 (1990年代初)
  - 维持功能区的划分
  - 综合核心处理机技术
  - 综合无线电频率技术
  - 操作系统遵循POSIX 1003.1b 实时扩展

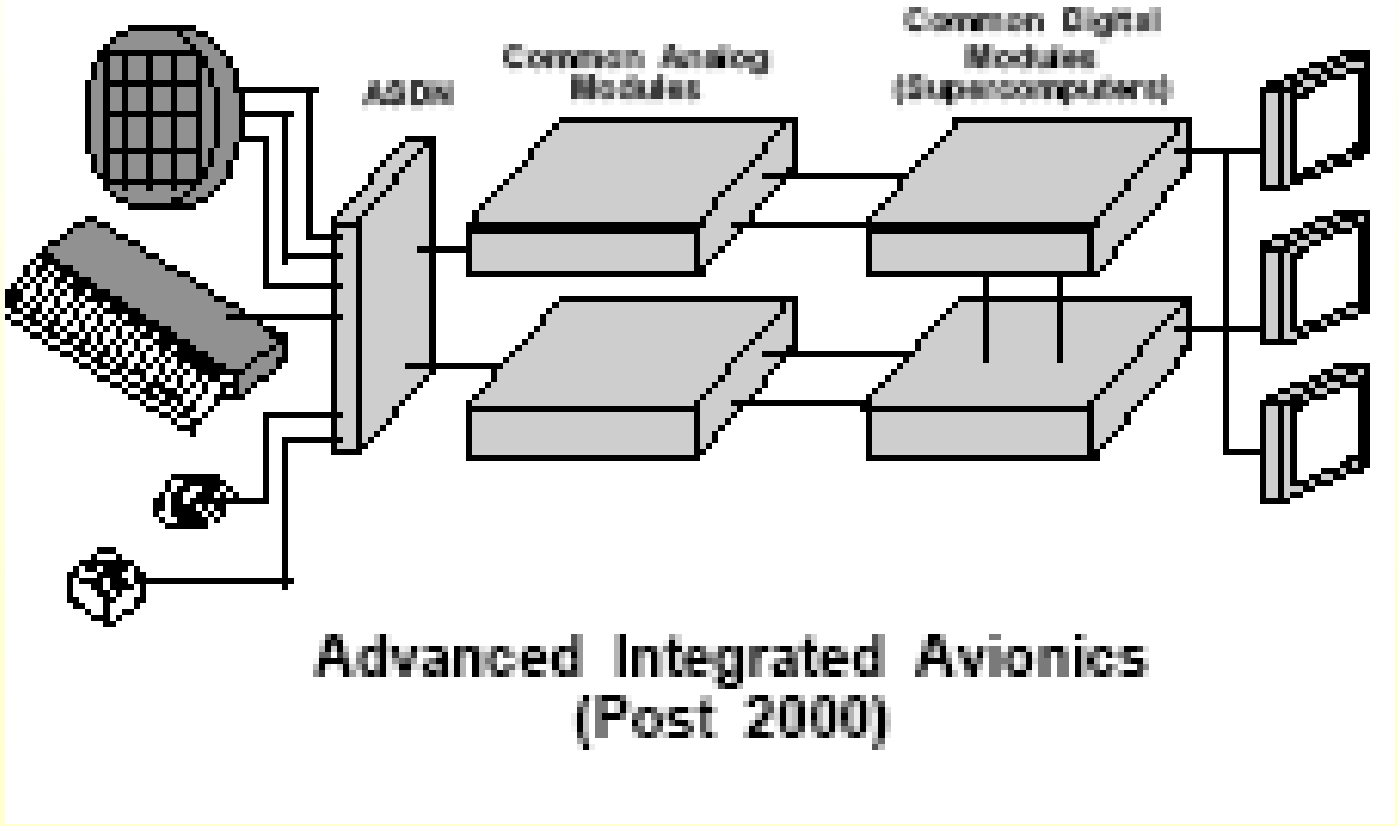
综合化系统结构


Pilot Vehicle Interfacing
Integrated RF Sensing
Integrated Core Processing
Integrated EO Sensing
Integrated Vehicle Management
Integrated Stores Management

开放式系统结构

- 采用COTS技术和成熟的工业标准
- SCI/RT或 Fibre Channel,统一互连网 络
- IEEE POSIX 1003.1b操作系统实时扩展
- RT-CORBA 中间件技术
- 基于构件模型的应用开发与验证
- 系统可开展性,便于系统功能增强
- 软硬件可重用性,便于软硬件升级
- 提高系统可靠性,采用容错技术
- 引入中间件导致四级系统结构

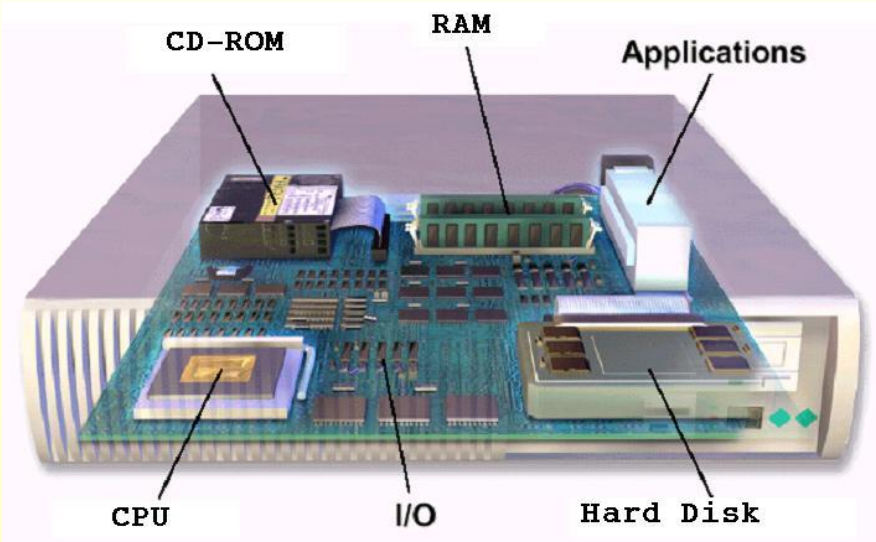
开放式系统结构


Common Digital
Modules
(Supercomputers)
Common Analog
Modules
ASDN
Advanced Integrated Avionics
(Post 2000)

什么是分布式系统

——意义与概念

集中式系统



几个场景

- 工厂生产系统
- 火车售票系统
- 工作站网络(NOW)
- 工作流系统
- 自动售货终端机
- 航空管制系统
- WWW（万维网）

概念和术语

- 分布式系统
- 分布式操作系统
- 分布式计算
- 并行系统
- 并行计算
- 分布式并行计算

文献：

1. George Cloulouris：Distributed Systems Concepts and Design（涵盖网络）
2. A.Goscinski:Distributed Operating Systems--the Logical Design（所有互连系统）
3. M.L.Liu:Distributed Computing：Principles and Applications
4. Andrew S.Tanenbaum:Distributed Operating System（涵盖并行）
5. Nancy A.Lynch: Distributed Algorithms (MIT)

分布式系统类型

- 分布式计算系统
  - 集群计算系统
  - 网格计算系统
  - 云计算
- 分布式信息系统
  - 事务处理系统
  - 企业集成系统
- 分布式普适系统
  - 家庭系统
  - 电子健保系统
  - 传感器网络

什么是分布式系统(1)

英国计算机科学委员会（1978）：

这样一种系统，其中包含多个独立但又有交互作用的计算机，对一个公共问题进行合作，其特性是包含多个控制路径，它们执行一个程序的不同部分而又相互作用.

什么是分布式系统(2)

P.H Enslow(1978):

① 包含多个通用资源部件（物理的或逻辑的），可以动态基础上被指定给予各个特定的任务；

② 这些资源部件在物理上是分布的，并经过一个通信网相互作用；

③ 有一个高级操作系统，对各个分布的资源进行统一和整体的控制；

④ 系统对用户是透明的；

⑤ 所有的资源都必须高度自治地工作而又相互配合，系统内不存在层次控制。

什么是分布式系统（3）

Andrew S.Tanenbaum:

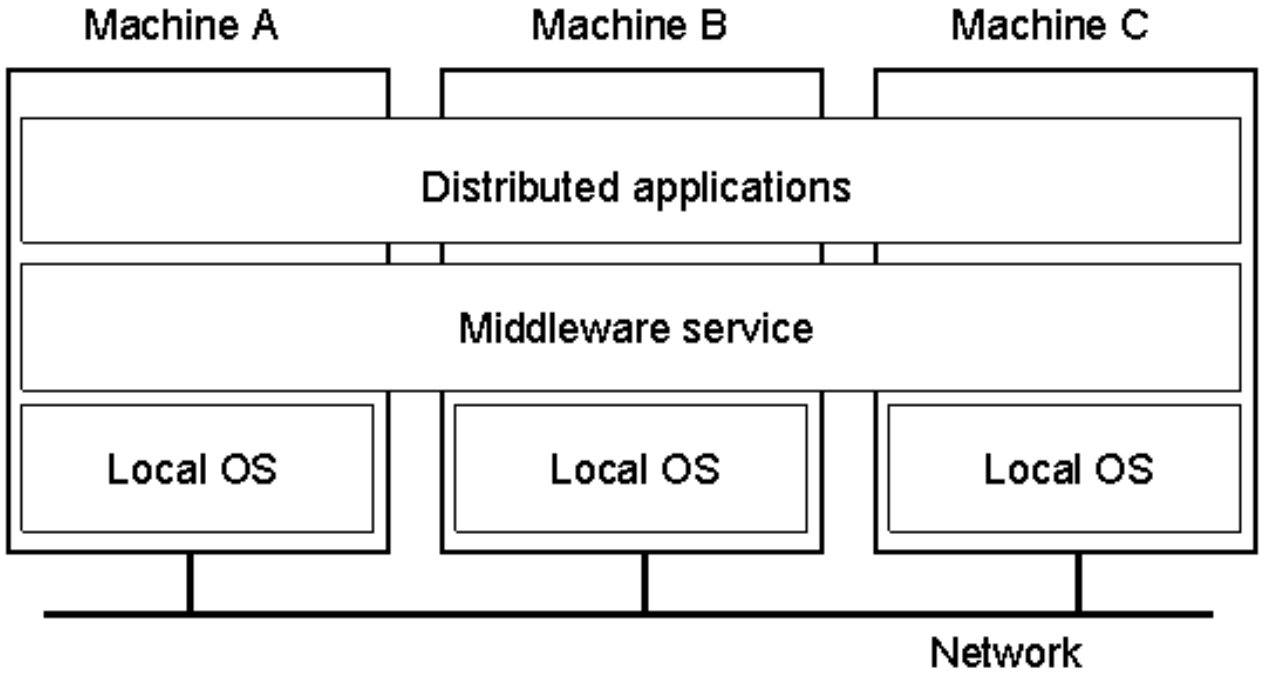
A distributed system is a collection of *independent* computers that appear to the users of the system as a *single* computer.

什么是分布式系统(4)

共同特性:

1. 软硬件结构上具有模块性
2. 工作方式上具有自治性
3. 系统功能上具有协同并行性
4. 对用户讲具有透明性
5. 运行时具有坚定性.

分布式系统组织



一个分布式系统组织成中间件形式，

中间件层分布在多台机器上。

分布式操作系统(1)

A distributed operating system is one that looks to its user like an ordinary centralized operation system, but runs on multiple, independent central processing units where

① **The use of multiple processors should be invisible to the user; or**

② **The user views the system as a virtual uniprocessor, not as a collection of distinct machines connected by a communication sub-system**

A distributed operating system should:

① Control network resource allocation to allow their use in the most effective way;

② Provide the user with a convenient virtual computer that serves as a high-level programming environment;

③ Hide the distribution of the resources;

④ Provide mechanisms for protecting system resources against accessing by unauthorized users;

⑤ Provide secure communication

计算模式

——的演变

三个计算时代：

- **手工计算时代：**

  实物交换

  公元前3000年中国人发明算筹，后演变为算盘

  1633年：奥芙特德发明计算尺

- **计算机计算时代：**

  1946年第一台电子计算机ENIAC

- **网络计算时代**

计算的目标

1930，普里斯伯格算术系统：
只包括自然数相加运算的数学系统是完备的
1974，100个符号的系统：1万亿台，1万亿次，1万亿年 (拉宾)

- Babage：差分机
  1822年英国剑桥大学数学家巴贝奇
- MEMEX 记忆延伸器：
  40年代 美国科学家Bush：《As we may think》
- 图灵测试法
  图灵1950：Computing Machinery and Intelligence

计算的革命刚刚开始！

计算的失败

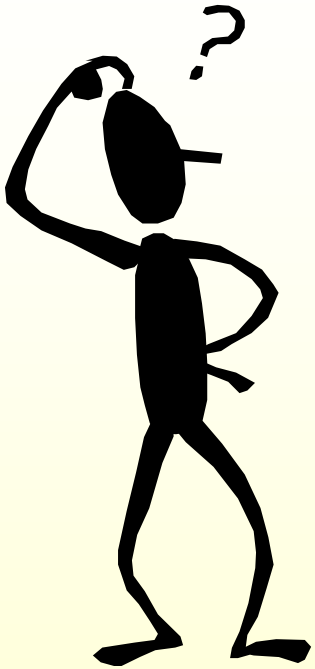
Butler Lampson:

- 计算机的性能
- 编程
- 软件工程
- RPC
- 分布式计算
- 连续对象存储技术
- 安全性

《二十一世纪的计算研究》1999

令人脸红的事情：物理学家发明了Web

计算的思维



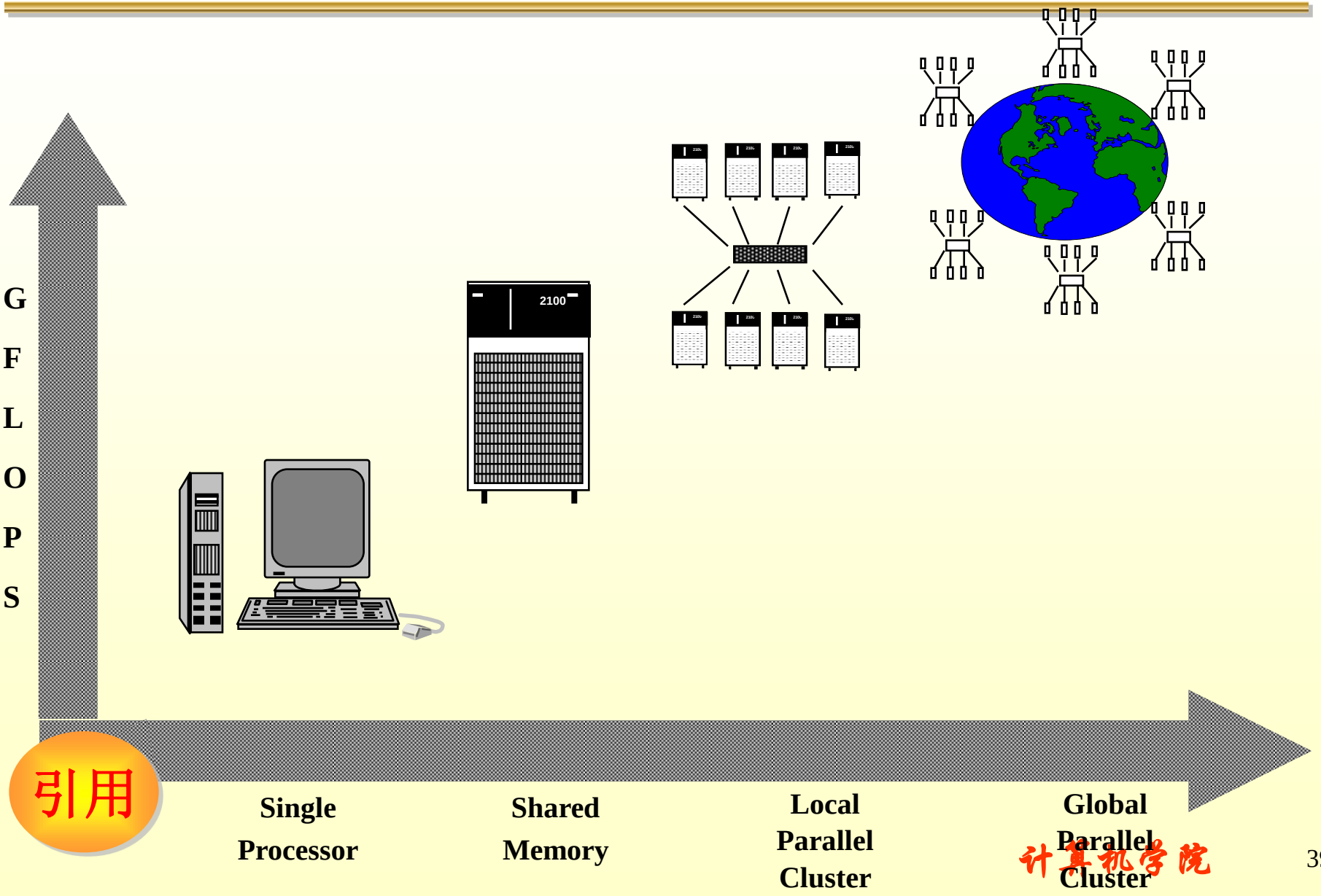
思维的三种方式：

- Work harder
- Work smarter
- Get help

发展的源动力：

- 需求
- 性价比

计算的发展


GFLOPS
2100
引用
Single Processor
Shared Memory
Local Parallel Cluster
Global Parallel Cluster

Butler Lampson:《二十一世纪的计算研究》 1999

推动因素（10～20年）：

- 模拟：比如人的纹理、身材扫描模拟等
- 通信：信息方便获取与远程参与
- 互动：与现实世界的交流，机器人等

计算未来的挑战：可用性和可扩展性！

分布式系统的源驱动力

- **计算速度**
  - Einstein的相对论

    任何物质传输速度不可能超过光速，电子运动传输速度约为光速的一半。
  - 元器件问题

    物理尺寸无限变小，使用硅材料的VLSI器件，其特征尺寸以0.1微米为极限，光刻蚀最小不小于0.13μm功，耗散热问题
  - 单机的极限速度：

    每秒10的9次 ～10的11次浮点运算
  - 冯.诺依曼溢口

    多个I/O设备的高速处理要求
- **孤岛问题**
- **可靠与容错性**
- **协同工作**

互联系统（1）

- **多计算机系统**

  多台分散的处理机组成的系统

- **多处理机系统**

  多个处理机构成的单机

- **网络系统**
- **分布式系统**

互联系统（2）

- **多计算机系统：60年代中期**

  通用计算机向大型集中式结构方向发展：I/O处理机，前端处理机

- **多处理机系统：70年代**

  集中式单机开始向网络互联和多处理机系统发展
  1）紧耦合的共享存储器
  2）分布式存储器
  3）混合型结构

- **网络系统：70年代**

互联系统（3）

分布式系统：1978年左右

采用多指令多数据流体系结构，特别是利用日益廉价的小型/微型机构成分布式系统

- California大学的1972年：DCS
- Maryland大学1979 ZMOB系统：256个Z80A组成一个环形结构
- Wisconsin大学 Crystal系统：50台VAX 11/750松耦合分布式系统

与分布式系统比较：

- 传统多计算机系统：不具备自治性，非并行求解
- 多处理机系统：缺乏自治性，强调统一操作系统下整体控制
- 网络系统：主要目标是资源共享，缺乏透明性。

系统设计目标分析

主要目标：

- **集成性**：来自大量的实际需求，一些传统的老系统，拥有不可替代的大量数据和软件，形成了一个个“孤岛”，需改造集成；各生产厂商生产出标准各异的应用系统需互联集成等。
- **资源共享**：分布在各地的资源是可交互的，大量信息的共享、获取数据库资料和各种服务是互联系统的重要需求。
- **协同工作**：企业内部、企业之间或团队之间需要既独立又协同的工作，如银行系统，航空、铁路售票和管理系统，旅馆业务等。
- **任务并行**：多机并行是解决高性能计算的最终途径
- **可靠与容错**：对许多有高可靠性要求的系统，多机互为备份是根本的解决方法

系统主要特性分析

主要特征包括：

- 自治性（Autonomy）：每个处理单元在系统中拥有的自主权，控制本地资源的权限等。
- 透明性（Transparency）：用户需对各类资源的了解和控制程度
- 异构性(Heterogeneity)：互联系统是由同构处理单元还是异构处理单元组成。
- 并行(Parallel)：可并行性，包括并行的机制，并行的规模，并行管理，并行任务的粒度等。
- 互协性(Interoperability)：各互联单元之间协同工作的能力
- 可扩展性(Expansibility)：系统是否能方便的扩展规模和功能
- 安全性(Security)：如何保证交互信息的可靠、正确和不受干扰

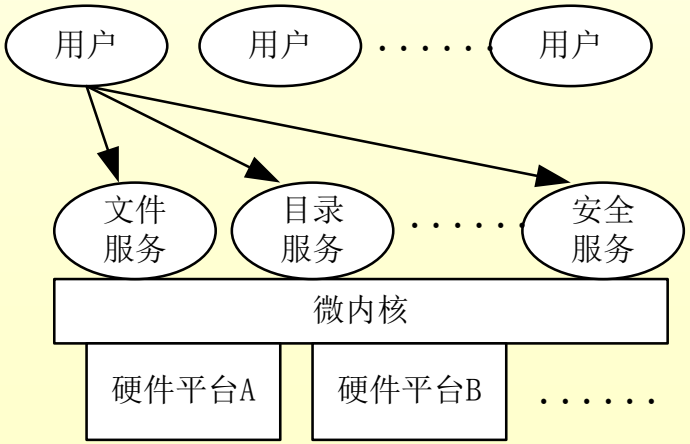
网络和网络操作系统（NOS）—最基本的互联系统

- 主要目标是信息共享和提供非透明服务，用户具有完全的自治性。对全局管理、并行操作、自治控制等特性无硬性要求，结点之间物理联系、逻辑关系松散，强调个性，没有一个完全一致的系统状态图。

分布式操作系统（DOS）模式 — 互联系统的高级形式

- 所有的系统资源都是对用户开放的，用户就像在使用一个完整的单机（基于微内核的和现有OS之上的二种构建方法）
- 每个机器安装统一的微内核以连接各不同的硬件系统，微核结构减小了操作系统尺寸，隐藏了硬件的异构性
- 分布式系统中处理机只有很少的自治性，主要由DOS和系统管理员统一管理

（Amoeba，Mach，Chorus，Clouds，Sprite，DCE）



NOS vs DOS

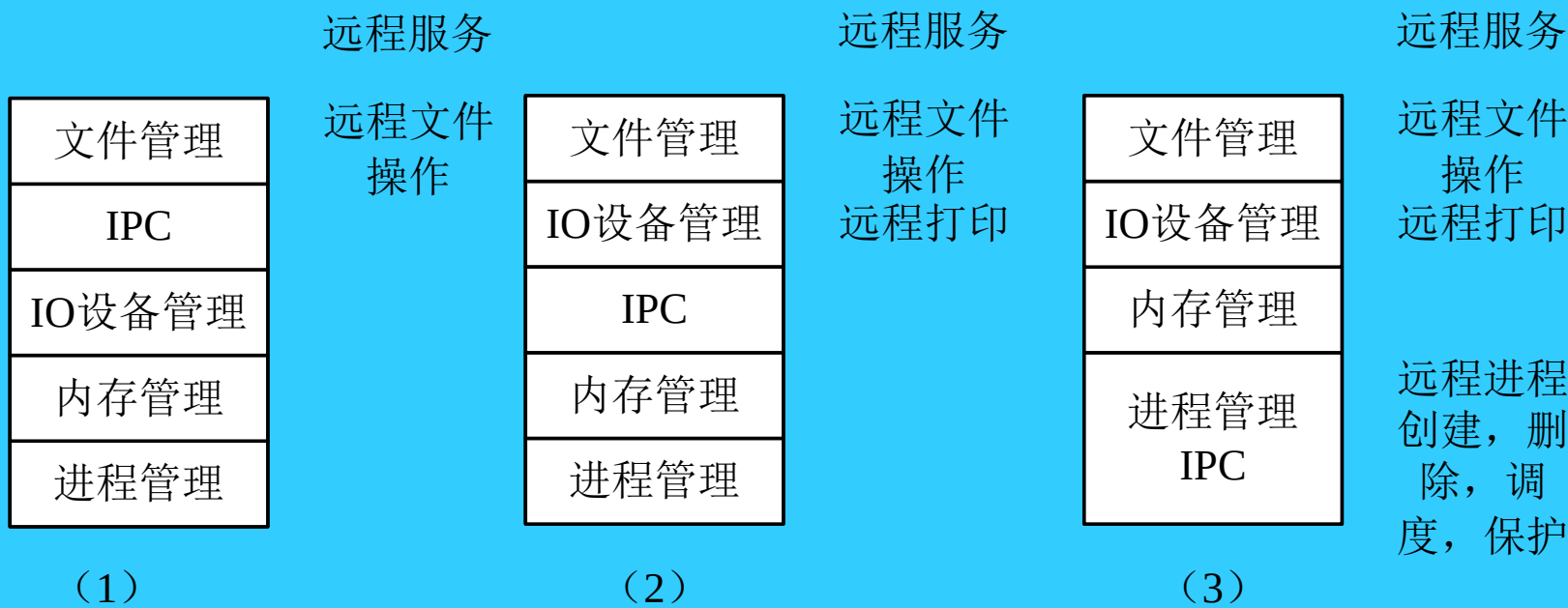
与分布式系统的最根本区别：

透明性：对用户屏蔽系统组件的分散性

[ISO 1992]:

- **访问透明性: 相同操作访问本地和远程资源**
- **位置透明性：不需要知道资源位置就可访问**
- **并发透明性：几个进程能并发共享资源，互不干扰**
- **复制透明性：无需知道副本的存在**
- **故障透明性：屏蔽错误**
- **移动透明性：不影响应用的情况下，允许资源和客户移动**
- **性能透明性：负载变化时，允许系统重构**
- **伸缩透明性：不改变系统结构前提下，允许系统扩展**

IPC位置



IPC在操作系统中位置

文件系统操作

1. 没有操作系统方法：文件传送（如USENET网络）
2. 网络操作系统方法：把不同文件系统连接起来

   Open (“/machine1/pathname”,READ)
3. 分布式操作系统方法：所有各子系统的文件系统组成一个整体文件系统，任何机器想要读password文件时

   open(“/etc/password”,READ ONLY)

不必指出该文件在哪里（LOCUS）

NOS vs DOS－访问控制

用户标识（UID）问题：多机远程访问不唯一性

1. 没有操作系统时：使用远程机器上用户名。
2. 网络操作系统时：对不同机器的UID进行变换，需要提供一个充分大的变换表
3. 分布式系统中：单一的UID

NOS vs DOS－执行地点

用户运行一个程序时，在哪里运行？

1. 没有操作系统时：远程登录，然后运行在那里的作业
2. 网络操作系统时：指定一个机器运行

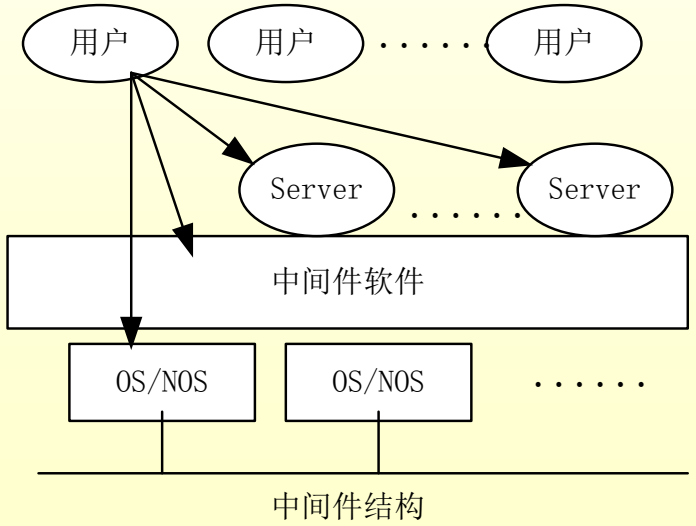
   remote vax4 job
3. 分布式操作系统: 用户只需要简单提交job,由操作系统寻找合适的处理机，用户察觉不到机器的边界。

中间件模式(middleware) —— 折中的的计算模式

分布式系统DOS具有上述的优势，却并不实用，其主要原因：

- 技术难度大；
- 继承性差，一般需对底层重新开发；
- 现实中存在大量的独立设备，具有自治性，很难把它们联入分布式系统

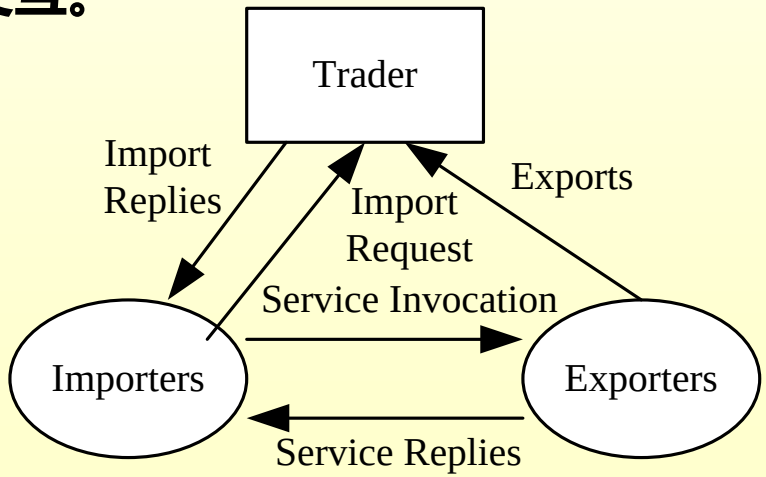
增加系统透明性，
抑制网络用户自治性
屏蔽底层异构性



ISO/CCITT RM-ODP模式

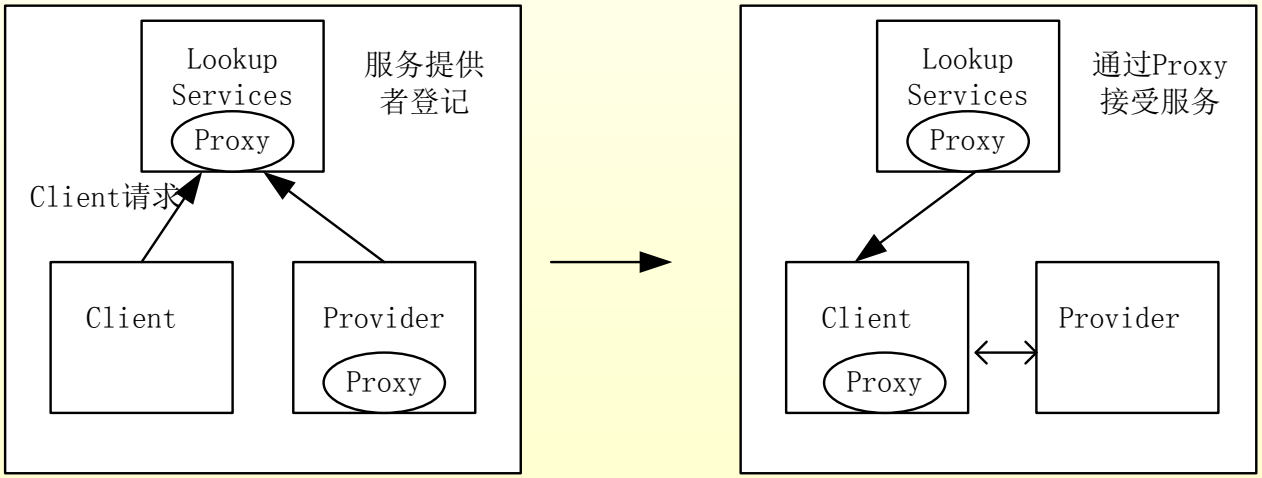
动态黄页服务（Dynamic Yellow Page Service）

- 服务提供者（Exporters）向交易员提供服务属性；
- 请求者（Importers）：通过有关服务属性透明地动态获取所需的服务；
- 交易员保有所有类型的服务属性和服务提供者数据库；
- 在收到服务请求时，交易员找到最合适的服务入口和服务者，并把它返回给请求者，此时可通过调用服务所提供的操作直接和所选定的服务器进行交互。



SUN公司1999年元月推出的Jini

它提出一种服务“即插即用”的思想：任何服务者首先在一个中心管理者（Lookup Services）中登记，其它Client即可通过Lookup Services下载中间代理（Proxy）透明地请求到该项服务，若此服务撤消，则取消该项服务



Jini服务过程

分布式系统（OS）

——概述

分布式系统特征

- 一组由网络互联的、自治的计算机和资源
- 资源为用户所共享
- 可以集中控制，也可以分布控制
- 计算机可以同构，也可以异构
- 分散的地理位置
- 分布式故障点
- 没有全局时钟
- 大多数情况下没有共享内存

分布式系统目标

- 能够使用户方便地访问资源
- 必须隐藏资源在一个网络上分布
- 开放的
- 可扩展的

资源可访问

- 资源：计算机，存储，打印机，文件，数据等。
- 资源可访问
- 协作与交换
- 信息安全

透明性（ Transparency ）

透明性	描述
访问	隐蔽数据表达方法和资源访问方法的不同之处
位置	隐蔽资源所处的物理位置
迁移	隐蔽资源的物理移动
重定位	隐蔽正在使用的资源迁移
复制	隐蔽资源的复制
并发	隐蔽若干用户共享同一资源所产生的竞争
故障	隐蔽资源的故障与排错恢复
持续	隐蔽软件资源所处的存储空间：内存或磁盘

开放性

- 通信标准与协议
- 服务通常通过接口（IDL语言描述）
- 互操作性：何种程度协同工作
- 可移植性
- 灵活组合的（策略与机制分开，Web缓存）

可扩展性

- 规模上可扩展
- 地域上可扩展
- 管理上可扩展

扩展与性能的矛盾

规模扩展问题

集中式的受限问题

Concept	Example
Centralized services	A single server for all users
Centralized data	A single on-line telephone book
Centralized algorithms	Doing routing based on complete information

分布式方式：

- 没有任何机器拥有系统完整信息
- 只根据本地信息决策
- 不会出现单点故障崩溃
- 没有全局性时钟

地域扩展问题

- 广域网同步与延迟
- 不可靠通信：点对点
- 集中组件导致资源浪费

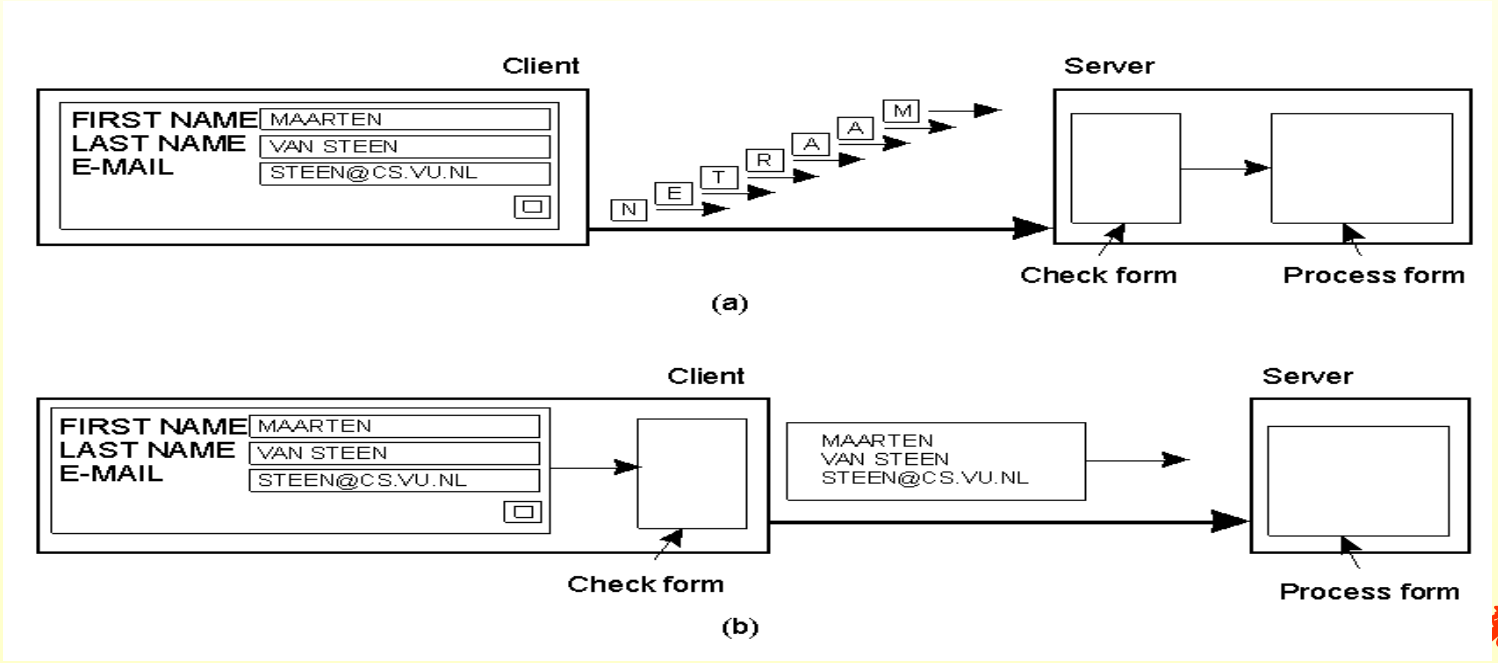
管理扩展问题

- 跨管理域
- 资源使用
- 权限与安全
- 配置与管理

扩展技术（1）

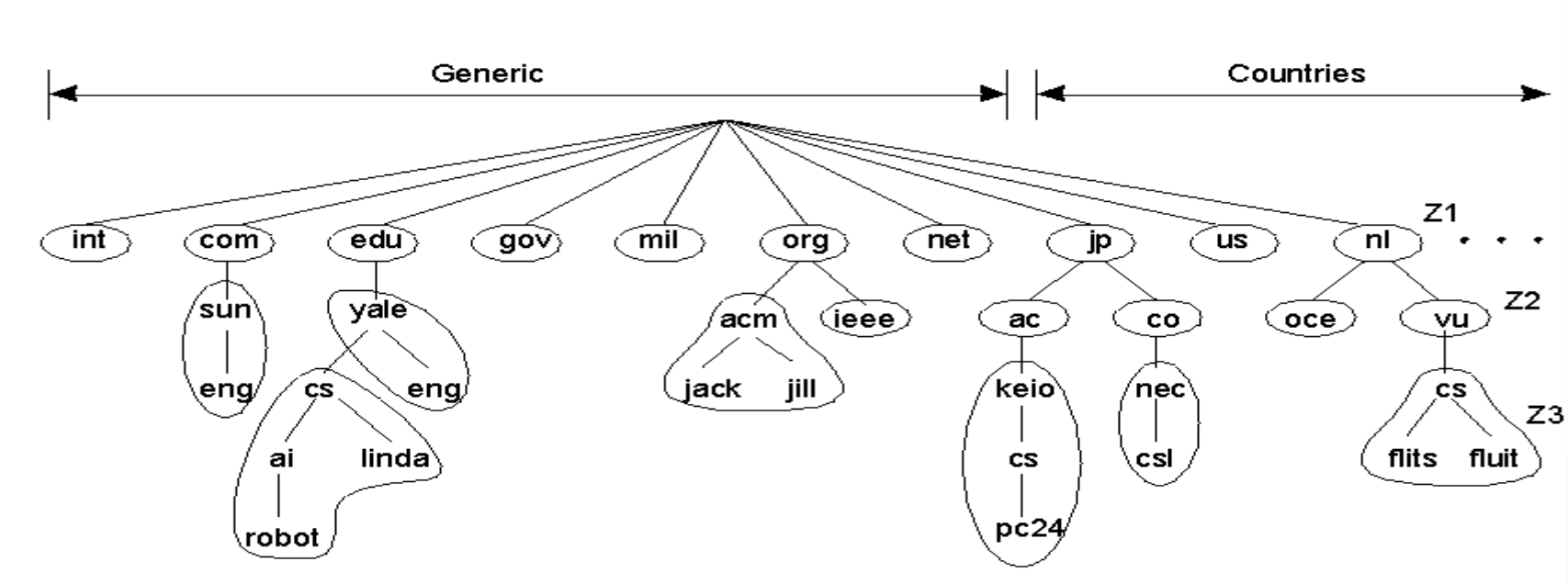
- 隐藏通行等待时间（解决：地域扩展）
  - 异步通信方式
  - 启动新控制线程执行请求
  - 将部分工作分散给客户（交互程序中）

表单访问数据库的例：



扩展技术（2）

- 分布（Distribution）技术
  - 将某组件分割为多个部分，分散到系统中



An example of dividing the DNS name space into zones

扩展技术（3）

- 复制技术（解决性能下降问题）
  - 能增加可用性
  - 有助于负载平衡
  - 新问题：一致性问题
  - 解决：全局同步机制

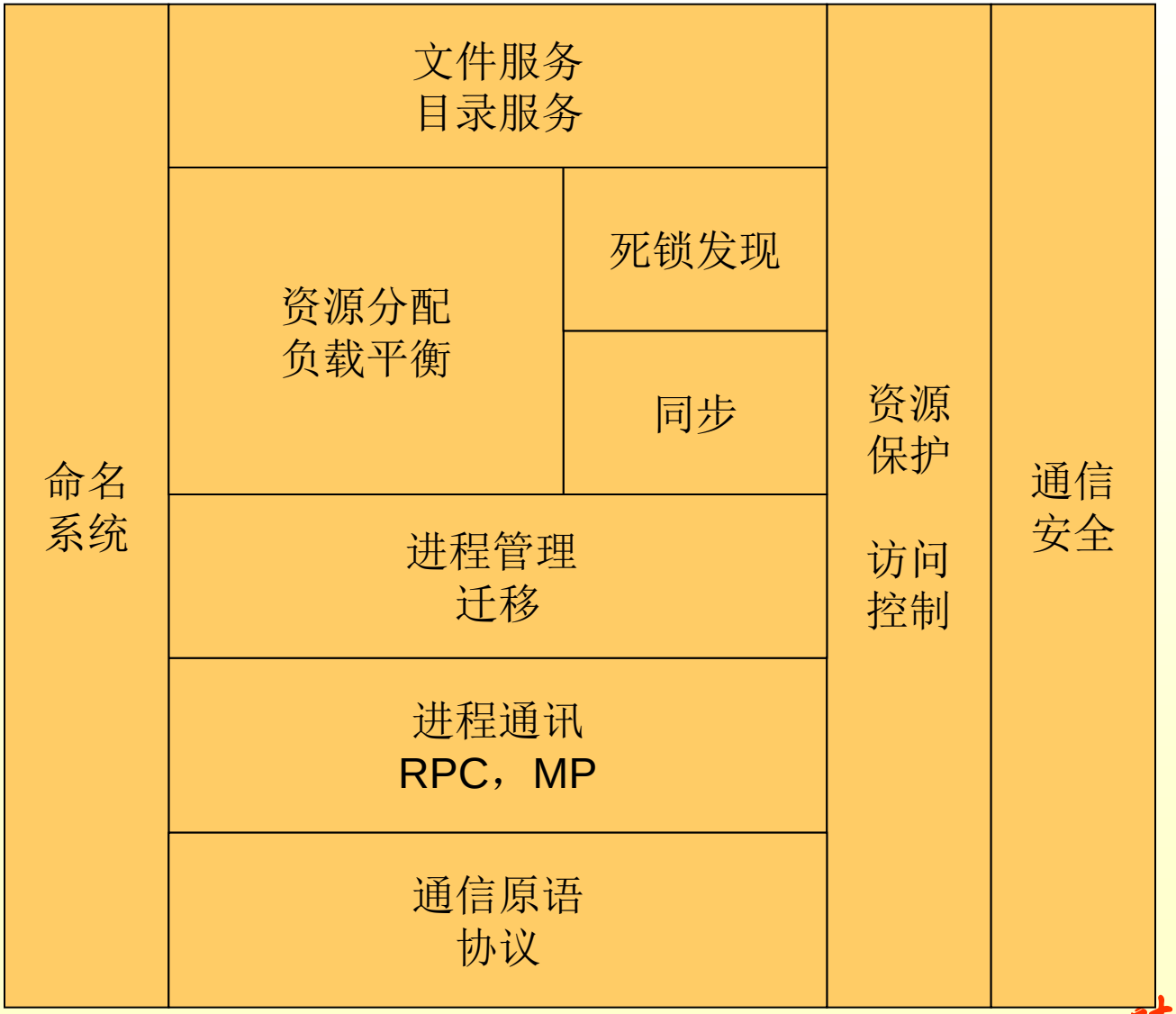
OS管理问题

- OS管理问题：进程，内存，资源和文件管理等，互连带来的方法、策略和算法问题
- 进程通信：message passing, C/S，RPC模型，依赖于通信原语和协议
- 通信进程必须同步，算法和策略，
- 命名服务，各种组件和实体需要命名策略
- 提高效率：资源管理，负载平衡，调度策略，进程迁移
- 死锁发现，保护机制，访问控制
- 分布式文件服务
- 安全策略

1）分布式进程：进程通信，同步，命名，管理

2）分布式资源：资源分配，死锁发现，安全和一些服务。

分布式OS内容


文件服务
目录服务
死锁发现
资源分配
负载平衡
同步
命名
系统
进程管理
迁移
资源
保护
访问
控制
通信
安全
进程通讯
RPC，MP
通信原语
协议

分布式计算

——的发展

计算的追求

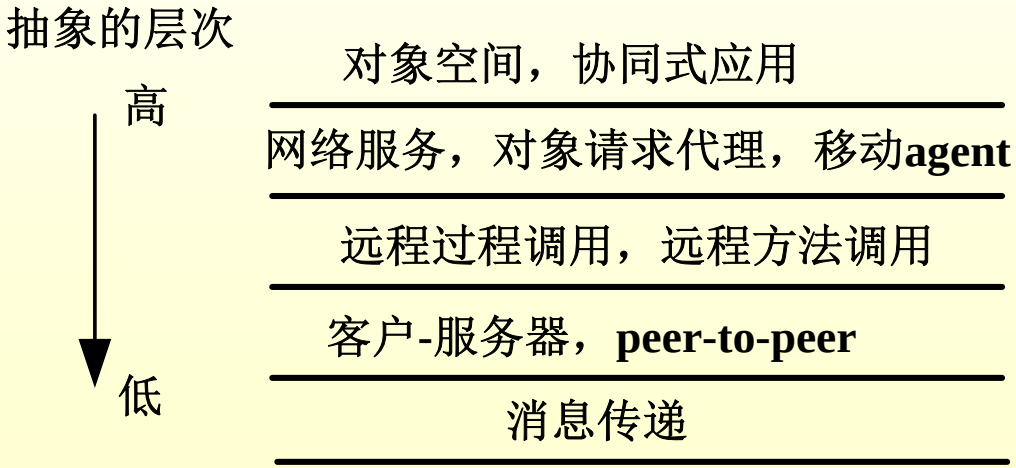
更高、更快、更强！

更廉、更方便！

提供无处不在的方便的、快速的、可靠的计算！

分布式计算技术范型（1）

- **范型（paradigm):一种模式、例子和模型（韦氏大词典）**
- **分层：按照抽象层次顺序分类描述**
  - **底层：消息传递，封装的细节最小**
  - **高层：对象空间**

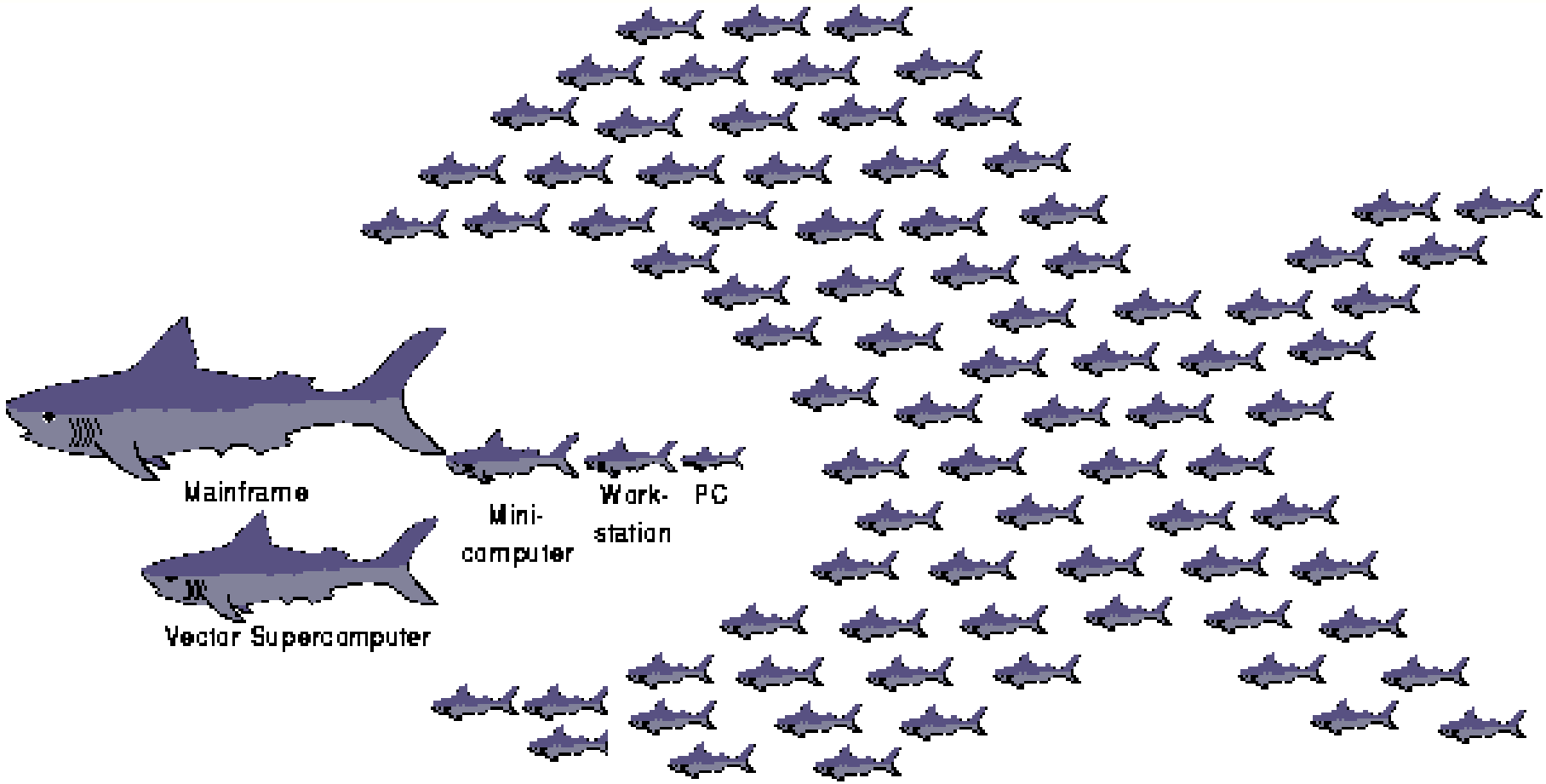


分布式计算范型抽象层次

分布式计算技术范型（2）

- 消息传递：基本操作send/receive
- 客户-服务器：HTTP，FTP等
- Peer-to-Peer：允许任何设备成为一种服务器，如Napster文件传输服务
- 远程过程调用RPC：SUN API，DCE API
  SOAP：实现支持Web的远程过程调用
- 分布式对象模式
  - 远程方法调用RMI：面向对象的RPC
  - 对象请求代理模式：对象请求代理充当中间件角色 ，CORBA
  - 基于构件的技术：COM，DCOM，Java Bean，EJB等
- 移动agent:携带执行路线，携带并传递数据，如Concordia，Aglet等
- 网络服务:服务对象注册，请求者查询和访问服务
- 协同应用（群件）：参与协同的进程形成一个组.Lotus QuickPlace
- 对象空间：某一应用的所以参与者都集中到一个公共对象空间，虚拟空间或会议室对象。如JavaSpaces.

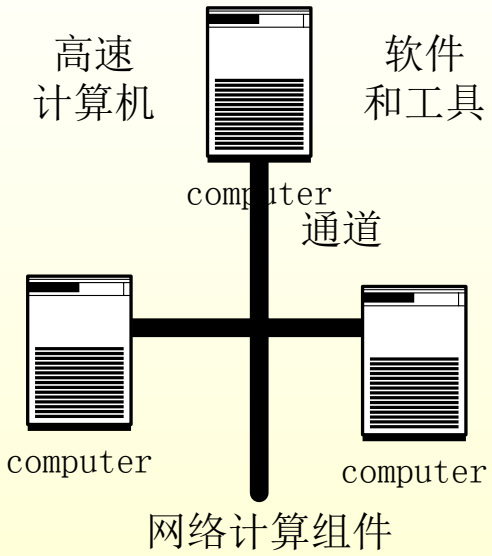
Now and Future



计算的组件

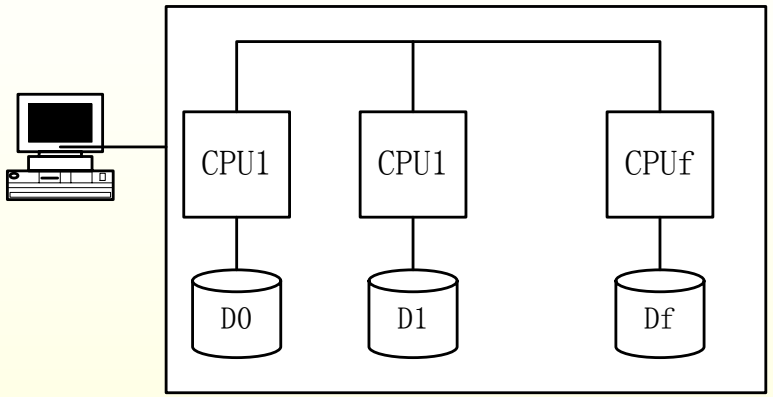
ILLIAC IV系统的论证者曾概括性的论述了发展新系统主要应考虑的因素

- **更高的性能**
- **可靠性和可用性**
- **性能价格比**
- **能尽可能保护投资**
- **可扩展性**



1996年夏，美国人建成了第一个Beowulf机群，它由16个DX4 100M处理机组成。

1997年又推出16个基于PII结点的机群，只用5万美元却具有每秒10亿次的浮点运算能力，而相同能力的并行机需投资数十倍



Beowulf结构

- Beowulf通信是通过目标以太网的TCP/IP协议，改造Linux核心技术，用一种“通道捆绑”（Channel banding）技术获得高速通信。
- 为了为用户提供单一系统映象，在一分布式框架中给进程分配全局唯一、跨越若干核心的PID是经常的做法。

"The word's fastest computer is now probably the Internet -- or whatever subset of its connected can be harnessed to the same computing job".
(Scientific American, May 1997, Page 115.).

- Network of Workstation (NOW),
- Metacomputing,
- Cluster of Workstation (COW),
- Global Computing,
- Grid Computing,

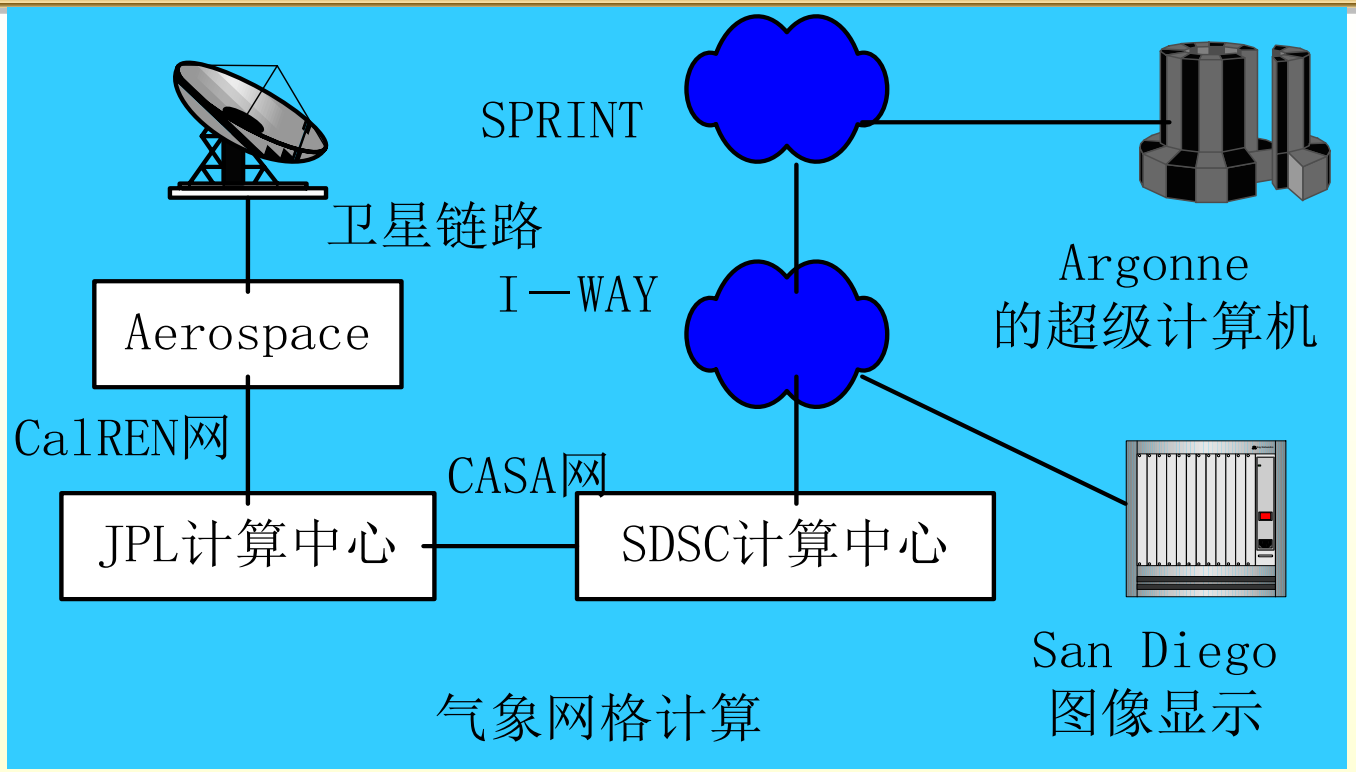
集群计算（cluster computing）

网格计算（Grid Computing）

引用

Source: www.globus.org & updated

第一个网格



- 输入对象运行在位于**Los Angeles**的**Aerospace**公司的**Sparc-10**机上;
- 管理对象和工作对象运行在位于**Chicago**的**Argonne**实验室**IBM SP-2**的20个结点上;
- **CAVE**显示对象运行在**San Diego**的**SGI**工作站上。

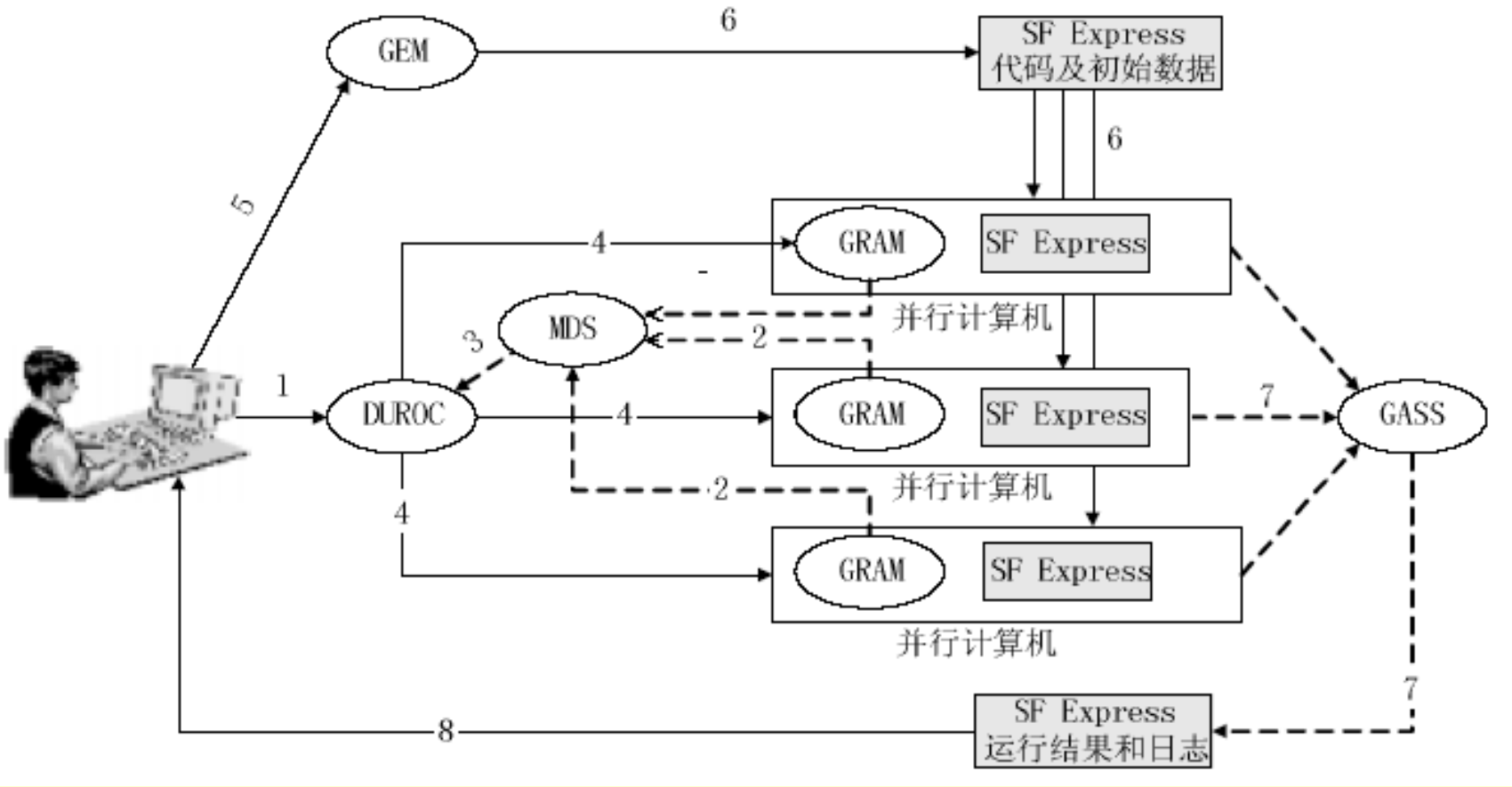
通过**ATM**虚电路交换信息**San Diego**的用户可以利用**Los Angeles**的设备取得卫星数据，通过在**Chicago**的并行计算，显示出近乎实时的气象云图。

- 1996年美国国防部资助的军事模拟仿真环境
- 1996.11单机（Intel Paragon)上模拟10000个战斗单位
- 1997年扩展到7个时区6台超级计算机，模拟了50000个战斗单位
- 1998年与Globus融合，集合13台并行机，模拟了100298个战斗实体
- 能适应网格动态变化，自动资源选择，自动调整运行状态，容错等

SF Express问题

- 如何管理同时在13台超级计算机上开始SF Express？
- 如何传递初始数据和程序并启动？
- 如果某计算机出现异常，怎么处理？
- 出现网络问题怎么办？
- 如果某台机器要退出怎么办？
- 如何获得结果和运行数据

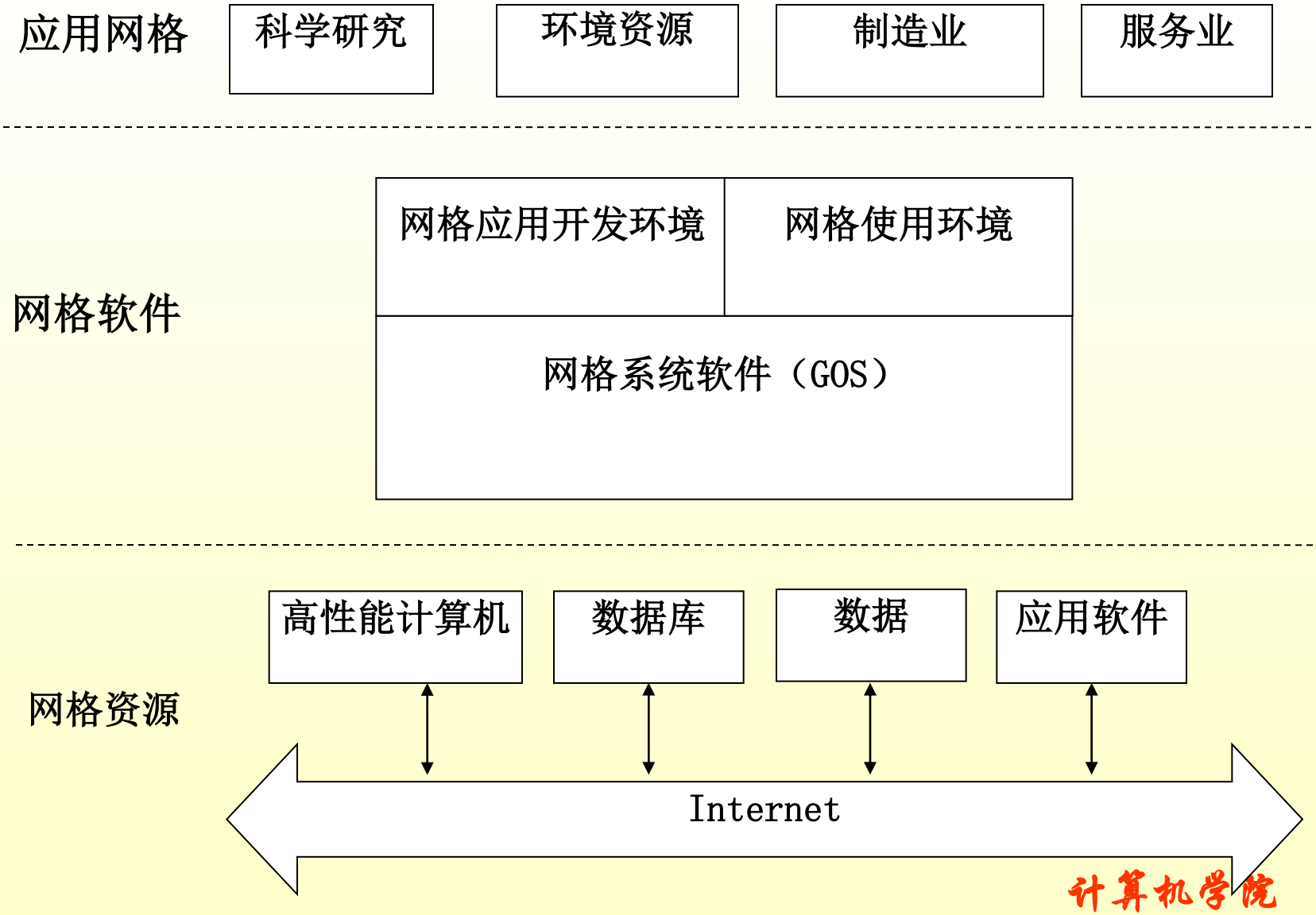
SF Express的结合



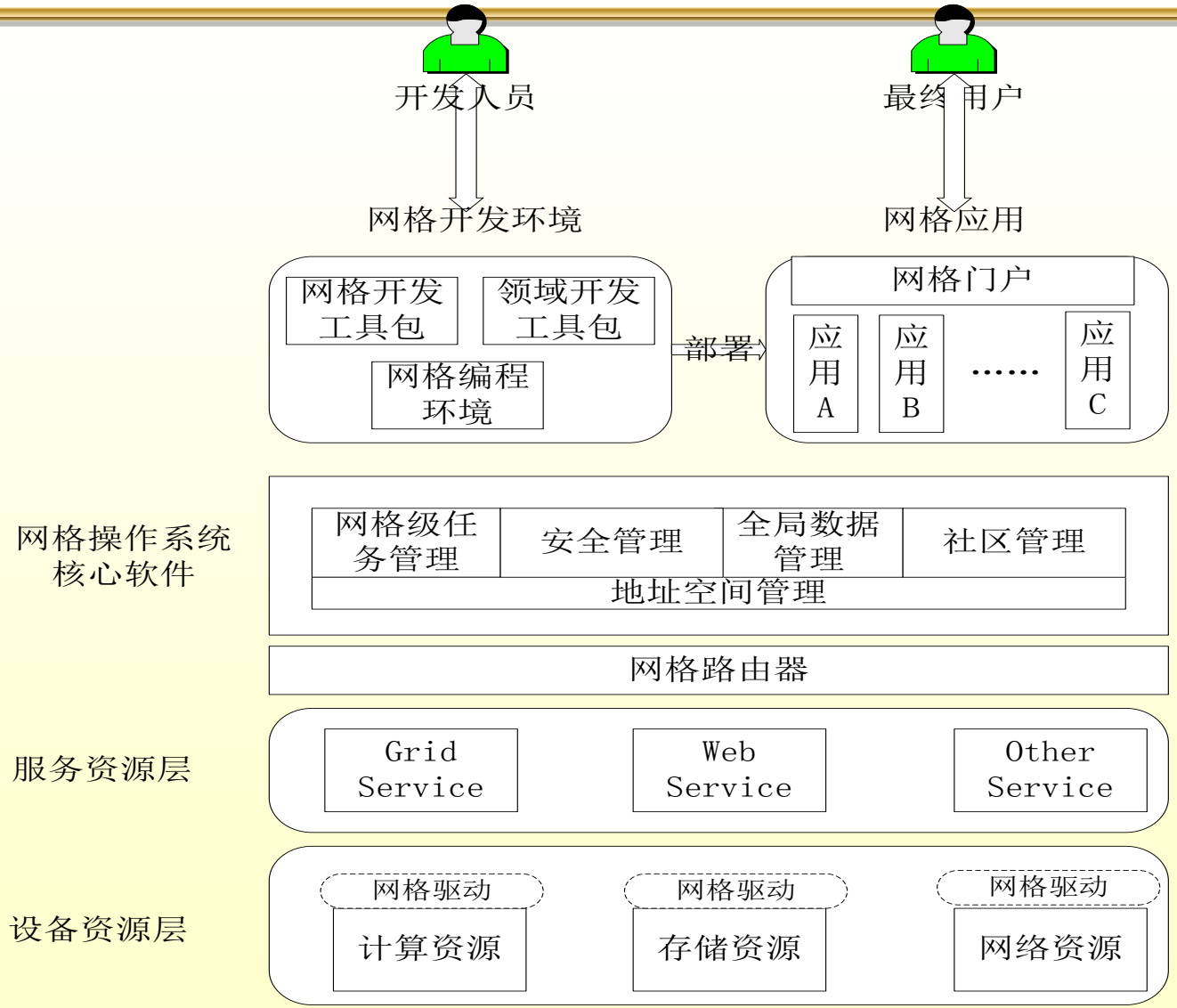
国家网格综述

- 九五期间863计划重点项目“国家高性能计算环境”
- 十五863重大专项“高性能计算机及其核心软件”重大专项
  - 项目周期：2002年-2005年
  - 863计划经费1亿人民币
  - 吸引地方政府、应用部门和产业界2-3倍的配套经费

国家网格（CNGRID）


应用网格
科学研究
环境资源
制造业
服务业
网格软件
网格应用开发环境
网格使用环境
网格系统软件（GOS）
网格资源
高性能计算机
数据库
数据
应用软件
Internet

Vega GOS 1.0体系结构



引用

应用网格

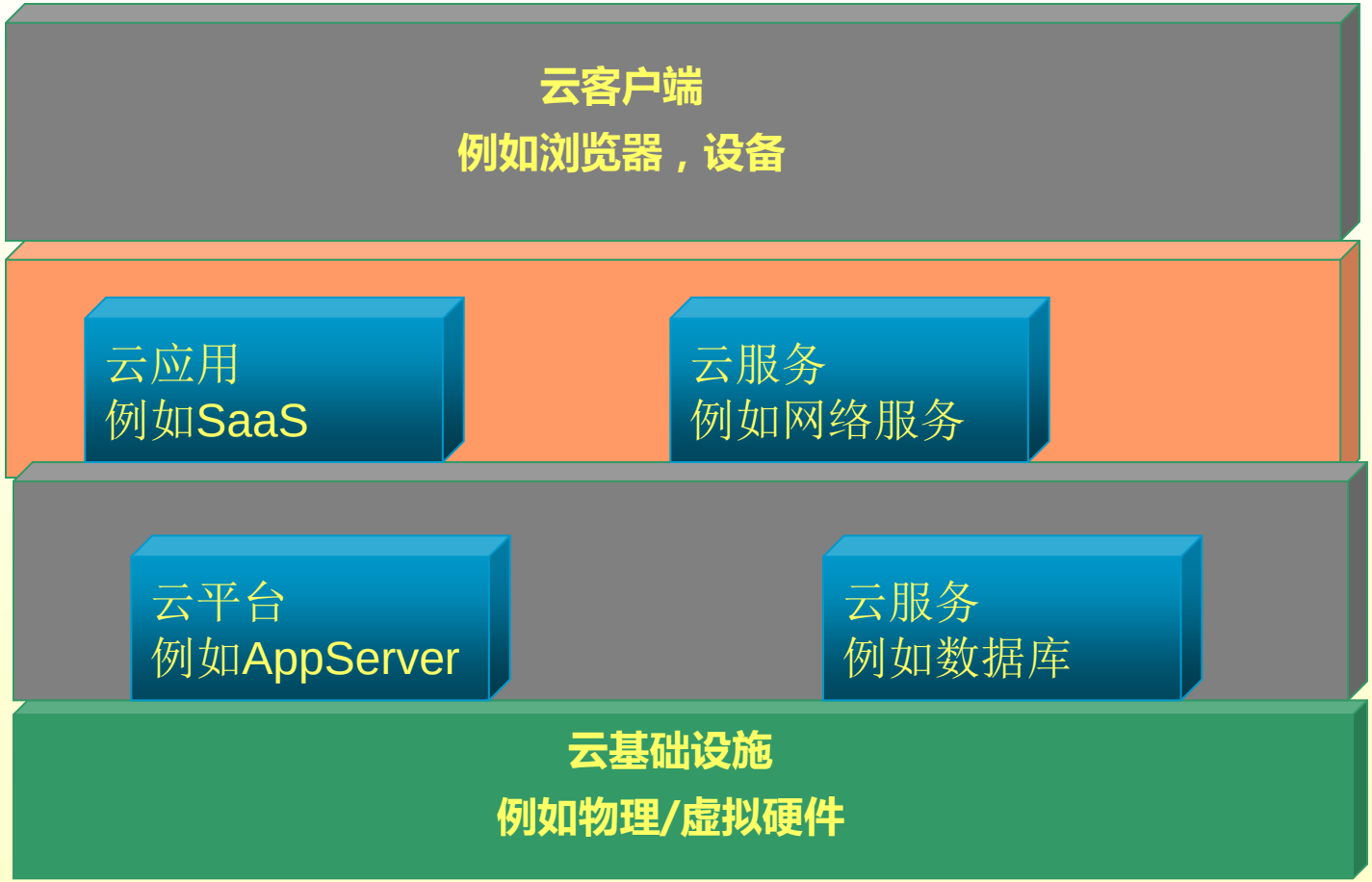
- 气象网格
- 资源环境网格
- 航空制造网格
- 科学数据网格
- 新药研发网格
- 生物信息网格
- 仿真应用网格
- 教育网格
- 城市交通信息服务网格
- 国家地质调查网格
- 森里资源与生态信息网格应用

Cloud Computing（1）

云计算（2007底）：网格计算出现之后的又一种新的计算机模式。

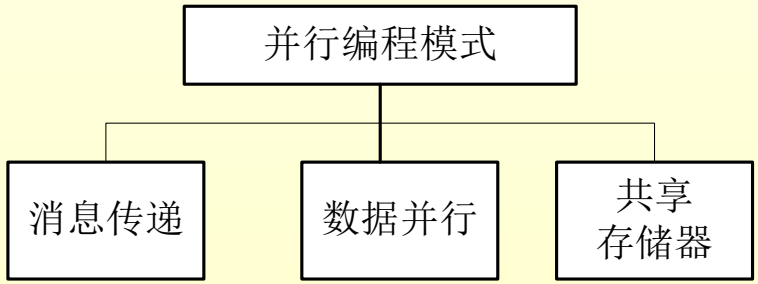
- **它将计算任务分布在大量计算机构成的资源池上，使各种应用系统能够根据需要获取计算力、存储空间和各种软件服务。Google云计算已经拥有100多万台服务器， Amazon、IBM、微软、Yahoo等的“云”均拥有几十万台服务器**
- **云计算是并行计算(Parallel Computing)、分布式计算(Distributed Computing)和网格计算(Grid Computing)的发展，或者说是这些计算机科学概念的商业实现。**
- **云计算是虚拟化(Virtualization)、效用计算(Utility Computing)、IaaS(基础设施即服务)、PaaS(平台即服务)、SaaS(软件即服务)等概念混合演进。**

- **云计算组成：**


云客户端
例如浏览器，设备
云应用
例如SaaS
云服务
例如网络服务
云平台
例如AppServer
云服务
例如数据库
云基础设施
例如物理/虚拟硬件

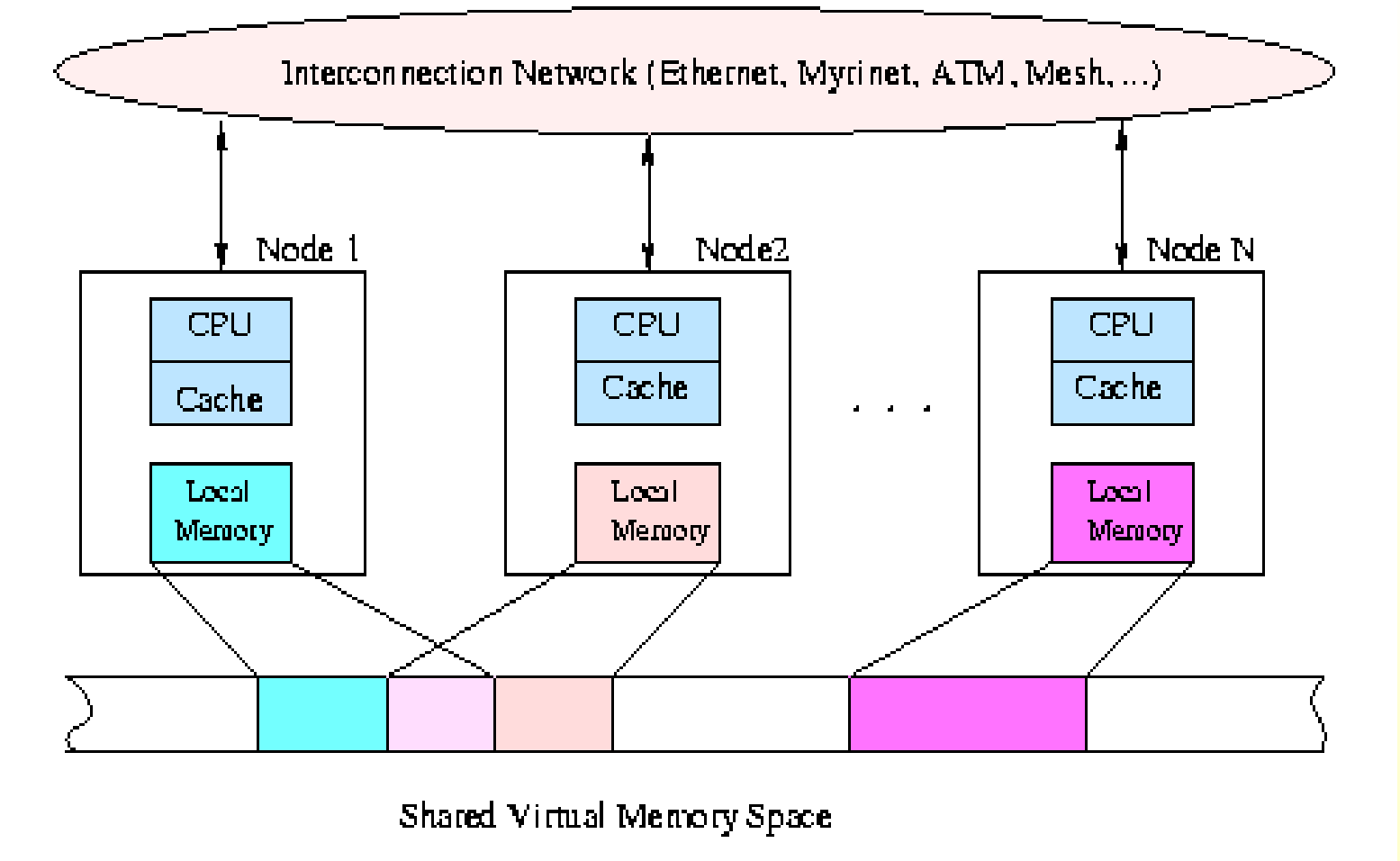
并行编程模式

- 消息传递方式：适合于分布式存储器模式，并行编程语言通常是扩充的C、C++、FORTRAN等语言，如PVM、MPI、P4等仅仅是在常规语言上增加了一些子函数来完成消息传递等并行功能。
- 数据并行方式：适合SIMD型并行机，每个结点机运行同一个程序处理不同的数据，并行编程语言是扩展C*、高性能Fortran等。
- 共享存储器方式：这种方式适合于共享存储器型并行机，允许全局共享变量，并行编程结构简单明了，使用常规或扩展的并行语言，如Linda等



并行编程模式

分布式共享存储系统



在分布存储系统上提供共享存储抽象

分布式共享存储（DSM）系统

- 结合共享存储的易编程和分布式存储的易扩展
- 在分布式系统的基础上通过软件或软硬结合的方法提供共享存储的编程环境
- 硬件DSM系统：DASH, FLASH, Alewife, DDM
  - 商品化系统：KSR-1，Origin 2000，NUMA-Q
- 软件DSM系统：IVY, Midway, Munin, Quarks, TreadMarks, CVM, JIAJIA, …
  - 又称为虚拟共享存储系统、共享虚拟存储系统
- 硬软件结合的DSM系统：Shrimp, Typhoon, Simple-COMA, …

DSM例子

```
#include <ioastrem.h>
shared{
int array[100];
}
void main(int,char *[]{
int num, I
spawn(host1);
spawn(host2);
spawn(host3);
for (int I =0;I<100;I++)
  shared->array[I]=-1;
cout<<"how many jobs?";
cin>>num;
parbegin
   routine[num](int total,int myid){
        shared->array[myid]=myid;
  }
parend
for (I=0;I<num;I++)
  cout<< "Hello word from job number"<<shared->array[I]<<endl
```

输出结果: Hello word from job number 0
Hello word from job number 1
……

消息传递例子

hello.c:

```
#include <stdio.h>
#include <pvm3.h>
main()
{  int cc,tid;
   char buf[100];

  printf("I'm t%x\n",pvm_mytid());
  cc=pvm_spawn("hello_other",(char **), " ",1,&tid)
       pvm_initsend(Pvm_DataDefault);

 /*在另一处理机上派生程序*/
 if (cc=1){

cc=pvm_recv(-1,-1);
/*接收返回消息*/
  pvm_bufinfo(cc,(int*)0,(int*)0,&tid);
  pvm_upkdtr(buf);                    }
  printf("from t%x:%s\n",tid,buf);
}else
  printf("can't start hello_other\n");
  pvm_exit();
  exit(0);
}
```

程序输出：

I'm t40002
from t40003: hello word ……

hello_other.c:

```
#include "pvm3.h"

 main()
  { int ptid,char buf[100];
     ptid=pvm_parent();
   strcpy(buf,"hell word ");
       gethostname(buf+strlen(buf),64)


   pvm_pkstr(buf);

    pvm_send(ptid,1);
  /* 运行结果传递回去*/
    pvm_exit();

      exit(0)
```

小结

- 课程目标
- 计算模式的演变和发展思维
- 互连系统的本质特征
- 三类系统（初、中、高级）的联系与区别
- 中间件模式（两种思想）
- OS与分布式OS的主要特征
- 各种分布式计算模式的理解

思考题

- 思考计算技术的变革与趋势
- 互联系统本质特征与差异？
- 中间件思想的意义？
- 为什么会出现集群与网格计算模式？
- 虚拟技术的应用
- 分布式计算模式的演变与思考
- 第一章研读及其后习题

Thanks !