

# 有关计算机体系结构的 思考汇总



樊建平 (fan@ict.ac.cn)  
中国科学院计算技术研究所  
HPC-OG课题组  
2003.10.11

# 内容

- 几类网格计算机
- Farming 计算模型与应用
- 指导计算机设计的ASC模型
- DSAG 的理论模型、程序模型
- 支持DSAG模型的操作系统

# 几类网格计算机

- 网格相关的概念
- 面向网格的计算机系统
- Dagger：一种散耦合的网格计算机  
体系结构
- 网格处理器
- 网格化的个人计算机
- 网格化的支持DSAG的计算机

# 网格相关的概念

- xxxGRID : bioGRID, PhyGRID, Sci DataGRID, EduGRID, TradeGRID
- GRID : 共享（计算资源—》各种资源）、动态、按需、服务等特征。研究协议、标准，理论模型与方法，平台、应用系统等。可以从Internet演化、应用特征、学科发展（处理网络空间）、关键特征
- GRIDxxx : 网格计算机、网格终端、网格磁盘等。
- 各种网格服务器在网格总的特征影响下形成。不同点主要体现在“拆”的方式方法不同，表现在部件定义、互连组织方式及计算机目标（高性能、服务器、个人计算机等）

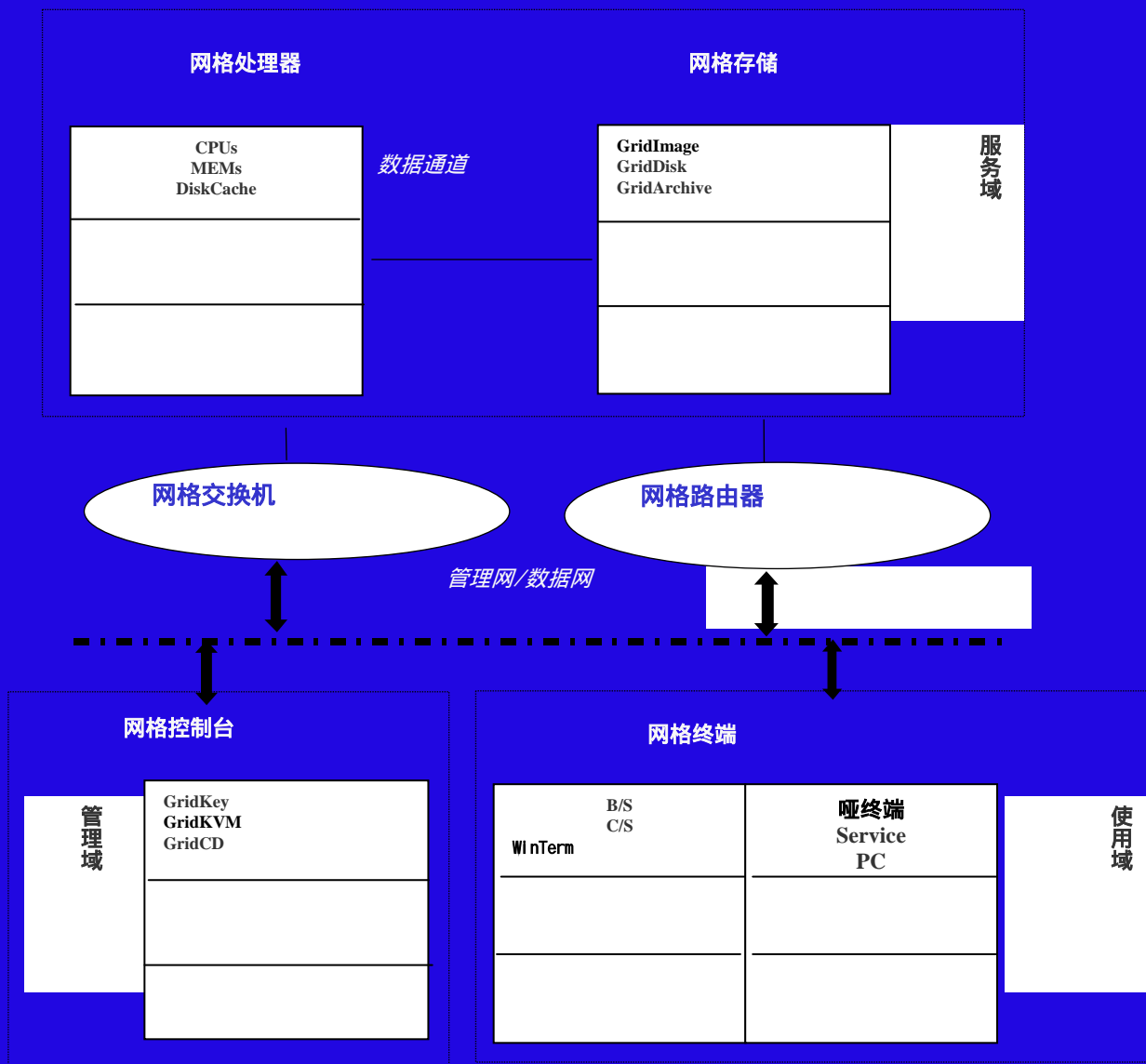
# 面向网格的计算机系统

- 面向网格的服务器：传统体系结构+传统操作系统+网格处理平台
- 面向网格的服务端（NC，PC）：支持网格处理协议

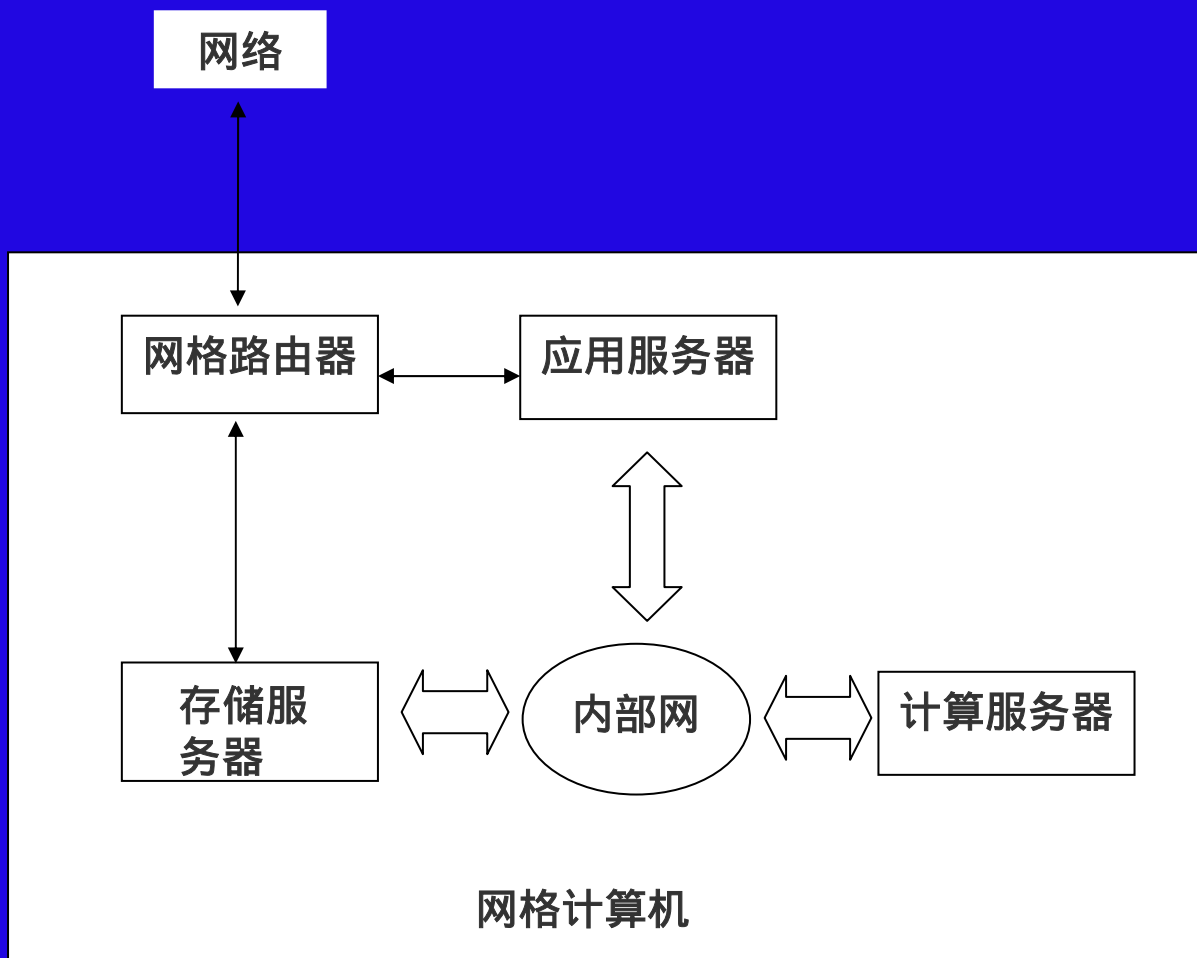
# Dagger：一种散耦合的网格计算机体系结构

## A Decoupled Architecture of Grid Computer

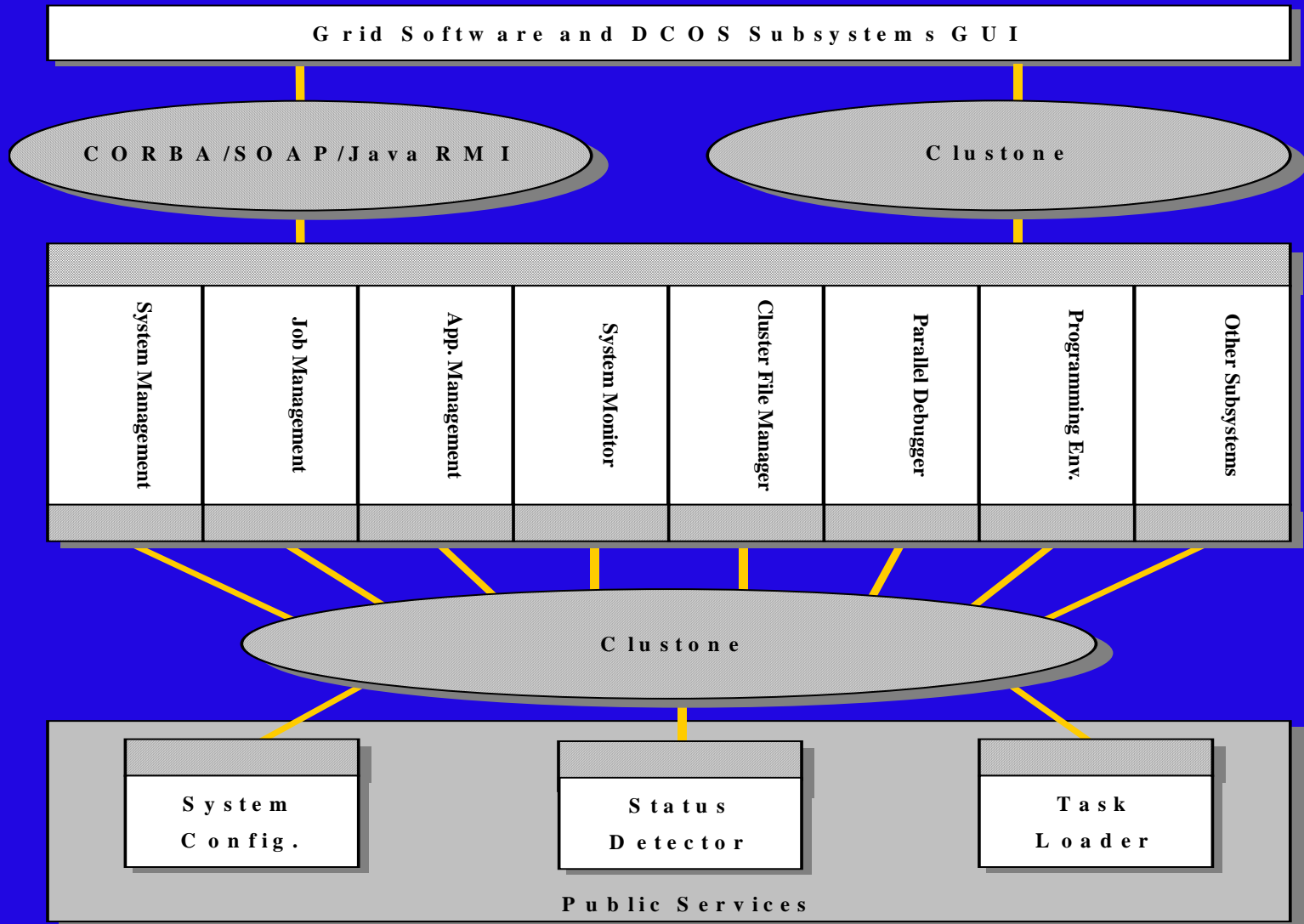
—由网格处理器、网格存储、网格控制台、网格终端、网格交换机、网格路由器6个网格部件和IntraGRID、InterGRID 2个网格构成方法（Construction Method）组成。有些网格部件又由一些网格零件组成，所谓“网格零件”即在网格环境下可见的功能单元



# Dagger的早期实践：体系结构



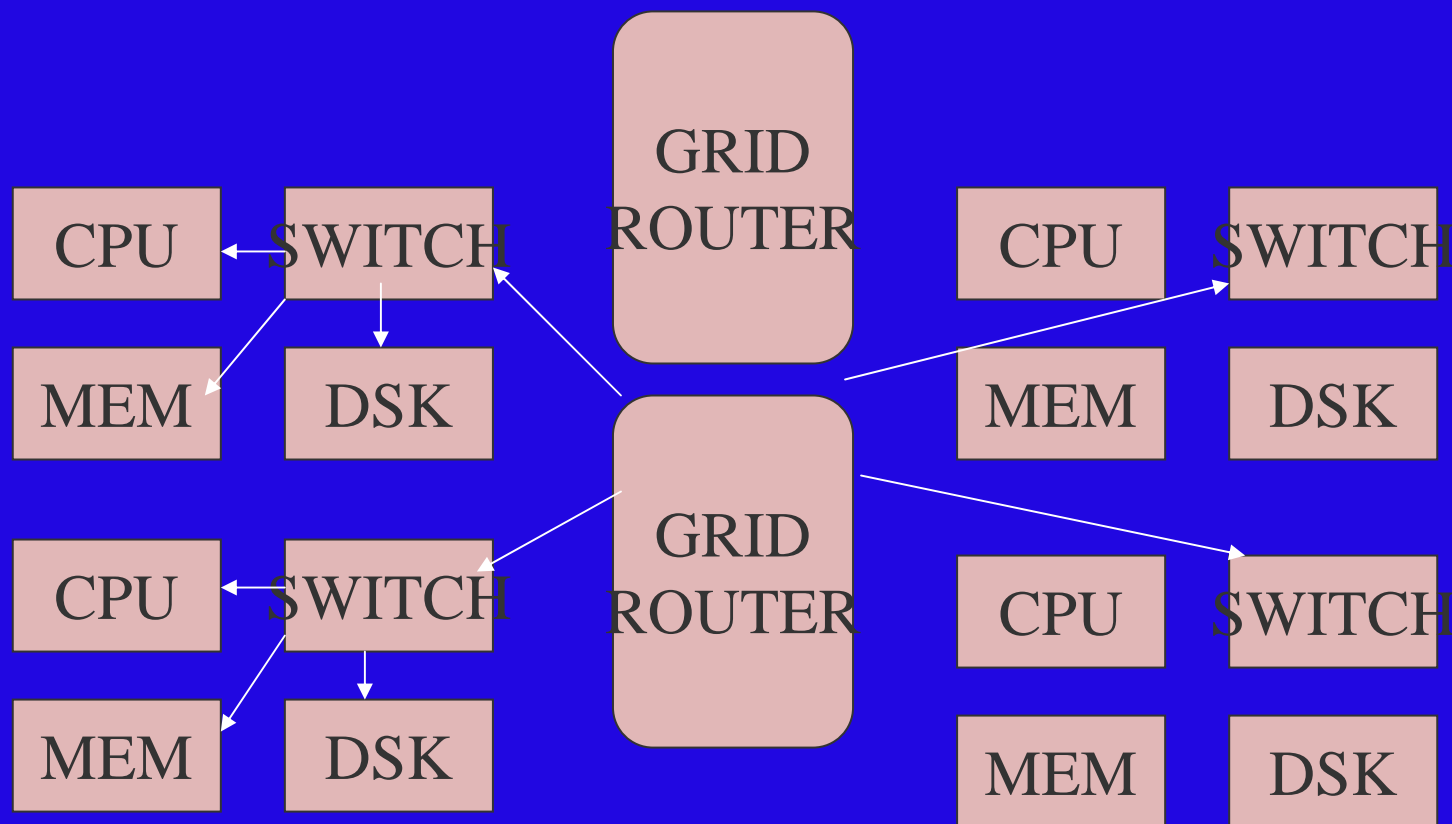
# 早期结构的操作系统：DCOS结构





# 网格处理器

—支持整个Internet是一台计算机的概念。  
是整个网格系统的发动机。

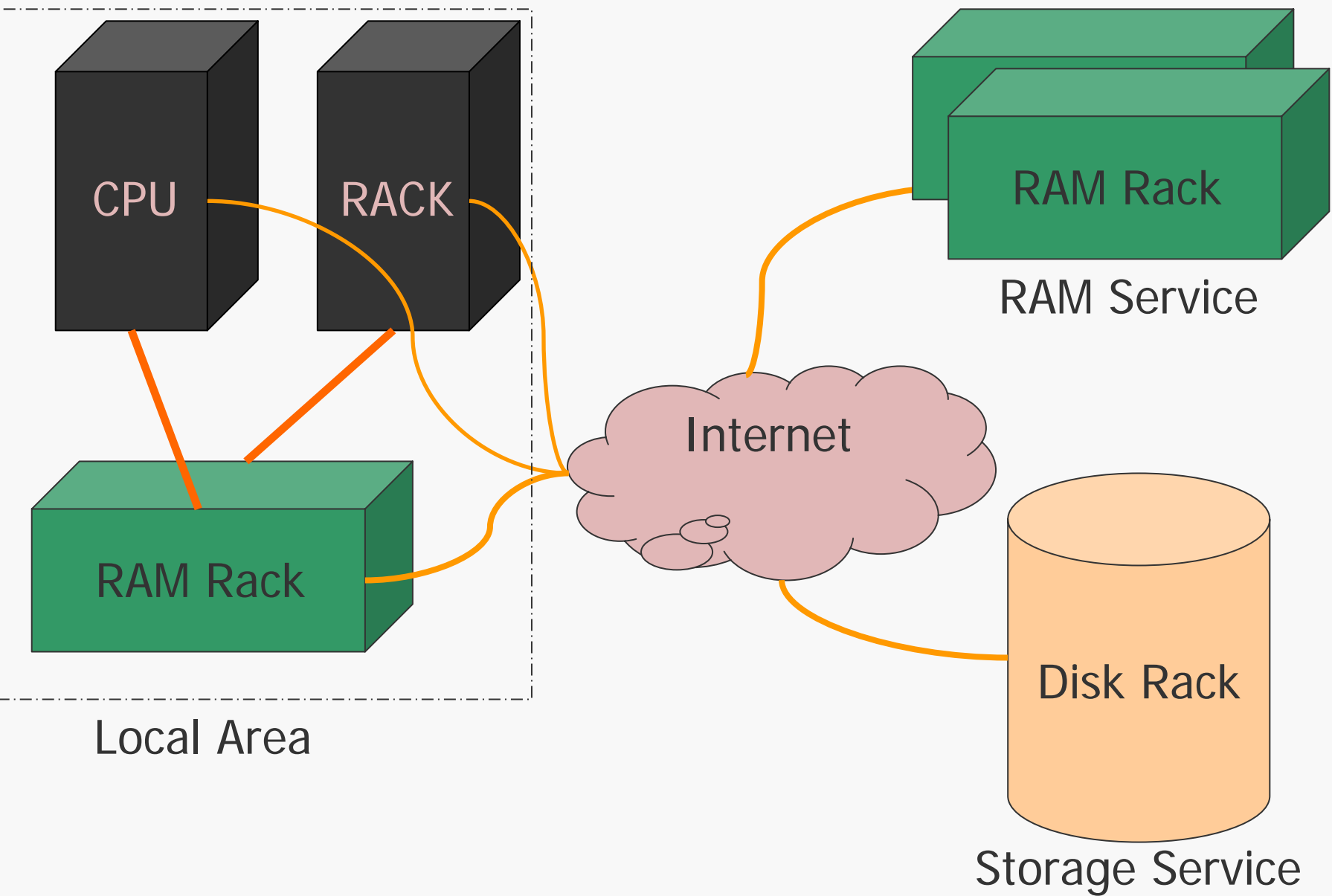


# 网格化的个人计算机

- 启发于联想的微网格概念：将家庭中使用的各种设备通过微网格协议互连。
- 网格化的个人计算机：定义连接PC内部设备（计算、显示、存储、交互、软件固件等）的网格协议。

# 网格化的支持DSAG的计算机

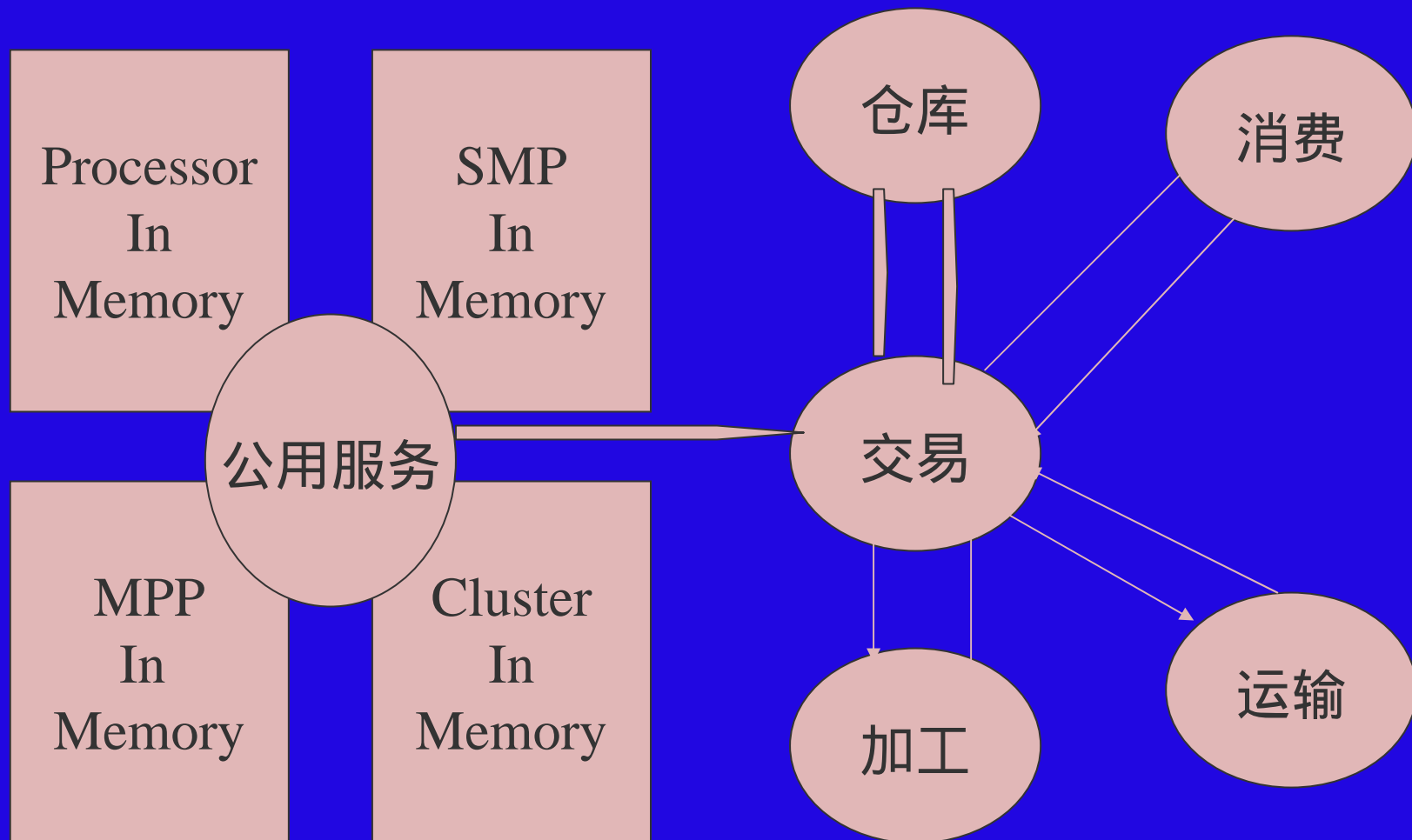
—面向大规模并行处理，利用网格理念，将计算机资源分拆为CPU组、MEM组、DISK组三类。通过光纤等高速互连手段实现体系结构快速重组，提高应用程序的高生产率。



# Farming 计算模型与应用

启发于原始农业生产方式的大规模并行处理。允许处理效率与互连带宽变化。生产过程在单户或村基础上完成。交易与深加工在连接度高、离用户近的地方进行。可以用于探索计算网格（Internet 为计算机）CPU分布与构成的思路之一。是否可以用于多粒度并行计算机的构成呢？

# Farming 计算模型



# 指导高性能计算机设计的ASC模型

算法设计人员使用计算机模型 (RAM, PRAM, LogP等) 进行设计与复杂度分析。计算机设计人员使用模拟器与 Benchmark程序对系统进行事先分析、事后测试。随着高性能构造器件多样化 (CPU: 商用、SOC、专用、FPGA、SIP等) 的发展及使用面的扩大 (xxx信息学、计算xxx), 发展ASC模型指导高性能计算机设计师越来越有必要。这也意味着高性能计算机设计师懂器件、体系、系统软件的同时, 必须懂应用系统。

# ASC模型（续）—例子

—针对生物信息学的需求，设计曙光4000H，指标为 100 TFL0PS 的计算机系统。假设有以下三种选择：光互连+商业PC主板、FPGA定制芯片+3D mesh、高性能SOC+2D Mesh，我们如何选择？如果还针对密码学，我们又如何选择呢？可以有一个统一的方法指导进行这类选择吗？



# ASC模型（续）——构成与层次

1、领域相关应用系统

2、基本算法及领域相关算法与程序

3、程序语言与环境：

4、管理软件：操作系统

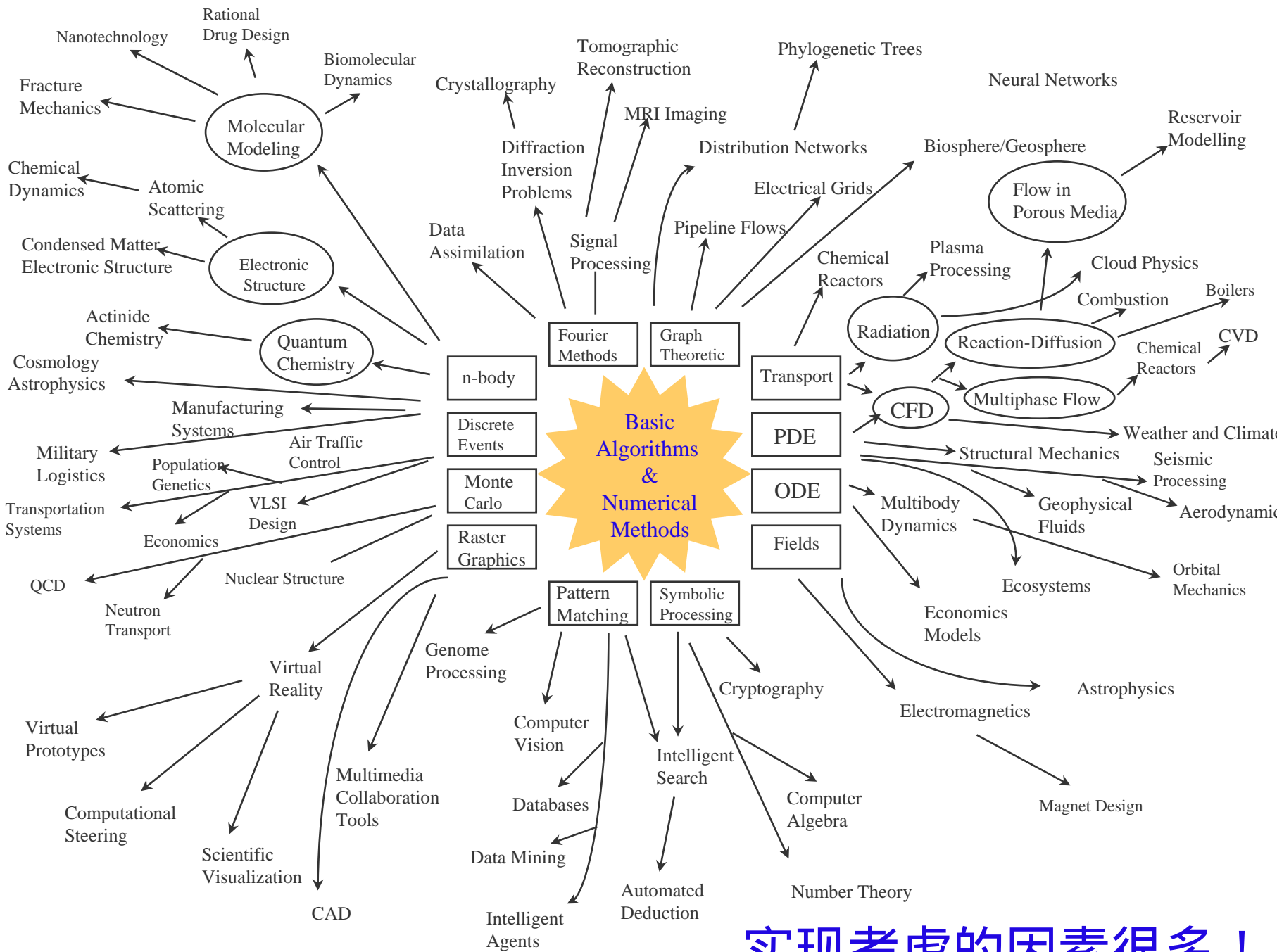
5、连接与体系结构层：互连带宽与方式，

6、器件与部件层：CPU、MEMROY、DSK、主板

# ASC模型（续）——功能与实现

——迫使计算机设计人员从应用（算法与系统）、软件（语言与操作系统）、硬件（器件与体系）三个层次多个角度（成本、研制周期、技术积累等）、分析与设计计算机系统。

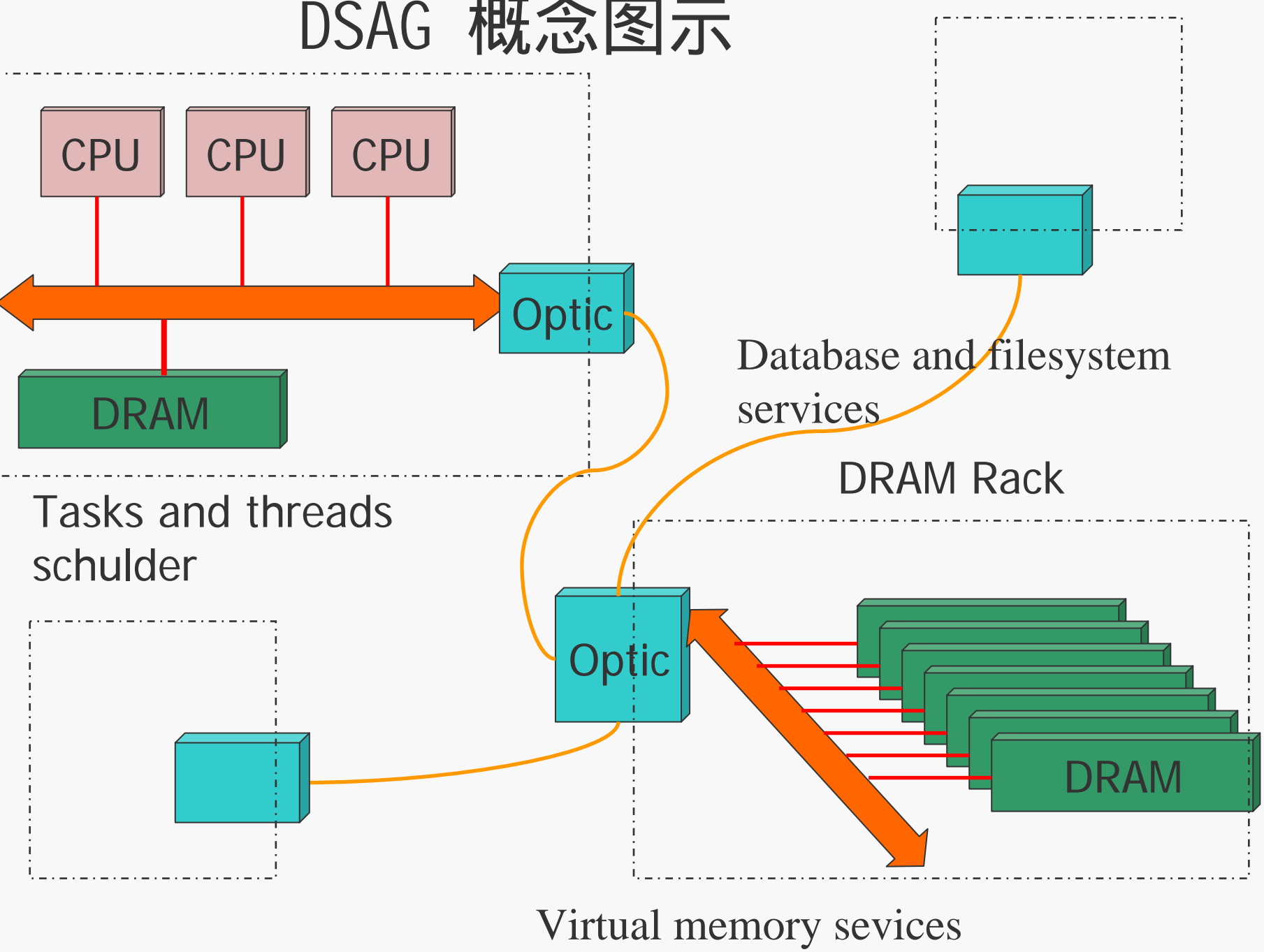
——实现考虑：借助自动控制相关理论、数学优化问题、成本交易模型等。



实现考虑的因素很多！

# DSAG 的理论模型、程序模型

# DSAG 概念图示



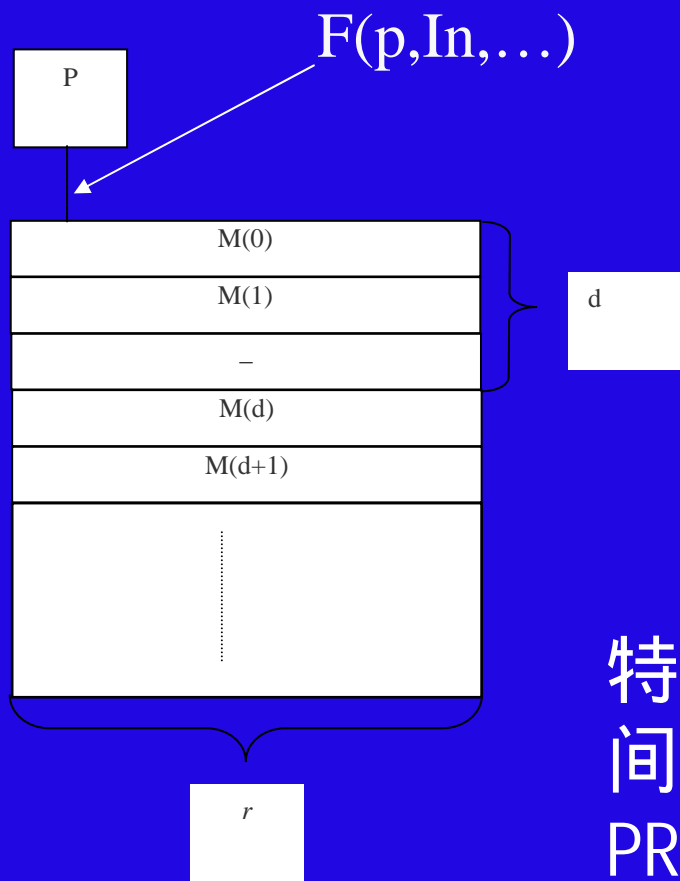
# 网格化的动态自组织计算机体系机构

## —DSAG体系结构

- 主要由CPU-BOX、MEM-BOX 和Disk-BOX构成： $\text{CPU}(N)+\text{MEM}(M)+\text{DSK}(L)$
- 通过动态映射可实现传统体系结构：
  - SMP:  $\text{CPU}(n)+\text{MEM}(m)+\text{DSK}(l)$
  - MPP:  $E[\text{CPU}(n)+\text{MEM}(m)]+\text{DSK}(l)$
  - Cluster:  $E[\text{CPU}(n)+\text{MEM}(m)+\text{DSK}(l)]$
  - NAS like:  $\text{CPU}(n)+E[\text{MEM}(m)+\text{DSK}(l)]$
- 还可以实现新的体系结构：
  - $\text{SMP}(0-1)+\text{MPP}(0-1)+\text{Cluster}(0-1)$
  - 动态D-SMP、动态D-MPP、动态D-Cluster
  - $\text{D-SMP}(0-1)+\text{D-MPP}(0-1)+\text{D-Cluster}(0-1)$
- Computer Architecture on Demand

# DSAG 的理论模型

## ■ RAM, CAM, PRAM 等已有模式



定义1. 一个RAM模型计算机 $C$ 是一个四元组，其中 $P$ 是计算机 $C$ 的处理器， $M$ 是 $C$ 的存储器， $In$ 是 $C$ 的指令集， $r$ 是 $C$ 的字长。存储器 $M$ 由无限个存储单元，，，...构成，每个存储单元含有 $r$ 个二进制位，其中 $r$ 是事先约定的正整数。通常约定， $M$ 的头 $d$ 个单元用作处理器 $P$ 的寄存器， $d$ 是事先约定的正整数。RAM模型计算机 $C$ 可以用图1描述。

特征：非线性执行， $P_1-P_n$ 之间按某种顺序执行。是否与PRAM模式等同？

# DSAG 的程序设计模型

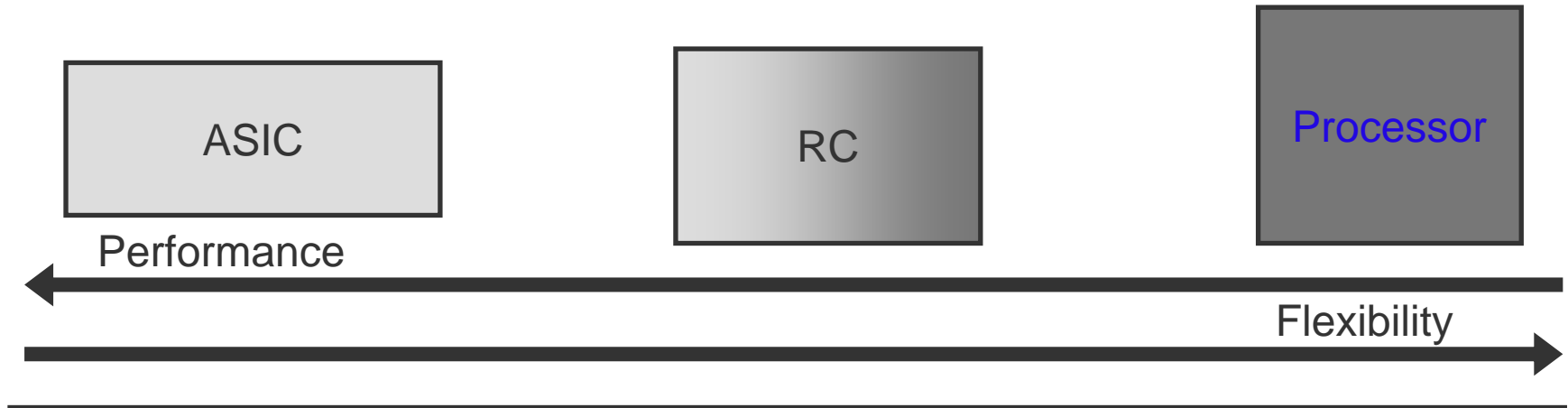
- 可重构计算的特征分析
  - 主要是FPGA技术发展带动的学科
  - 以最小的Gate No实现高速计算
  - 根据程序改变自己（预制而非动态）
- DSAG 支持的并程序特性分析



# RC(Reconfigurable Computing) 介绍

# Reconfigurable Computing - defined

- RC is the middle ground between ASICs and microprocessors. ASICs are the ultimate in speed but lack flexibility while processors have the ultimate in flexibility but lack speed.
- Its key feature is the ability to perform computations in hardware to increase performance, while retaining much of the flexibility of a software solution.



---

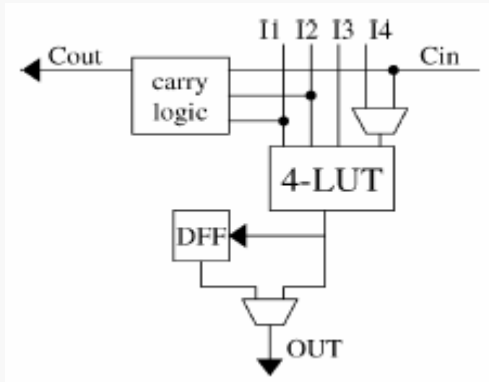
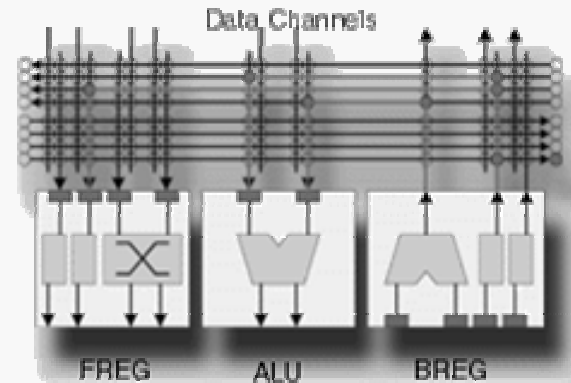
Names: Adaptive Computing (AC), Reconfigurable Computing (RC), Run Time Reconfiguration (RTR) & Dynamically Reconfigurable Computing

# The many names of RC

Fuzzy Hardware	Neural Networking	Transputing
Custom Computing	Genetic Computing	Soft Computing
Hyper-Specificity Computing	Reconfigurable Computing	Adaptive Computing
Hardware Virtualization	Transformable Computing	Instruction Set Metamorphosis

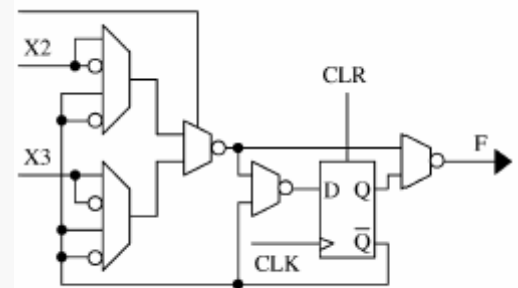
# RC Hardware Granularity

*Coarse grain* functional units consist of ALU and memory units



*Medium grain* Functional units consist of many input LUT<sub>X1</sub>

*Fine grain* Functional Units take in 2-3 inputs to produce a single output

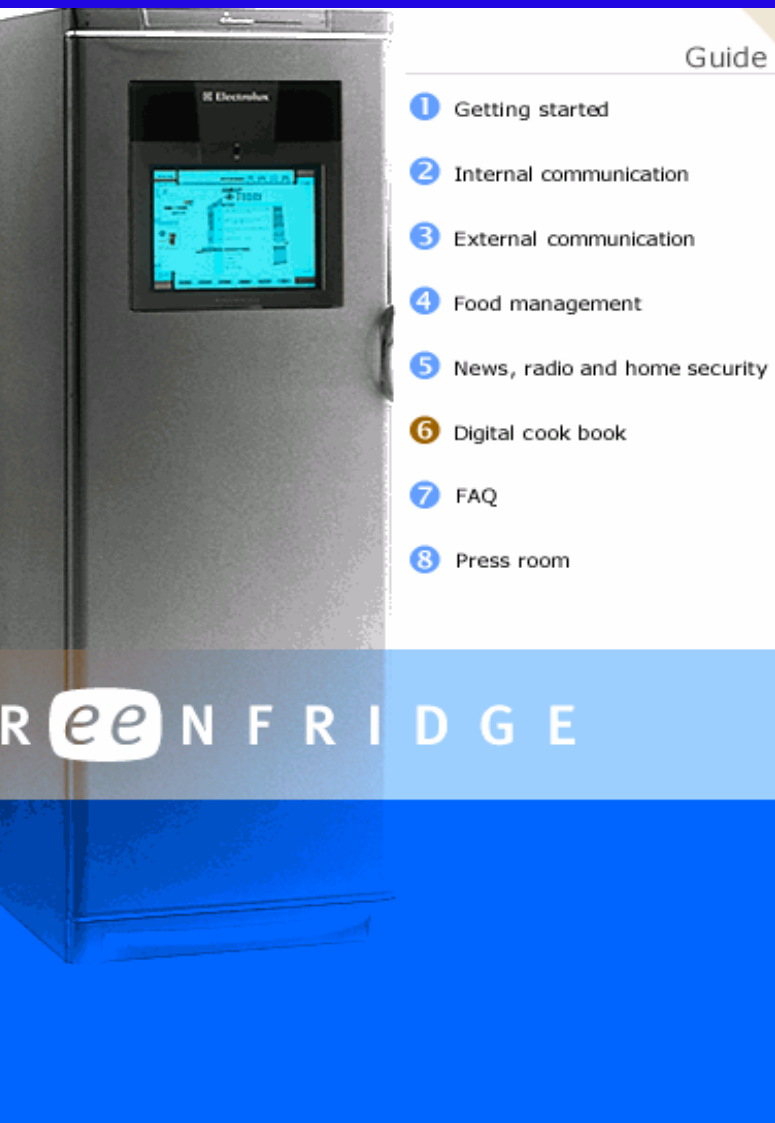


The functional unit from a Xilinx 6200 cell

# Application Domains

DSP	Robotics	Sorting	Wireless
Event-Driven Simulation	Data Format Converters	Parallel Computing	Cryptography / Security
General Computation	Human-Computer Interactions	Hough / Radon transforms	Optoelectronics
AI / neural networks	Video compression	Tunable Oscillators	Language recognition

# .... going into every type of application



# DSAG 并行程序设计模型分析

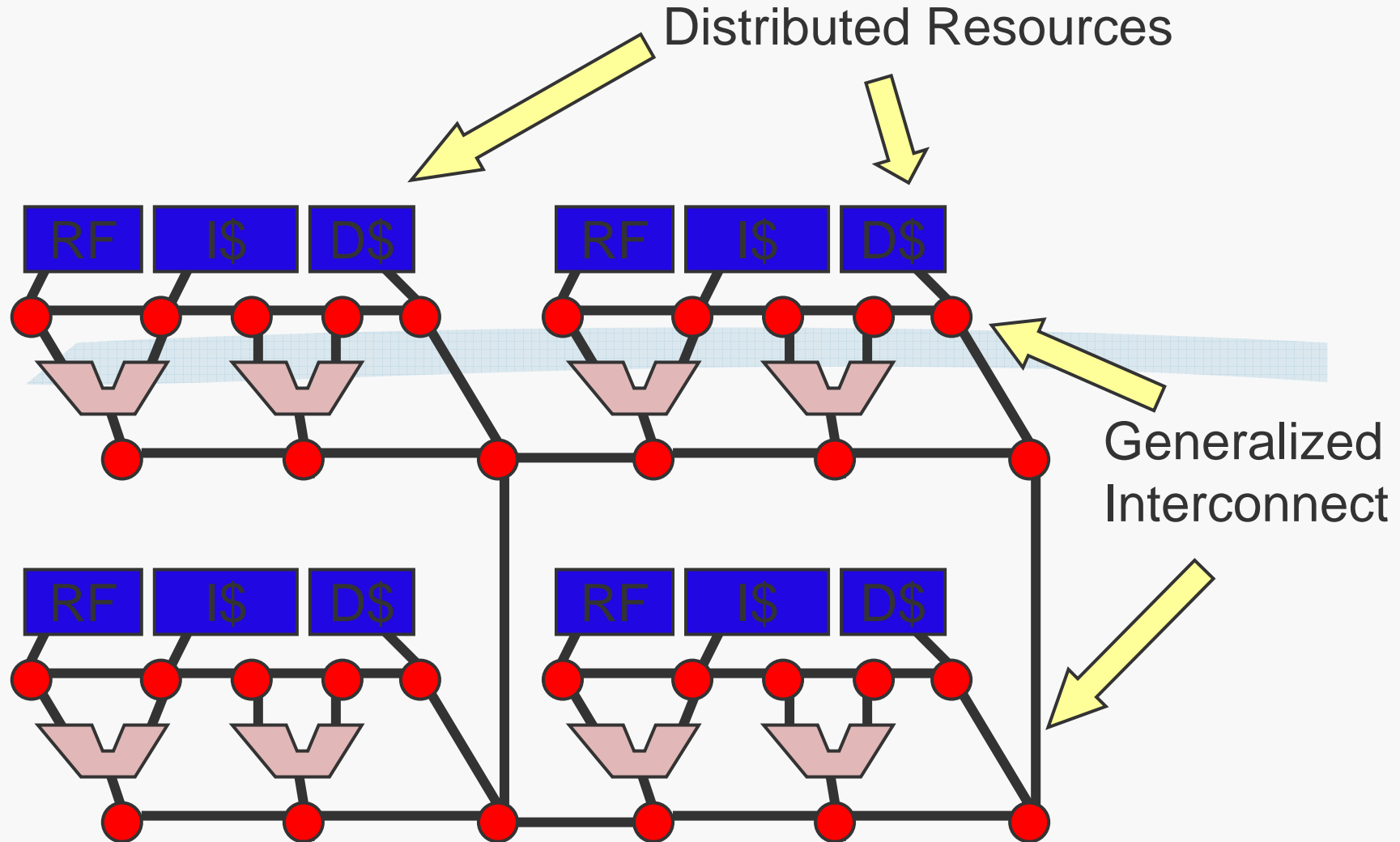
指令级并行：MIT RAW Project

线程级：DSAG

进程级：DSAG

任务级：DSAG

# Idea: distribute everything over a generalized on-chip network





# 支持DSAG体系结构的高性能计算机可 高效运行的应用程序类型

- 已存在的三类MPI、Pthread 和PVM 程序  
无需修改均可高效地运行于DSAG机器
- 编制新的应用程序时，可直接调用共享  
内存或消息传递算法模块，利用已有成  
果，加速程序的开发。
- 设计新算法解决新问题时，可以按问题  
本身的并行特征自由设计算法，无需考  
虑机器的特征。

# DSAG 对已有并行程序的支持

- 已存在的三类MPI、Pthread 和PVM 程序无需修改均可高效地运行于DSAG机器。
- 机器通过自动判定（装载过程中进行分析—执行代码二进制分析）或执行命令指定类型，将自己变成相应的硬件形态（SMP、MPP、Cluster)执行相应的程序。

# DSAG 的高生产率特征

- 编制新的应用程序时，可直接调用共享内存或消息传递算法模块，利用已有成果，加速程序的开发。一个应用程序可能包括对三类结构库函数的并行调用。例如程序员开发通过投票方式确定基因比对结果的程序（一组数据调用三组函数库独立处理，结果比对，2：1为执行完），机器将自动调整为三部分（SMP、MPP、Cluster），并行执行三个独立的程序，数据可以共享！

# DSAG 新的程序设计模式

- 设计新算法解决新问题时，可以按问题本身的并行特征自由设计算法，无需考虑机器的特征。
- 例如对WEB空间的内容进行过滤与扫描时，相同数据用不同pattern过滤时，机器可以手动或自动变成MISD(Multi-Instr. Single Data) 计算机进行处理。

# 支持DSAG模型的操作系统

- microkernel in every device
- Physical Single address spaces
- Parallel OS running at Host node
- LINUX Interface with computing library.

谢

谢!

