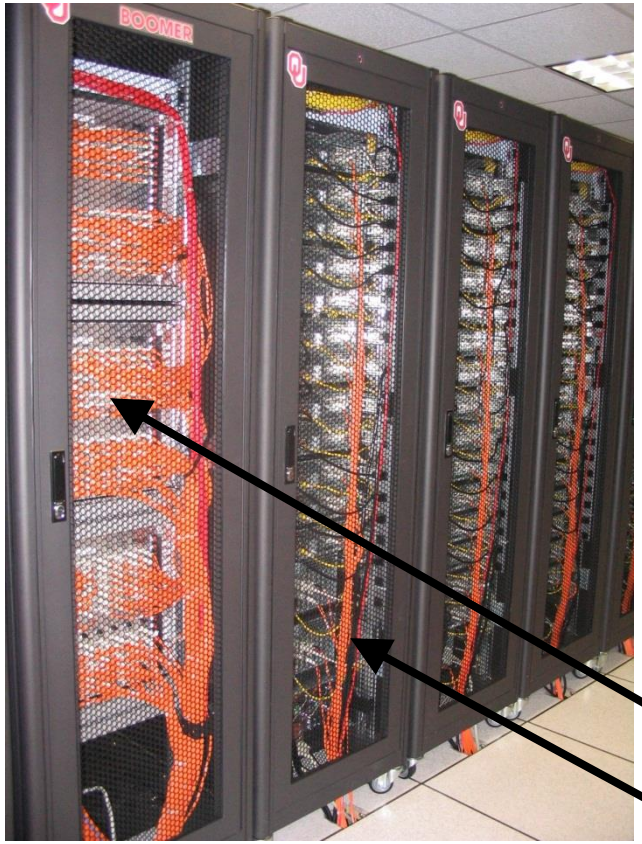
第6章 集群、网格和云计算

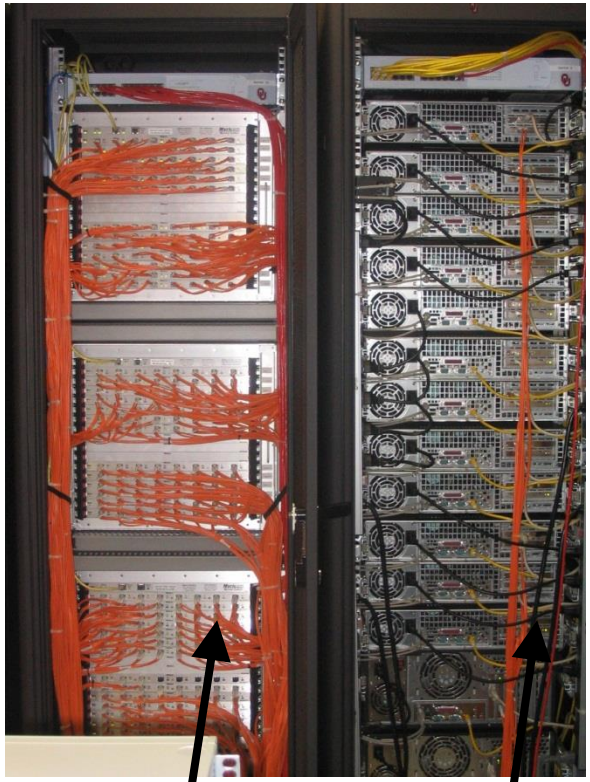
目录

6.1 集群概述

- 集群计算机系统及其特点
- 集群系统的中间件
- 集群系统的通信软件和网络服务
- 云计算
- 实例

An Actual Cluster



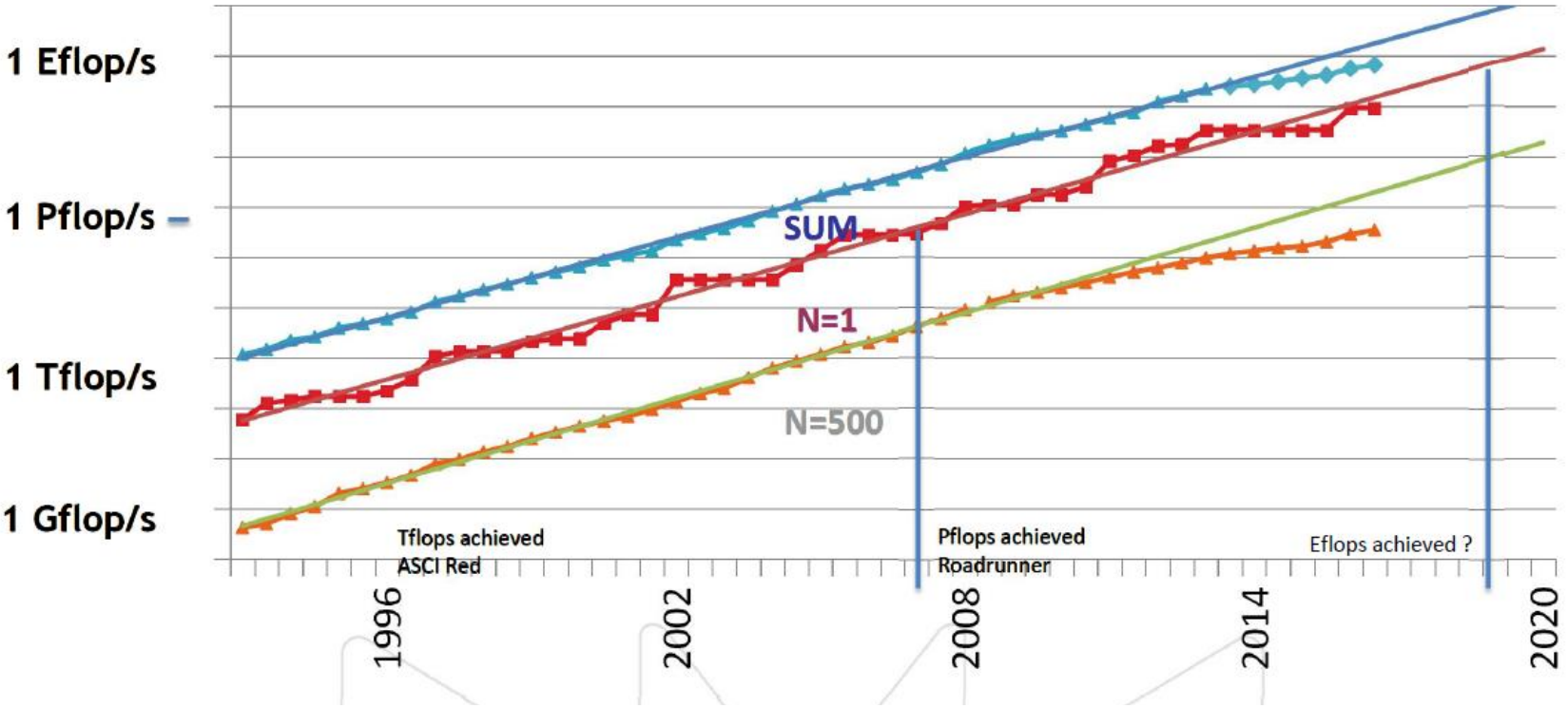


Interconnect

Nodes

超级计算机性能增长趋势

- Top500性能增长的趋势



Meta → Gega → Tera → Peta → Exa

10^6 → 10^9 → 10^{12} → 10^{15} → 10^{18}

集群式计算机产生的原因

- **集群（Cluster）**计算机系统能够以较短的研制周期、集成最新技术、汇集多台计算机的力量，达到了较高的性能价格比，其技术发展在国际上受到重视。
- 通过高速互连网络把通用的计算机（如高档计算机、工作站或PC机）联结起来，采用**消息传递机制(MPI、PVM等)**，向最终用户提供单一并行编程环境和计算资源，因此它通常也被称作“计算机群”，“工作站群”，“工作站网络”，“网络并行计算”等。

集群系统的概念和组成

- 集群系统是利用**高速通信网络**将一组计算节点连接起来
- 在并行程序设计和集成开发环境支撑下**统一调度**、协调处理；
- 集群系统中的主机和网络可以**同构**或**异构**
- 利用**消息传递方式**实现机间的通信
- 建立在**操作系统**上的**并行编程环境**完成系统的资源管理及相互协作。

集群系统的概念和组成

- 集群系统是利用**高速通信网络**将一组计算节点连接起来
- 在并行程序设计和集成开发环境支撑下**统一调度**、协调处理；
- 集群系统中的主机和网络可以**同构**或**异构**
- 利用**消息传递方式**实现机间的通信
- 建立在**操作系统**上的**并行编程环境**完成系统的资源管理及相互协作。

6.1 集群概述

1. 集群系统的组成

集群系统是利用高速网络将一组高性能工作站或高档PC机连接起来，在并行程序设计以及可视化人机交互集成开发环境支持下，统一调度，协调处理，实现高效并行处理的计算机系统。

Cluster、NOW、COW

从结构和结点间的通信方式来看，属于分布存储系统。

集群系统的关键技术

(1)高效的通信系统

在用户空间实现通信协议

精简通信协议

Active Message通信机制

(2)并行程序设计环境

PVM(Parallel Virtual Machine)

开始于1989年夏天,美国橡树岭国家实验室(ORNL)；是一套并行计算工具软件，支持多用户及多任务运行；支持多种结构的计算机，工作站、并行机以及向量机等；

支持C、C++和Fortran语言；自由软件，使用非常广泛；编程模型可以是SPMD或MPMD；具有容错功能，当发现一个结点出故障时，自动将之删除

MPI(Message Passing Interface)

在1992年11月至1994年元月产生。

能用于大多数并行计算机、计算机机群和异构网络环境，支持C和Fortran两种语言,编程模型采用SPMD

Express

美国Parasoft公司推出；能在不同的硬件环境上运行；支持C和Fortran两种程序设计语言。

(3)并行程序设计语言

在多处理机系统中，必须用并行程序设计语言编写程序。或者把已经用串行语言编写的程序转换成并行语言程序之后，才能在多处理机系统上运行。

(4) 负载平衡技术

一个大任务可分解为多个子任务，把多个子任务分配到各个处理结点上并行执行的技术称为负载平衡技术

对于由异构处理结点构成的并行系统，相同的负载在各结点上的运行时间可能不同。因此，准确的负载定义应是负载量与结点处理能力的比值

负载平衡技术的核心就是调度算法，即将各个任务比较均衡地分布到不同的处理结点上并行计算，从而使各结点的利用率达到最大。

(5)并行程序调试技术

用并行程序设计语言编写程序，比用串行程序设计语言更容易出错，因此，在多处理机系统中，用并行程序设计语言编写程序更加依赖于并行调试工具。

并行程序调试的主要困难：

并行程序的执行过程不能重现。

(6)可靠性技术

在多处理机上运行的程序通常比较大，程序执行时间很长（几十个小时或几十天）。如果在程序执行过程中出现偶然故障（如电源掉电、磁盘满、某一台处理机故障等），则整个运算过程要从头开始。

定时设置检查点，保存现场信息。当出现故障时，只要回复到上一个检查点，不必从头开始执行。

集群系统的概念和组成

- 集群系统是利用**高速通信网络**将一组计算节点连接起来
- 在并行程序设计和集成开发环境支撑下**统一调度**、协调处理；
- 集群系统中的主机和网络可以**同构**或**异构**
- 利用**消息传递方式**实现机间的通信
- 建立在**操作系统**上的**并行编程环境**完成系统的资源管理及相互协作。

集群式计算机的重要部件

- 多个高性能计算机
- 基于内核的操作系统
- 高性能网络/开关
- 网络接口卡
- 快速通信协议或服务
- 集群中间件（单一映像系统、资源管理与调度）
- 并行编程环境和工具
- 应用程序

高性能计算节点的系统软件

- 高性能计算节点研究并开发具有自主知识产权的系统管理软件。它由机群管理系统、网上用户任务管理系统和局部资源管理系统组成，它们主要有：
  - 单一系统映像系统管理软件
  - 用户管理、任务管理和资源管理软件
  - 高速通信支撑软件
  - 负载平衡和软件容错技术
  - 资源接口软件
  - 可视化并行计算性能评价工具软件
  - 网络并行计算查询和管理
  - 网络远程登录和管理
  - 关系式数据库
  - 支持检查点操作和进程迁移

- **集群计算机系统能够以较短的研制周期**、集成最新技术、汇集多台计算机的力量，达到较高的性能价格比，通过高速互连网络把通用计算机（如高档计算机、工作站或PC）连接起来，采用消息传递机制（MPI，PVM等），向最终用户提供单一并行编程环境和计算资源，因此它通常也称为“计算机群”、“工作站群”、“工作站网络”或“网络并行计算”等。

表 6-1　TOP 500 中集群系统的发展

比较项目	时间				
	2005 年 11 月	2010 年 11 月	2015 年 11 月	2018 年 11 月	2019 年 6 月
集群数量（台）	361	414	426	442	453
最快集群在 TOP 500 中的名次	5	3	1	1	1
最快集群配置的处理器数目（个）	8000	120640	3120000	2397824	2414592
最快集群的峰值速度（GFLOPS）	57600	2984300	54902400	200794.88（TFLOPS）	200794.88（TFLOPS）
最快集群的 Linpack 指标（GFLOPS）	38270	1271000	33862700	143500（TFLOPS）	148600（TFLOPS）

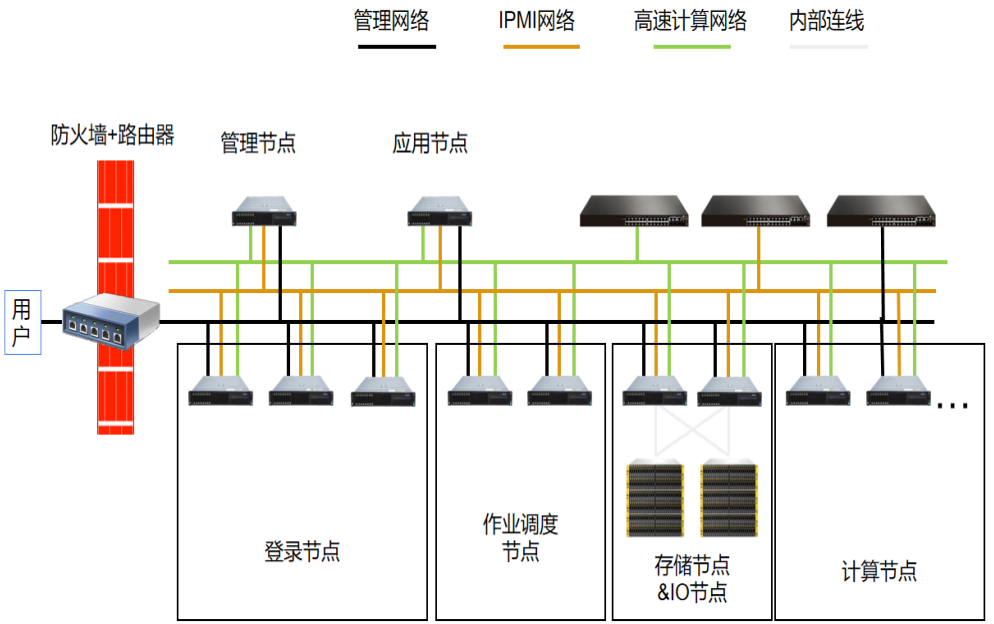
集群计算机根据研制理念的不同，可以分为：

- NOW类型集群：追求高速通信
- Beowulf类型集群：尽可能使用现成硬件、免费系统软件、基于TCP/IP建立通信库

广义地讲，由SMP节点构成的集群（称为CLUMPS或Constellations）。

6.1集群概述

集群计算机系统的应用面非常广，除了科学计算外，还可以用于事务处理，如用作Web服务器、网络文件服务器、超级Mail服务器以及海量廉价存储系统等。集群计算机的基本结构如右图所示，包含负责对集群进行监控和管理等工作的管理节点、负责完成计算任务的计算节点、负责存储数据的集群存储、负责节点间互联的高速网络。



6.1集群概述

集群根据性能特性，集群计算机可以分为三类：

高可用性（High Availability）集群	简称HA集群。这类集群致力于提供高度可靠的服务—利用集群系统的容错性对外提供7×24小时的不间断服务，如高可用的文件服务器、数据库服务等关键应用。
负载均衡（Load Balance）集群	这类集群可以使任务在集群中尽可能平均地分摊到不同的计算节点处理，充分利用集群的处理能力，提高对任务的处理效率，如LVS（Linux Virtual Server，Linux虚拟服务器）
高性能计算（High Performance Computing）集群	简称HPC集群。这种集群上运行的是专门开发的并行应用程序（如MPI、Hadoop、Spark等），它可以把一个问题的计算任务分配到多个计算节点上，利用这些计算节点的资源来完成任务，从而完成单机不能胜任的工作（如果问题规模太大，单机计算速度太慢）。这类集群致力于提供单个计算机所不能提供的强大的计算能力，如天气预报、石油勘探与油藏模拟、分子模拟、生物计算等。

6.1集群概述

集群我国在研制高性能计算机方面，已经取得很多成就。这些高性能计算机主要分为如下三大类：

① **PVP向量型超级计算机**，如国防科技大学1983年研制的银河Ⅰ（1亿次/秒）、1994年研制的银河Ⅱ（10亿次/秒）。

② **MPP大规模并行处理超级计算机**，如中国科学院计算技术研究所1995年研制的曙光1000（25亿次/秒）、国防科技大学1997年研制的银河Ⅲ（130亿次/秒）和2009年研制的天河一号（4701万亿次/秒）、中国国家并行计算机工程技术研究中心2016年研制的神威太湖之光（12.5亿亿次/秒）。

③ **集群计算机，**清华大学1999年研制的THNPSC-1（320亿次/秒）、中国科学院计算技术研究所1999年研制的曙光2000-Ⅱ（1117亿次/秒）、上海大学2000年研制的自强2000（4500亿次/秒）、国防科学技术大学2013年研制的天河二号（10.07亿亿次/秒）。

6.2集群系统的软硬件组成

计算节点

计算节点是集群系统中数量最多的节点，是用来完成用户提交的计算任务。集群的性能取决于所有计算节点的性能及其发挥情况。因此计算节点需要有强大的性能，计算机的性能不仅取决于计算性能、还取决于**存储性能和通信性能**，是一个**系统整体的综合**表现。

不同应用对于系统的计算、存储和通信的需求不同，
性能的发挥也受到计算节点内存大小、
计算部件性能以及网卡性能等因素的限制。
随着各种加速部件（**特别是GPU**）的发展，
在计算节点上配置多块GPU或者其他加速部件来提高浮点计算性能等成了大势所趋。

6.2.2 网络

集群系统一般有三套网络，

一套IPMI网络用作底层硬件管理；

一套高速互联网络用作操作系统管理；

一套高速计算互联网络，主要负责计算软件在计算时集群节点之间的数据通信。除了计算网络，另2套网络对网络性能要求不高，一般使用性价比高的以太网。

三套网络中的计算网络是十分重要的，因为集群系统的节点比较多，使得数据在两个节点之间流动需要经过多个交换机。这导致计算网络延迟高，网络的高延迟在需要频繁通信的应用中会形成性能瓶颈。

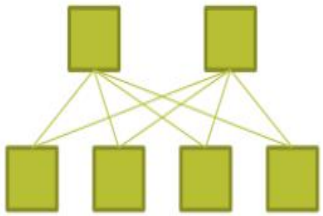
6.2.2 网络

三套网络中的**计算网络是十分重要的**

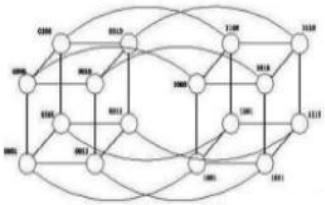
如下图所示的胖树等网络，

网络拓扑结构会带来线路、设备的可靠性、安全性问题，

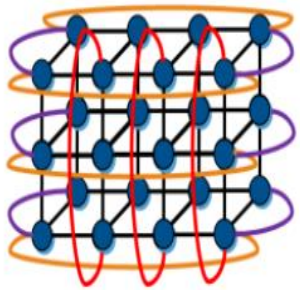
这些问题都需要由通信协议及其实现的通信软件和网络服务来解决。



Fat Tree



Hypercube



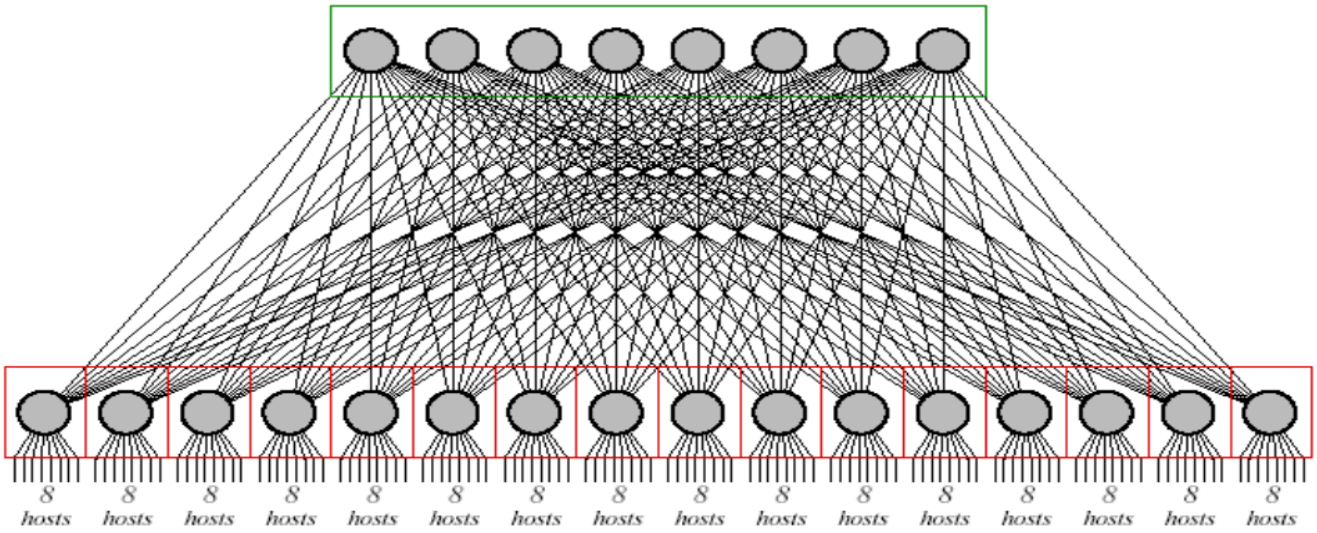
Torus

6.2.2 网络

高性能的集群计算机中，往往使用轻量级通信协议(如RDMA)。

有三类RDMA网络，分别是Infiniband、RoCE、iWARP。其中Infiniband(简称IB)是一种专为RDMA设计的网络，性能上Infiniband网络最好，但就性价比而言还是RoCE和iWARP比较高。

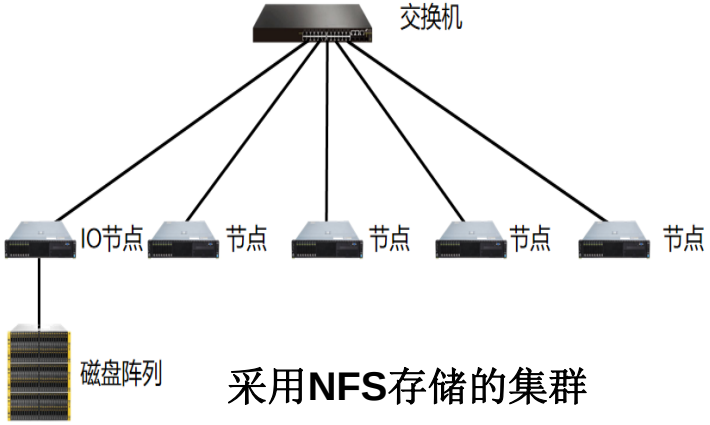
下图是128节点InfiniBand网络拓扑。



6.2.3 存储节点

存储是集群计算机系统的一个重要组成，负责保存数据。在集群系统中把数据集中起来通过一个存储系统提供数据管理和读写服务。存储系统由文件系统和存储硬件组成。文件系统可以采用NFS(Network File System)或者并行文件系统。

右图所示的是采用NFS的集群。NFS即网络文件系统，是文件系统之上的一个网络抽象，它允许网络中的计算机之间共享资源。本地NFS的客户端应用可以透明地读写位于远端NFS服务器上的文件。可以在集群中选择一个配置较好且支持配置大量硬盘的服务器作为I/O节点，这种方式简单、价格低廉，但是存储性能低。

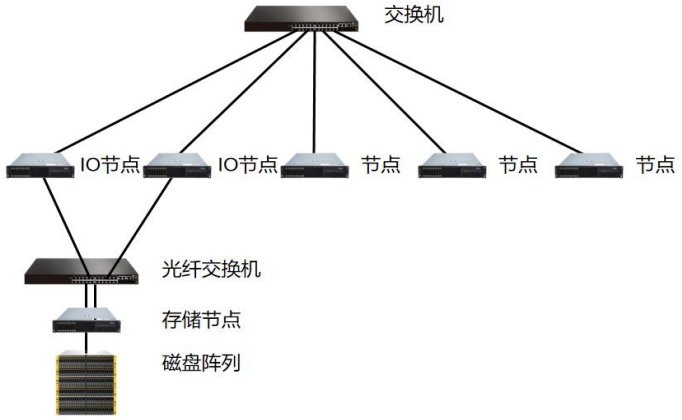


采用NFS存储的集群

6.2.3 存储节点

并行存储系统

由并行文件系统和并行存储硬件组成。并行存储系统需要使用并行文件系统才能充分发挥大量存储设备的性能。并行文件系统有开源的Lustre、商用的GPFS、ParaStor等。这些并行文件系统一般包括索引模块和数据管理模块，将这些模块部署在多个通用服务器上，配置成相应的I/O节点，实现并行存储统一管理和通过多个I/O节点对后端存储节点的并行访问。并行存储系统的性能和I/O节点的数量等有关。



采用并行存储的集群

6.2.4 管理节点

管理节点的主要功能是通过各种软件对集群系统进行安装、维护、运行状态监控、资源管理和作业管理等。

集群系统的管理和监控系统

集群系统的管理和监控系统

集群系统一般都由大量的功能不同的计算机构成，管理工作量随着计算机数量的增多而大大增加，需要管理系统来提高系统的管理和维护效率。

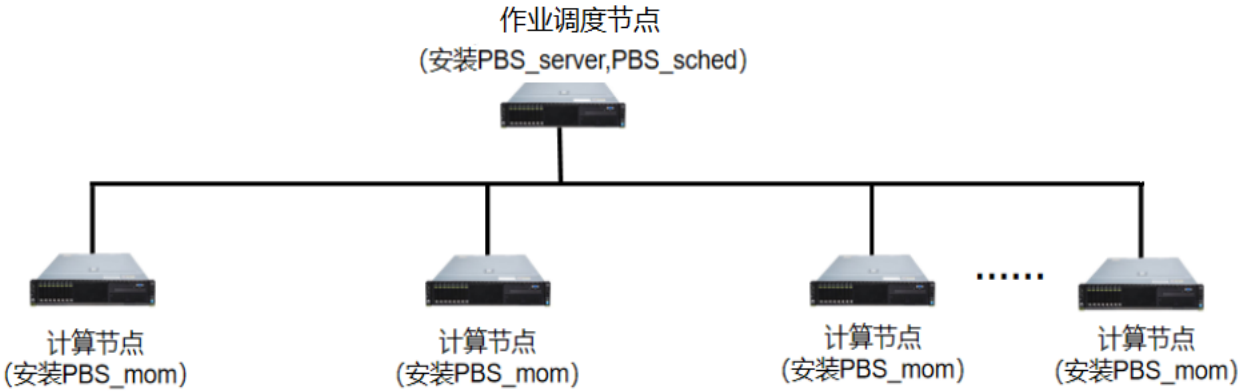
用于集群性能监控的软件有很多，如早期比较著名的开源软件ganglia。监控软件一般可以对所有服务器的进程、CPU、GPU、内存、网络、磁盘使用情况等信息进行抓取和显示。

6.2.4 管理节点

集群的作业管理系统

集群系统的管理和监控系统

主要功能是作业提交、资源监控、调度规则设置、计算节点设置、任务调度、结果返回等。它由三部分组成：用户服务器、作业调度器和资源管理器。常用的作业管理系统有PBS，LSF等。PBS的目前包括OpenPBS，PBS Pro和Torque三个主要分支。其中OpenPBS是最早的作业管理系统之一。下图是OpenPBS安装示意图。



6.2.5 MPI并行编程

MPI（Message Passing Interface，消息传递接口）是一种消息传递接口，用于实现基于多进程的并行编程。MPI是基于FORTRAN或者C/C++的一个实现进程间通信的库，而不是一门新编程语言。

最基本的MPI函数：

(1)	MPI_Init();	初始化MPI环境
(2)	MPI_Comm_size();	获取进程数量
(3)	MPI_Comm_rank();	获取本进程进程号
(4)	MPI_Finalize();	退出MPI环境
(5)	MPI_Send();	点对点发送信息
(6)	MPI_Recv();	点对点接收信息
(7)	MPI_Bcast();	广播
(8)	MPI_Reduce();	规约
(9)	MPI_Gather();	收集
(10)	MPI_Scatter();	散发
(11)	MPI_Barrier();	同步

6.2.5 MPI并行编程

(1) MPI_Init (int* argc, char** argv[])

初始化并行环境。

(2) MPI_Comm_size (MPI_Comm comm, int* size)

获得通信域comm中规定的group包含的进程的数量。

(3) MPI_Comm_rank (MPI_Comm comm, int* rank)

得到本进程在通信空间中的rank值，即在组中的逻辑编号（该rank值为0到进程总数-1间的整数，相当于并行进程的ID）。MPI编程时主要通过这个编号对进程进行区分和任务分配。

(4) MPI_Finalize (void)

该函数的作用是退出MPI系统，释放占用的资源。

6.2.5 MPI并行编程

MPI程序基本结构：

```
#include "mpi.h"
#include <stdio.h>
#include <math.h>
void main(argc,argv)
int argc;
char *argv[];
{   int  myid, numprocs, i=0;
    int  namelen;
    char processor_name[MPI_MAX_PROCESSOR_NAME];

    MPI_Init(&argc,&argv);
    MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
    MPI_Comm_rank(MPI_COMM_WORLD,&myid);
    MPI_Get_processor_name(processor_name,&namelen);
    fprintf(stderr,"Process %d on %s\n",
            myid, processor_name);
//××××××××××××××
////××××××××××××××
 MPI_Finalize();
    }
```

6.2.5 MPI并行编程

(5) MPI_Send(buf, count, datatype, dest, tag, comm)

该函数的作用是将从buf开始的count个数据发送给进程编号为dest的进程。

buf：需要发送的数据的地址。

count：需要发送的数据的个数（注意，不是长度。例如要发送一个int整数，这里就填写1；如要是发送“China”字符串，这里就填写6。C语言中字符串末有一个结束符，需要多一位）。

datatype：需要发送的MPI_Datatype数据类型。MPI_Datatype是MPI定义的数据类型，可在MPI文档内找到常用数据类型和MPI定义的数据类型对应表。

dest：目标进程号。需要发送给哪个进程，就填写目标进程号。

tag：数据标签。接收方需要有相同的消息标签才能接收该数据。

comm：通信域。表示需要向哪个组发送数据。

6.2.5 MPI并行编程

(6) MPI_Recv(buf, count, datatype, source, tag, comm, status)

该函数的作用是将接收到的数据保存在buf里。

buf：保存接收到的数据的地址。

count：接收数据的个数。它是接收数据长度的上界，具体接收到的数据长度可通过调用MPI_Get_count 函数得到。需要注意的是，MPI中发送和接收的数据数量可以不等，发送数据数量可以大于等于接收数量，但是如果准备接收数据的数量大于发送数据数量会造成死锁。

datatype：要接收的MPI_Datatype数据类型。

tag：数据标签，需要与发送方的tag值相同的数据标签才能接收该数据。

comm：通信域。

status：MPI_Status数据状态。接收函数返回时，将在这个参数指示的变量中存放实际接收数据的状态信息，包括数据的源进程标识、数据标签等。

6.2.5 MPI并行编程

(7) MPI_Bcast(buf, count, datatype, rank, comm)

该函数的作用是由进程rank向所有进程发送数据类型为datatype、从buf开始的count个数据。

(8) MPI_Reduce(sendbuf, recvbuf, count, datatype, op, rank, comm)

该函数的作用是所有进程对从sendbuf开始的count个元素做op运算，并依次存放在进程rank上recvbuf开始的缓冲区。其中，op运算如下：

种类	操作
MPI_MAX	最大值
MPI_MIN	最小值
MPI_SUM	求和
MPI_PROD	求积
MPI_LAND	逻辑与
MPI_BAND	按位与

种类	操作
MPI_LOR	逻辑或
MPI_BOR	按位或
MPI_LXOR	逻辑异或
MPI_BXOR	按位异或
MPI_MAXLOC	最大值且相应位置
MPI_MINLOC	最小值且相应位置

6.2.5 MPI并行编程

（9）MPI_Gather(sendbuf, sendcount, sendtype, recvbuf, recvcount, recvtype, rank, comm)

该函数的作用是进程rank向所有进程（包括自己）收集数据，每个进程从地址sendbuf开始向进程rank发送sendcount个数据。进程rank将接收的数据按进程号存放到从地址recvbuf开始的缓冲区，对应每个进程接收缓冲区的大小为recvcount。

（10）MPI_Scatter(sendbuf, sendcount, sendtype, recvbuf, recvcount, recvtype, rank, comm)

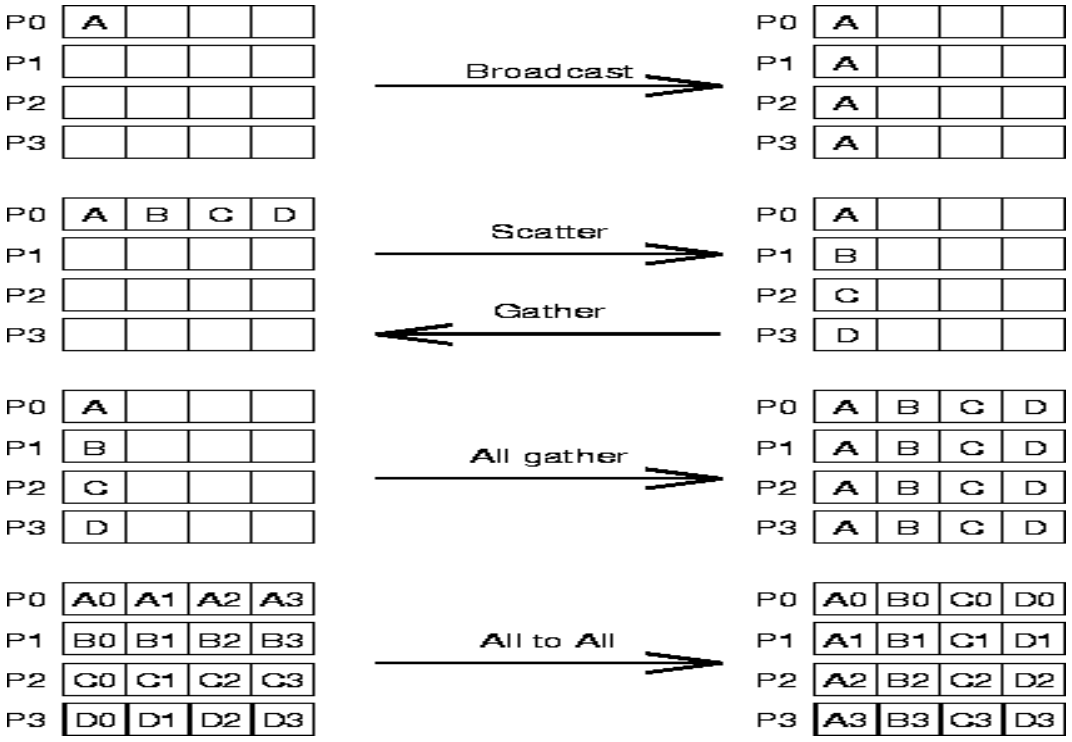
该函数的作用是从进程rank上将数据散发给所有进程（包括自己）。sendbuf和recvbuf分别是发送和接收地址。sendcount是每个进程收到的数据个数，因此sendbuf至少需要sendcount×numprocs个数据，否则一些进程将收到一些随机数据。

6.2.5 MPI并行编程

(11)
MPI_Barrier(comm)

该函数的作用是同步所有进程。该函数会阻塞通信域中所有调用了本函数的进程，直到所有的调用者都调用了它，进程中的调用才可以返回。

MPI有很多功能强大的函数，初学者只要掌握上述函数就能进行MPI并行编程。这些函数的效率也很高，其中5和6是点对点通信，7～10是组通信。注意，组通信函数是一个组的进程都需要执行的。



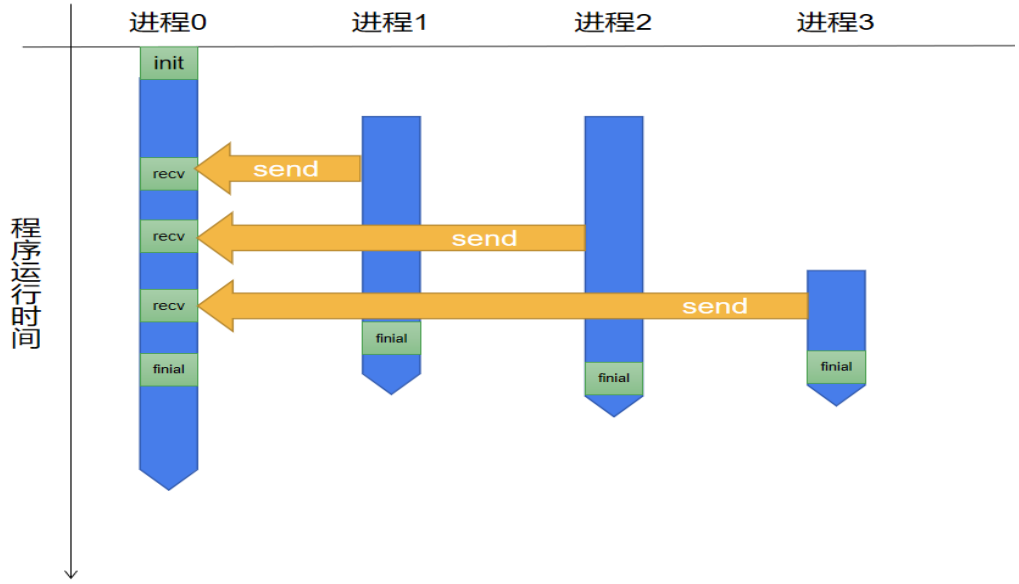
6.2.5 MPI并行编程

右图为MPI函数在多进程编程中的使用方法示例

```
1.  #include <stdio.h>
2.  #include <string.h>
3.  #include "mpi.h"
4.  void main(int argc, char* argv[])
5.  {
6.      int numprocs, myid, source;
7.      MPI_Status status;
8.      char message[100];
9.      MPI_Init(&argc, &argv);
10.     MPI_Comm_rank(MPI_COMM_WORLD, &myid);
11.     MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
12.     if (myid != 0) {    //非 0 号的其他进程发送消息
13.         strcpy(message, "Hello World!");
14.         MPI_Send(message, strlen(message) + 1, MPI_CHAR, 0, 99,
15.             MPI_COMM_WORLD);    //其中 0 代表发给 0 号进程
16.     }
17.     else {     // myid == 0，即 0 号进程接收消息
18.         for (source = 1; source < numprocs; source++) {
19.             MPI_Recv(message, 100, MPI_CHAR, source, 99,
20.                 MPI_COMM_WORLD, &status);
21.             printf("接收到第%d 号进程发送的消息：%s\n", source, message);
22.         }
23.     }
24.     MPI_Finalize();
25. } /* end main */
```

6.2.5 MPI并行编程

在初学MPI编程时Send和Recv要一一对应，收发数据的长度相同。如下图是启动4个进程时程序运行过程示意图。可以看到，当开启四进程运行时，1-3号进程发送消息，0号进程接收到消息并打印。



程序运行过程示意图

6.3 集群系统的设计和维护

6.3.1 集群系统的设计

集群的设计有较高的技术门槛，包括集群计算系统的设计和机房设计。一般由客户提出需求，并考虑以后需求变化导致系统的扩展，然后和厂商或者集成商进行测算得出集群系统的相关指标如计算、网络和存储性能等。

集群计算系统的设计

集群计算系统的设计主要涉及计算机、存储系统、网络三部分：

- 计算机：集群系统由一个个独立的商用服务器节点组成，这些节点提供了强大的计算能力和对集群的管理等功能。
- 存储系统：需要事先考查应用的特点，估算集群运行可能的带宽、延迟、存储容量等，并考虑以后的扩展，对参数进行一定的放大后根据经费情况进行存储设备选型。
- 网络：集群一般有三套网络：计算网络、管理网络、IPMI网络。

机房设计

集群机房设计需要考虑的事情很琐碎，包括机房选址、电力系统、空调、机柜、机房布局设计、承重、机房监控、消防、防尘等。机房是集群最重要的基础设施，机房设计不好，轻则可能造成系统不稳定，重则可能导致重大事故。

6.3.1 集群系统的设计

1. 集群计算系统的设计

（1）计算机

一般包含计算节点、登录节点、管理节点、调度节点、I/O节点等。

设计时通常需要考虑的：

① 节点数量。

② CPU种类、核数和主频。如Intel、AMD、ARM等。

③ 加速部件。GPU、DCU、FPGA。

④ 内存大小和访存速度。通常结合CPU内存通道数配置内存。

⑤ 刀片/机架式/高密度机型。

⑥ 网卡。IB、以太网光纤。

此外还要考虑主板（一定要支持IPMI远程管理）、电源等。

6.3.1 集群系统的设计

（2）存储系统

- NFS。配置简单，不需要额外软硬件。但是速度和容量受限
- 并行文件系统。需要配置并行文件系统（例如：开源的Lasture或者商用的GPFS）和专用存储阵列。速度快，容量高，价格贵
- 并行文件系统+通用服务器构建存储系统。价格便宜、速度快

6.3.1 集群系统的设计

（3）网络

- 计算网络
- 管理网络
- IPMI网络。
- 防火墙和路由器

6.3.1 集群系统的设计

① 双路供电，机房供电充足。
② 电力系统设计。
③ UPS和电池设计。
④ 强弱配线架设计。
⑤ 机房空调设计。
⑥ 机房布局设计。
⑦ 承重设计。
⑧ 防尘设计。
⑨ 机房监控设计。
⑩ 气体消防系统设计。

6.3.2 集群系统的维护

集群系统的维护很重要，是集群提供持续计算服务的基本保障。集群的维护工作包括对集群运行环境的维护和对集群计算系统的软硬件维护。

系统支撑软硬件的日常维护

系统支撑软硬件主要包括节点机、存储系统、网络设备、集群管理软件及数据库等基础软硬件设施。

- 节点机的维护：节点机是指计算节点、管理节点、I/O节点等集群内部所有的服务器。系统镜像备份、批量镜像还原，批量软件安装、更新和卸载等。
- 存储系统的维护：存储系统主要包括并行文件系统、磁盘阵列等。
- 网络设备的维护：网络设备维护的目标是：通过网络、安全系统管理服务，降低网络设备故障率，提高网络设备的运行性能，为集群提供稳定可靠的内部网络和安全的对外端口。
- 集群管理软件及其数据库的维护：集群管理软件包括集群管理的软件、脚本以及保存各种日志数据的数据库。

6.3.2 集群系统的维护

应用系统的日常维护

通过对应用系统的维护，分析用户不断更新的需求，分析应用系统对服务平台性能的要求，提出系统优化或者扩容解决方案，保障应用系统的处理服务性能。主要包括：

① 对集群的软件进行定期的更新、维护，对防病毒软件的防护状态与更新情况进行每天检查。

② 业务数据维护和备份。

③ 业务系统日常维护。

④ 对业务管理系统健康状态进行检查与分析。

⑤ 对系统用户信息进行维护和修改，添加系统用户，更改系统用户信息、权限，调整系统的管理人员、操作人员、监督人员以及同步数据。

机房环境的日常维护

机房环境的日常维护主要包括对电源和线路、空调、UPS等的维护。

6.4 集群系统的性能测试

6.4.1 性能评价和测量

计算机性能评价是指采用测量、模拟、分析等方法和工具对计算机系统性能进行量化分析，计算机性能测量是指采用基准测试程序包来度量计算机系统的性能。

1.性能评价的指标

只有对计算机系统的硬件、软件等各个方面进行更为准确的评价，才能全面反映计算机系统的性能。计算机速度是衡量计算机系统性能最直接和最主要的指标之一。

2.性能的描述

计算机系统的性能主要反映了一个系统的使用价值，即性能价格比。广泛的性能含义包括系统处理能力、响应速度、工作效率、可靠性、可使用性、可维护性等。

6.4.1 性能评价和测量

3.性能评价的对象

性能评价的对象是整个计算机系统，但计算机系统包括硬件、软件等复杂的系统，又与工作环境、工作方式、应用对象等有密切的关联，所以要明确地划清计算机系统的环境（边界环境），其中最主要的是工作负载。

4.性能评价的手段

性能评价的手段主要有测量技术（有实际系统存在并可从系统直接测得数据）和模型技术（只能从模型测得数据）。

5.性能的评价

一般来说，系统结构的执行速度是用户最关心的，因此产生了很多针对不同目的的基准测试程序（Benchmark），但性能评价是随着需求和软硬件的变化而发展变化的，一个能满足所有需求的性能评价方法是不存在的。

6.4.2 Linpack 测试

Linpack全称为Linear Equations Package，是一种较为常用的计算机系统性能测试线性方程程序包，

其中包括求解稠密矩阵运算、带状的线性方程、求解最小平方问题以及其他各种矩阵运算。

HPL（High Performance Linpack，高性能Linpack）

针对现代并行计算机提出的测试方式，

用户在不修改任何测试程序的基础上，

可以通过调节问题规模大小N（矩阵大小）、进程数等测试参数，使用各种优化方法来执行该测试程序，以获取最佳的性能。

HPL测试结果是TOP 500排名的重要依据。

6.5 高性能集群计算机系统实例

上海大学高性能计算中心成立于2007年9月。2000年自主研制的第一代集群式高性能计算机“自强2000”共有218个CPU；2004年上海大学与HP合作研制建设完成的第二代集群式高性能计算机“自强3000”共有192个节点机；2013年已经建成的第三代集群式高性能计算机“自强4000”共有162个节点机。

“自强2000”
集群机

“自强3000”
集群机

“自强4000”
集群机

6.6 网格

6.6.1 网格概述

网格（Grid）技术，是20世纪90年代中期随着计算机网络技术和分布式计算技术的不断发展而诞生的一种全新技术。它以高速网络为依托，借助于一套完善的网格中间件的支持，将分布于网络上的各种资源加以整合，为使用者提供一套完善、使用方便的支持环境。

在此基础上，网格使用者可以方便地对网格中的各种资源加以动态的有效利用，解决各个不同领域中的科学、工程、商业等问题。

自从网格技术出现以后，网格相关的各种研究在全世界范围内得到了广泛的重视。网格技术已经成为对国家科技进步、国民经济发展、综合国力提高和国家安全具有重要意义的关键技术。在这种形势下，我国也认识到了网格的巨大作用。863高科技计划启动了网格专项研究，在网格节点建设、网格应用等方面开展研究。同时，国家自然科学基金等国家、社会基金也开始对网格的相关研究加以支持

6.6.2 网格技术简介

网格技术是一种通过高速网络来统一管理各类不同物理位置的资源（超级计算机、大型数据库、存储设备、各种仪器设备、知识库等）并运用系统软件、工具和应用环境使其成为互相协调的先进计算设施的技术。

①网格体系结构研究

网格体系结构研究是研究网格技术和构建网格系统的关键。网格体系结构由三部分构成：网格分层、各层所提供的网格服务和为了提供这些网格服务所必须遵循的网格协议。

②网格资源访问规范

网格系统是建立在各种各样不同类型、不同平台、不同用途的资源基础之上的，这些资源需要以不同的手段、遵循不同的协议来访问。

③网格资源索引机制

网格资源索引系统为网格的用户提供资源索引服务，是网格能够作为一个整体加以运行的关键，资源的分类与描述是资源索引的基础。

6.6.2 网格技术简介

④网格数据管理规范

在网格系统中存在大量的数据服务，网格系统需要提供可靠的、高效的数据管理机制，在网格用户和网格资源之间提供一个可靠、高效的数据通道。

⑤网格服务质量

描述网格服务质量就是对一个资源进行评价与计量，包括资源的质量如何、资源的数量如何体现等，建立一个通用的网格资源评价与计量体系是很困难的。

⑥网格安全与网格用户管理机制

网格安全包括数据存储和传输的安全、资源访问的安全、各种应用和相关数据的安全、用户信息的安全等；网格用户管理技术要实现网格用户的认证与授权等功能。

6.6.2 网格技术简介

⑦网格应用支持工具与开发环境

网格应用支持工具与开发环境为网格系统的用户提供一套能够比较简单、有效地使用网格系统的各种资源来完成应用开发的工具与编程环境。

⑧应用网格中的理论、模型、方法和算法研究

以网格的方式来解决应用系统的问题，必须解决应用过程中需要面对的各种理论、模型、方法和算法问题，必须研究在网格条件下用于解决资源优化和安全保证等问题的各种理论和模型，并在此基础上研究新的方法和算法。

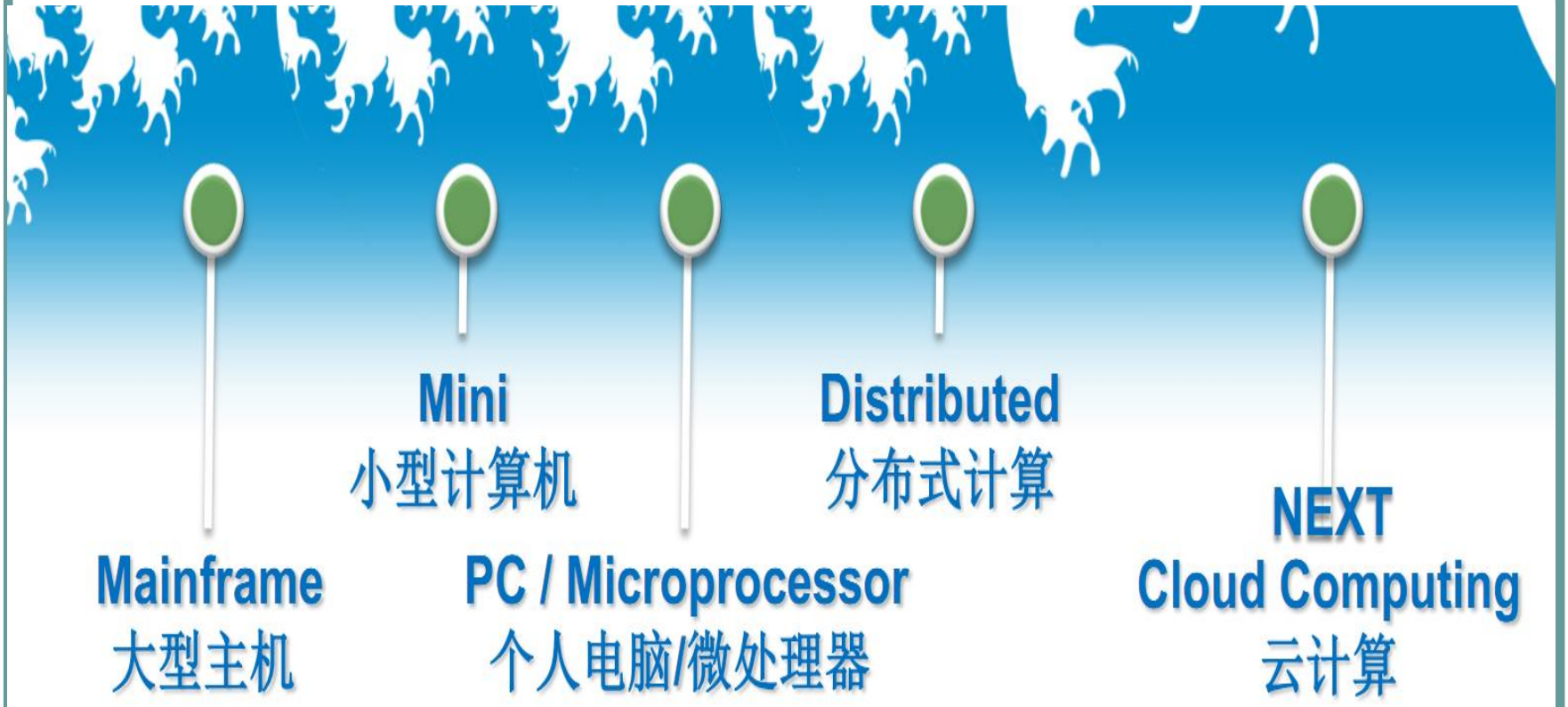
6.7 云计算

“云计算”成为计算机技术发展中最响亮的名词之一已经是不争的事实，云计算已经到来，并正在不断地改变大家的生活。

云计算种基于互联网的计算模式，是一种来自网络的资源，使用者可以随时按需申请并获取“云”上的资源，并且可以按需扩展，按使用量付费即可。

- 云计算
  - **Cloud computing: Systems, Networking, and Frameworks**
- 关键技术　计算机系统
  - 低功耗的CPU来做云是非常有前景的，
    - 尤其对于数据密集型的应用
  - 分层存储机制 大数据存储
  - 虚拟化技术
  - 并行计算技术
  - 网络协议栈调优
  - 信息安全
  - ……

信息技术的浪潮


Mainframe
大型主机
Mini
小型计算机
PC / Microprocessor
个人电脑/微处理器
Distributed
分布式计算
NEXT
Cloud Computing
云计算

6.7.1 云计算概述

云计算有许多定义，其中中国云计算专家委员会对云计算做了定义：云计算是一种基于互联网的计算方式，通过这种方式，共享的软硬件资源和信息可以按需提供计算机和其他设备。

在云计算模式中，用户所需的应用程序并不运行在用户的个人电脑、手机等终端设备上，而是运行在互联网上大规模的服务器集群中。用户所处理的数据也并不存储在本地，而是保存在互联网上的数据中心里。

云计算得到广泛应用，与传统的网络应用模式相比，具有如下特点：

① 超大规模及廉价性：各互联网巨头的“云”均有数以万计的服务器，并通过高效的建设、管理方法降低“云”的成本；

② 虚拟化：“云”将多种资源进行虚拟化和池化共享；

③ 高可靠性：云计算使用数据多副本容错、计算节点同构可互换等措施来保障服务的高可靠性；

④ 通用性：同一个“云”可以同时支撑不同的应用运行；

⑤ 高扩展性：“云”的规模可以动态伸缩；

⑥ 按需服务：云计算平台能够根据用户的需求快速部署软件，配备计算能力及资源。

6.7.1 云计算概述

云计算的系统结构分为四层：物理资源层、资源池层、管理中间件层和SOA（面向服务的系统结构）构建层。

云计算服务分为以下三类：

（1）IaaS（Infrastructure as a Service，基础设施即服务）：IaaS是把硬件设备等资源封装成服务通过网络对外提供，并根据用户对资源的实际使用量或占用量进行计费的一种服务模式。

（2）PaaS（Platform as a Service，平台即服务）：PaaS对资源的抽象层次更进了一步，它提供用户应用程序的运行环境。

（3）SaaS（Software as a Service，软件即服务）：SaaS平台供应商将某些特定应用软件功能封装成服务，统一部署在自己的服务器上，客户可以根据实际需求，通过互联网向其订购所需的应用软件服务，按订购的服务数量和时长向其支付费用。

云计算

- 云计算体系结构研究
  - 云层次栈：云计算的3种服务归属于不同的软件架构层
  - 云应用层
  - 云软件环境层
  - 云软件基础设施层
- 云计算栈
    - 基础设施即服务层(IaaS)
    - 平台即服务层(PaaS)
    - 软件即服务层（SaaS)
- 云计算关键技术研究

IaaS

- IaaS 技术
  - 虚拟化 Virtualization
    - Server Virtualization
    - Storage Virtualization
    - Network Virtualization
- IaaS 服务
  - 资源管理Resource Management Interface
  - 系统监控 System Monitoring Interface

6.7.1 云计算概述

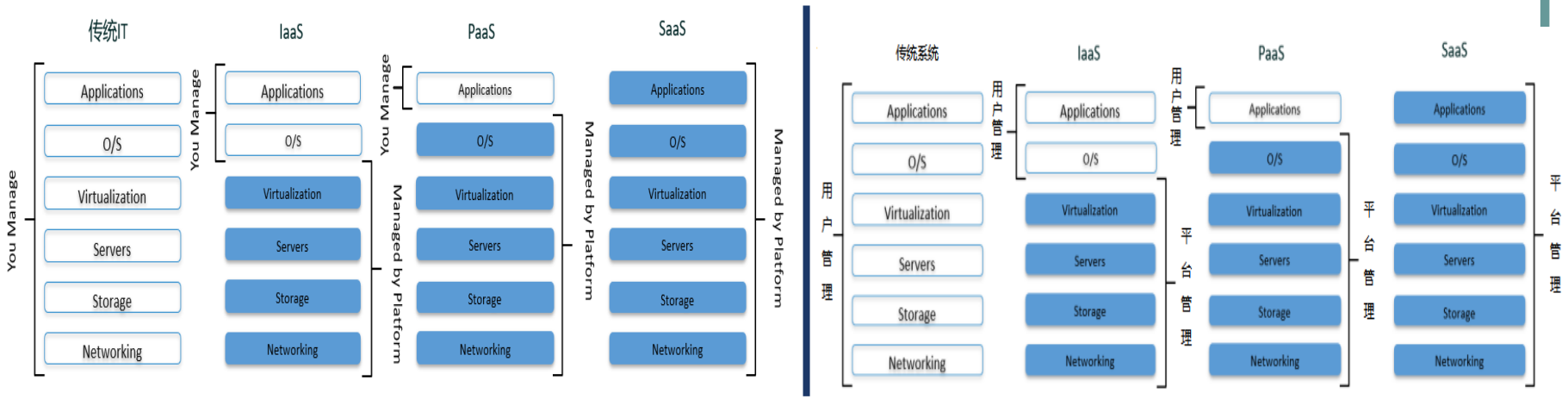


图6-11 云计算服务模式图

图6-11显示了云计算服务模式。传统模式下，用户需从购买硬件开始构建系统；IaaS、PaaS、SaaS分别从硬件层、操作系统层、应用层提供不同层次的云服务。这三种云计算服务有时称为云计算堆栈，因为它们像堆栈一样，位于彼此之上。

6.7.2 云计算的关键技术

云计算得到迅速发展得益于一些关键技术的发展，具体如下：

①虚拟化技术

重要的核心技术之一，它为云计算服务提供基础架构层面的支撑；
虚拟化技术是利用软件或者固件管理程序构成虚拟化层，把物理资源映射为虚拟资源，以虚拟资源为用户提供服务的计算形式，旨在聚合计算资源并在此基础上合理调配计算机资源，使其更高效地提供服务。

云计算平台常用的虚拟化技术

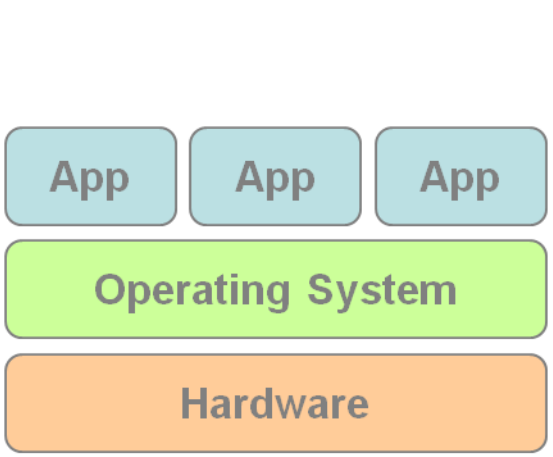
有虚拟机（Virtual Machine）和容器（Container）两种：

- 虚拟机技术：指通过软件模拟的具有完整硬件系统功能的运行在一个完全隔离环境中的完整计算机系统；
- 容器技术：是更轻量级的虚拟化技术，其原理是在原有系统基础上实现进程隔离，目的是为进程提供独立的运行环境，使其无法访问容器外的其他资源。

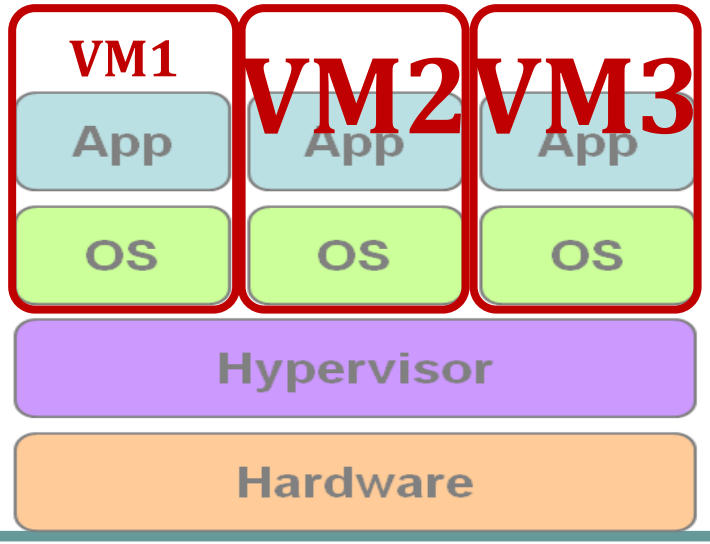
容器和虚拟机之间的主要区别在于虚拟化层的位置和操作系统资源的使用方式。

虚拟化技术

- Virtualization is an abstraction of logical resources away from underlying physical resources.
  - Virtualization technique shift OS onto hypervisor.
  - Multiple OS share the physical hardware and provide different services.
  - Improve utilization, availability, security and convenience.



Traditional Stack



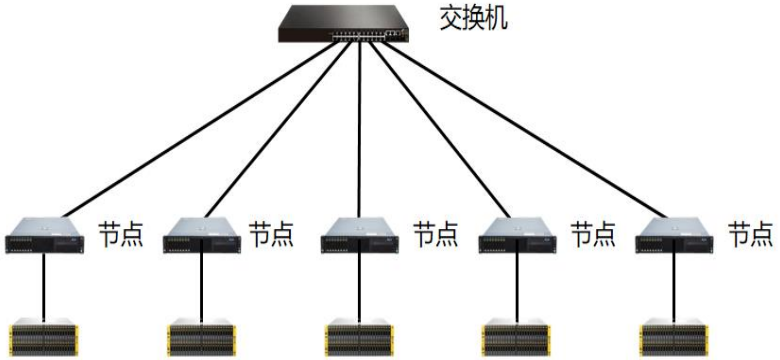
Virtualized Stack

6.7.2 云计算的关键技术

②分布式存储技术

云计算采用分布式存储技术，将数据存储在大量的计算节点中。这种模式不仅摆脱了硬件设备的限制，同时扩展性更好，能够快速响应用户需求的变化。

分布式存储与传统的集中式存储不同。集中式存储系统是在一套由一个或多个节点组成的存储系统中存储所有数据，系统所有的功能均由这些节点集中处理。集中式存储强调将存储集中部署和集中管理。集中式存储虽然有技术成熟、可用性高等优点，但面对海量数据，其缺点也越来越明显，如扩展性差、成本高等。



分布式存储图

6.7.2 云计算的关键技术

③分布式并行编程模式

在云计算项目中，分布式并行编程模式被广泛采用。分布式并行编程模式创立的初衷是更高效地利用软、硬件资源，让用户快速、便捷地使用应用或服务。在分布式并行编程模式中，后台复杂的任务处理和资源调度对用户来说是透明的，这大大提升了用户体验。

分布式计算框架MapReduce是当前云计算的主流并行编程模式之一，用于大规模数据集的数据并行计算。

MapReduce的主要思想就是分而治之，把一个复杂的任务（大量数据）划分为大量简单的任务（大量数据块），然后将这些任务尽量调度到存储了该数据块的主机上进行并行处理。MapReduce框架将任务自动分成大量子任务，通过Map和Reduce两步实现任务在大规模计算节点中的自动分配。

6.7.2 云计算的关键技术

④大规模数据管理

云计算不仅要保证数据的存储和访问，还要能够对海量数据进行特定的检索和分析。由于云计算需要对海量的分布式数据进行处理、分析，因此数据管理技术必须能够高效地管理大量的数据。

⑤云计算平台管理、调度和计费

云计算系统的平台管理技术，需要高效调度大量服务器等资源，使其更好地协同工作。同时，云计算平台管理方案要更多地考虑到定制化需求以满足不同场景的应用需求，也需要一个高效灵活的收费系统支持各种动态收费业务。

6.7.2 云计算的关键技术

⑥信息安全

在云计算体系中，安全涉及很多层面，包括网络安全、服务器安全、软件安全、系统安全等。云安全产业的发展，将使传统安全技术进入一个新的阶段。

⑦绿色节能技术

云计算平台机器数量庞大，能耗极高。能耗也是云计算厂商的主要成本之一。云计算的优势在于能够利用虚拟化等技术提高资源的利用率、减少物理服务器的数量，从而达到大大降低运营成本的目的。

小结

- 云计算体系结构研究
- 云计算关键技术研究
  - 虚拟化技术
  - 数据存储技术
  - 资源管理技术
  - 能耗管理技术
  - 云检测技术

6.7.3 Openstack开源虚拟化平台

OpenStack既是一个社区，也是一个项目和一个开源软件。OpenStack是一个可以管理拥有大量计算、存储和网络等资源的数据中心的云计算平台操作系统，通过仪表板为管理员提供所有的管理控制，通过Web界面为用户提供云计算资源等服务，开发者可以通过API访问云计算资源和创建云应用，可以为公有云、私有云等不同规模的云提供可扩展的、灵活的云计算。

OpenStack的主要服务：

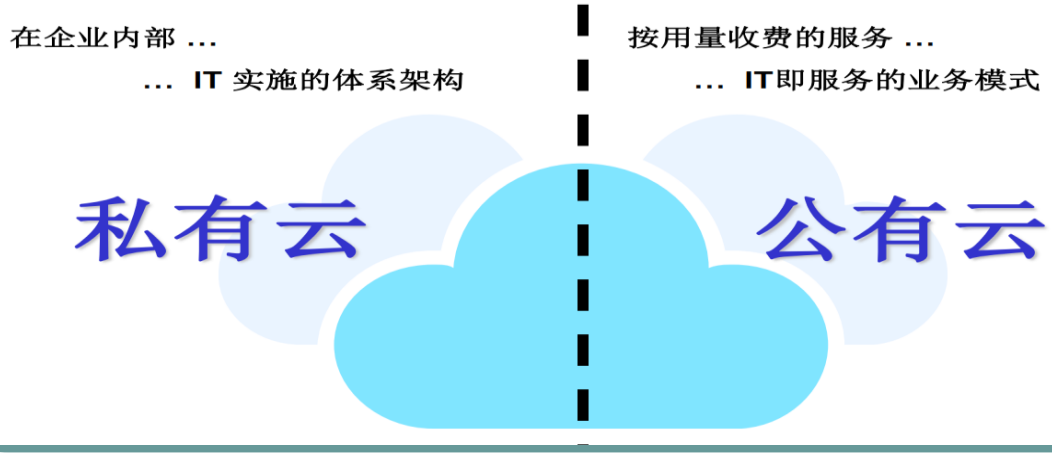
(1)计算服务Nova　(2)对象存储服务Swift　(3)镜像服务Glance

(4)认证服务Keystone　(5)网络服务Neutron　(6)块存储服务Cinder

(7)仪表盘Horizon

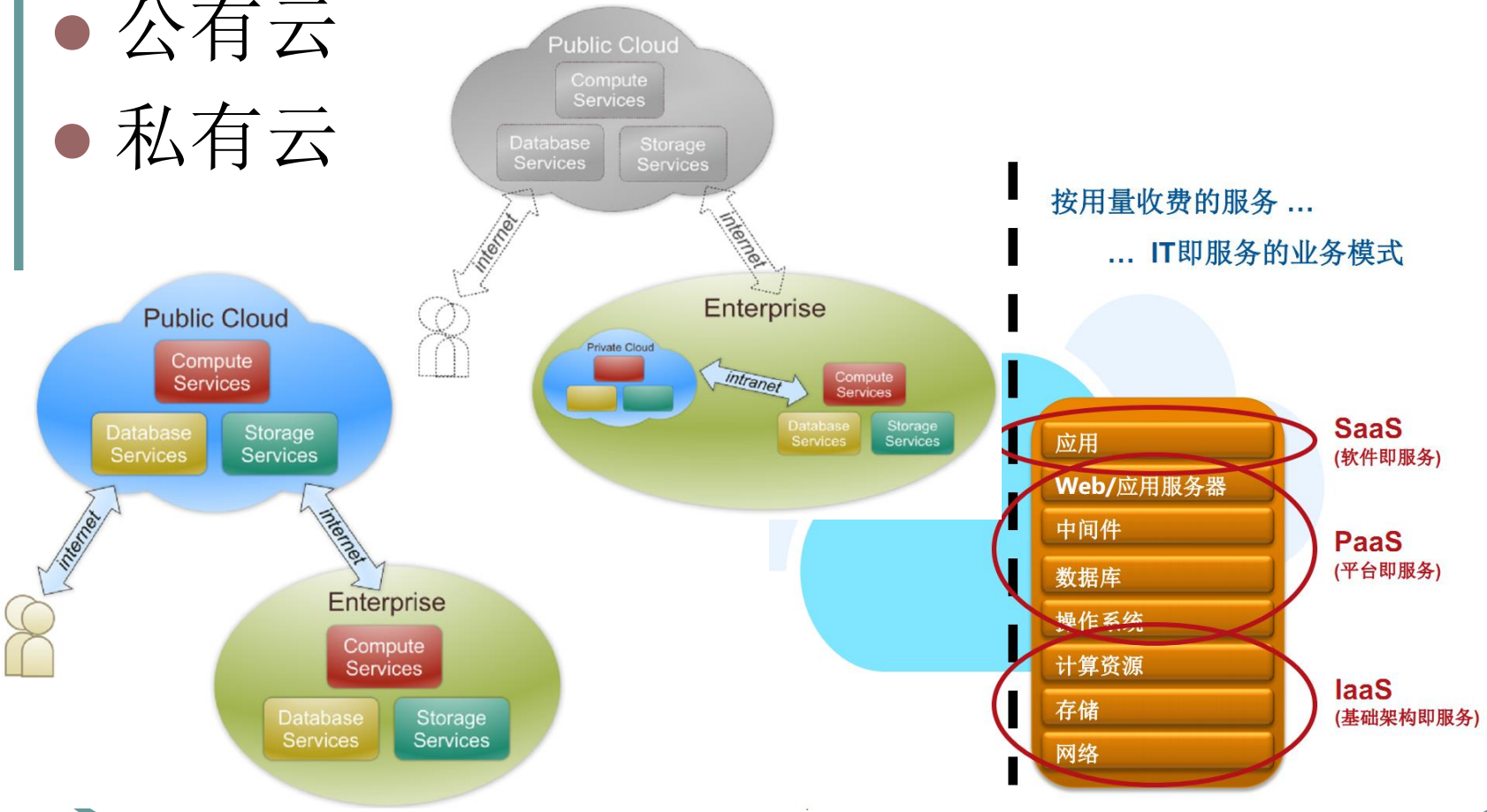
开发部署模型

- 4种云应用部署模式:
  - 公有云 Public Cloud
  - 私有云 Private Cloud
  - 社区云 Community Cloud
  - 混合云 Hvbrid Cloud



开发部署模型

- 公有云
- 私有云



6.8 大数据

数据	数据是对客观事物的符号表示，是用于表示客观事物的素材，如数字、字母、图形符号等。在计算机科学中，数据是指能输入到计算机并被计算机程序处理的所有符号的总称。
大数据	大数据本身是一个抽象的概念，通常指通过传统的数据库软件无法在可接受的范围内进行获取、存储、管理和计算的巨大的数据集，这些数据来源于人们社会生活的方方面面，例 如互联网、传统行业、音/视频、移动设备、监控设备等。

6.8 大数据

大数据具有如下的4V特征：

Volume 大量	大数据的特征首先就体现为“大”。随着信息技术的高速发展，数据开始爆发性增长。社交网络（抖音、微博、QQ、微信等）、工业控制等各种途径产生的音/视频等，都成为数据的来源。存储数据量从过去的GB级别达到TB乃至PB级别。
Variety 多样	大数据的数据类型繁多,处理非结构化数据的难度和复杂度远远高于结构化数据。大数据处理必须面对各种复杂的数据类型，从各种类型的数据中快速获得有价值的信息。
Velocity 高速	大数据的处理速度快。在数据处理速度上有个著名的“1秒定律”，即要在秒级时间内给出分析结果，否则数据就会失去价值，因此大数据对处理速度有非常严格的要求，服务器中大量的资源都用于处理和计算数据，很多平台都需要做到实时分析。
Value 价值	价值密度低。大数据需要在大量的各种类型数据中挖掘有价值的信息，相对于传统数据库价值密度低、处理难度高。但是，通过分布式存储和并行计算等技术可以大大提高效率，获得有价值的信息。

6.8 大数据

大数据技术突破了传统数据库的限制，能在超大规模的多机系统和分布式存储平台上进行数据处理。但是，大数据技术仍然是将社会生活中的数据传输到大数据计算平台进行集中处理，进而获得需要的信息的。

边缘计算与云计算形成互补关系。云计算聚焦非实时、长周期数据的大数据分析，能够为业务决策提供依据；边缘计算则聚焦实时、短周期数据的分析，能更好地支撑本地业务的实时智能化处理与执行；云计算具有远远超过边缘端的更强的计算能力和存储能力，数据首先经过边缘端的处理，然后传输到云计算平台进行更高级的分析，以获取有价值的信息。

大数据处理3种模式

- 离线计算
- 在线处理
- 流计算
- Hadoop使用较广泛的离线计算框架
- Spark利用内存处理提升系统性能
  - 一种物理网络社会大数据的数据服务框架
  - High-Speed Big Data Analysis Framework