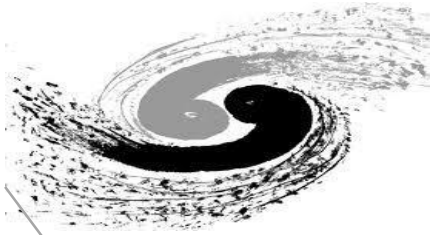


# 高能物理离线计算

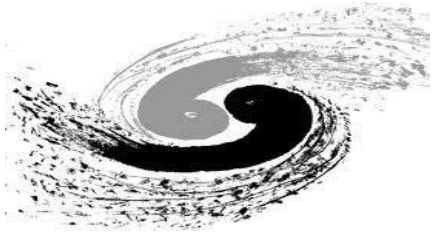
第一届（2020年）高能物理计算暑期学校

石 京 燕

高能所计算中心

shijy@ihep.ac.cn

# 提纲

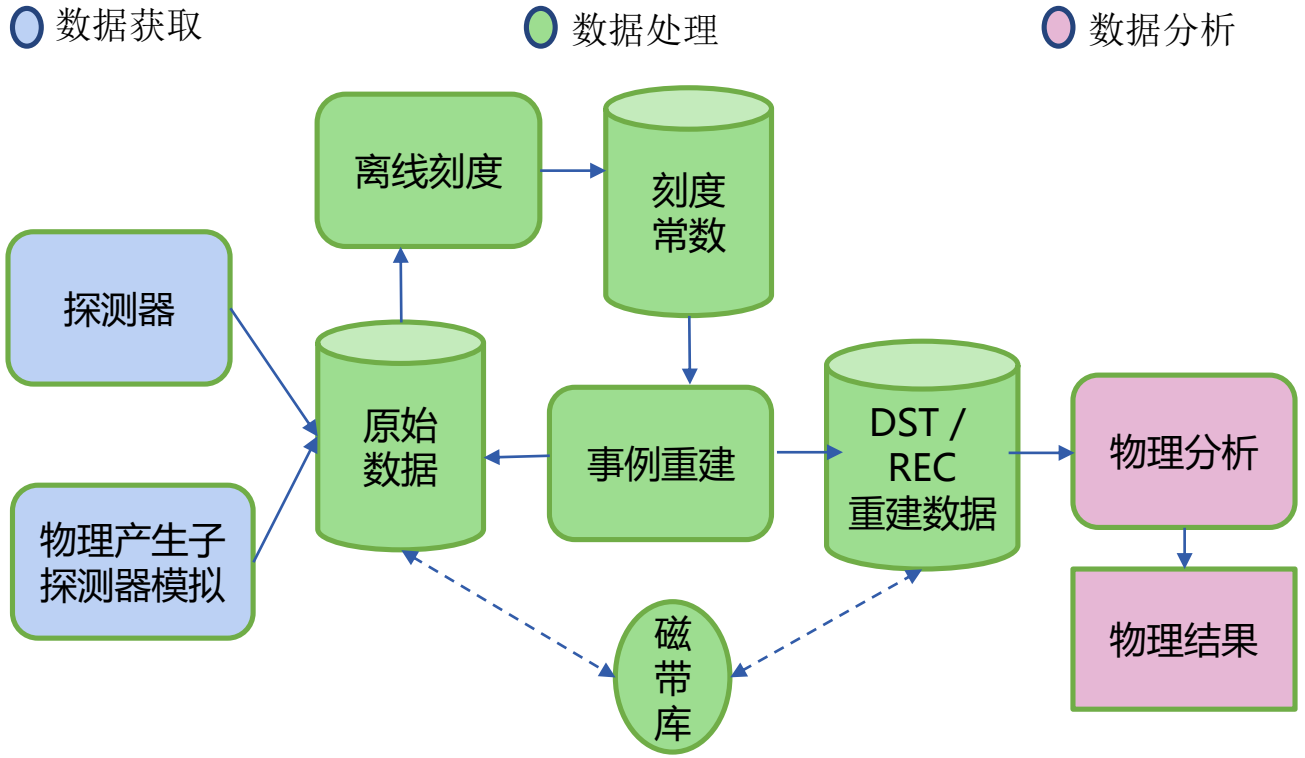


1. 高能物理离线数据处理过程
2. 高能物理离线数据处理特点
3. 高通量计算与高性能计算
4. 网格计算与分布式计算
5. 高能所计算平台
6. 总结

# 高能物理数据处理过程

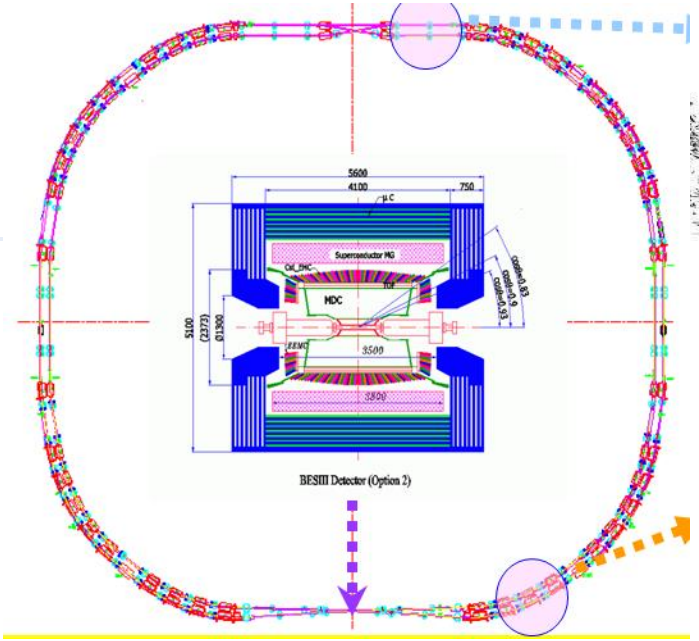
在线数据
实时获取

传到离线，
长期保存

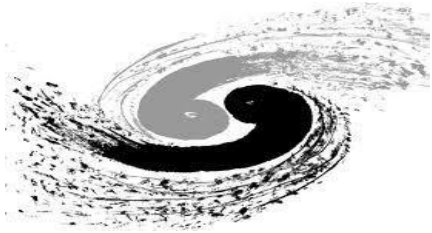


# 高能物理计算的不同阶段

- 实验的设计和建造阶段
  - 可行性研究，
  - 实验装置的设计与优化
  - 物理模拟
- 实验运行
  - 数据采集和刻度
  - 事例选择和事例重建
- 物理分析
  - 数据分析、计算与评价
  - 物理结果的确定
  - 效率的确定
  - 信号和本底（信噪比）的测量
  - 误差的估算与修正



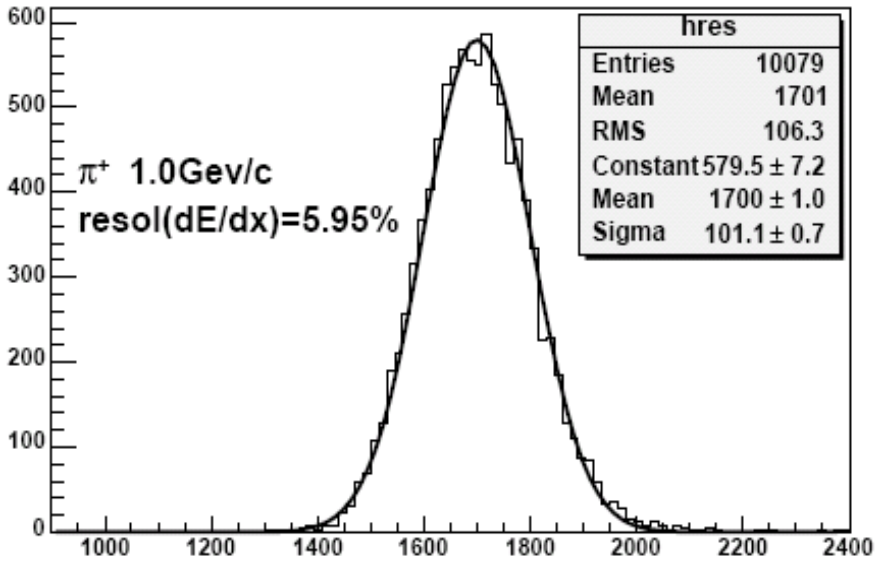

# 提纲

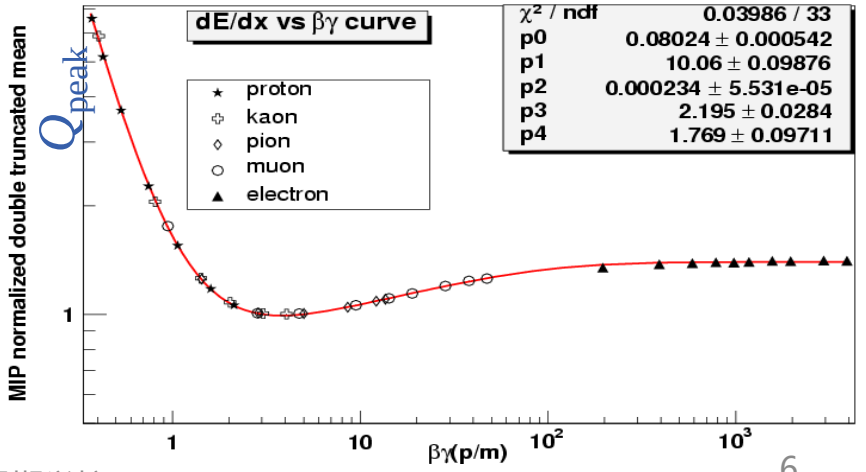


1. 高能物理离线数据处理过程
2. **高能物理离线数据处理特点**
3. 高通量计算与高性能计算
4. 网格计算与分布式计算
5. 高能所计算平台
6. 总结

# 高能物理离线计算分类及特点

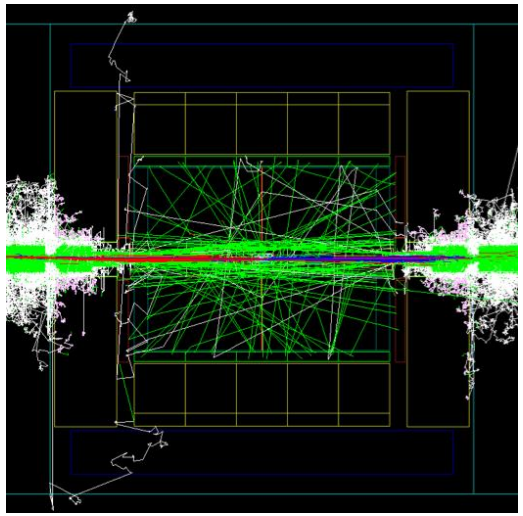
- 特点：
  - 实时性不高，是实验的一项长期工作
  - 计算过程被不断调整，优化
  - 分析计算得到的数据及结果可重现
- 过程：
  - 刻度（Calibration）：消除实验外部条件对电子学信号与物理测量之间转换关系的影响
  - 校准（Alignment）：定位探测器精度
  - 事例模拟
    - 模拟与真实数据最相近的人造事例
    - 需要背景噪声研究，修正，错误估算
  - 事例重建：
    - 重建粒子轨迹和顶点
    - 标注粒子类型和衰变
    - 加入物理约束（能量动量保留）
  - 物理分析
    - 从背景数据中找到物理信号
    - 计算截面、分支比等
    - 需要大量培训，具备高超物理数据分析技术

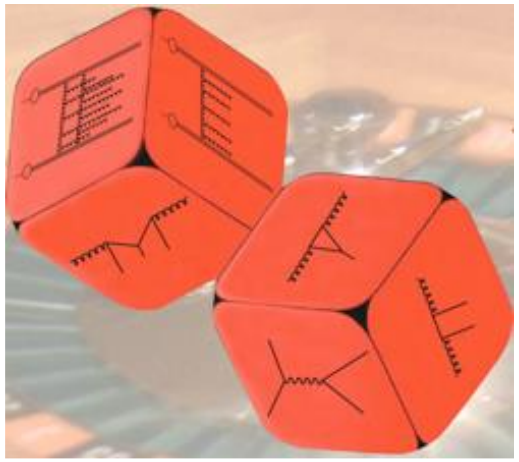


# 物理模拟

- 利用计算机来模拟物理过程-事例产生
  - 物理过程是随机过程
    - 事例的产生
    - 粒子在介质中的输运
  - 信号：需要研究的物理过程
  - 本底：除信号以外的其他物理过程以及探测器的噪声

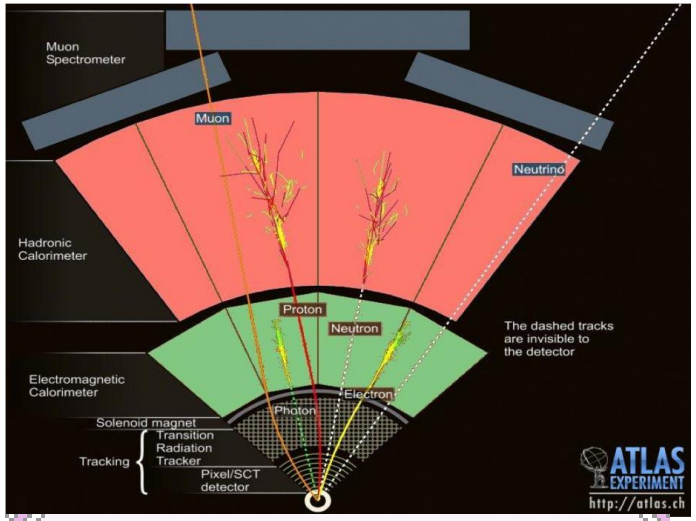


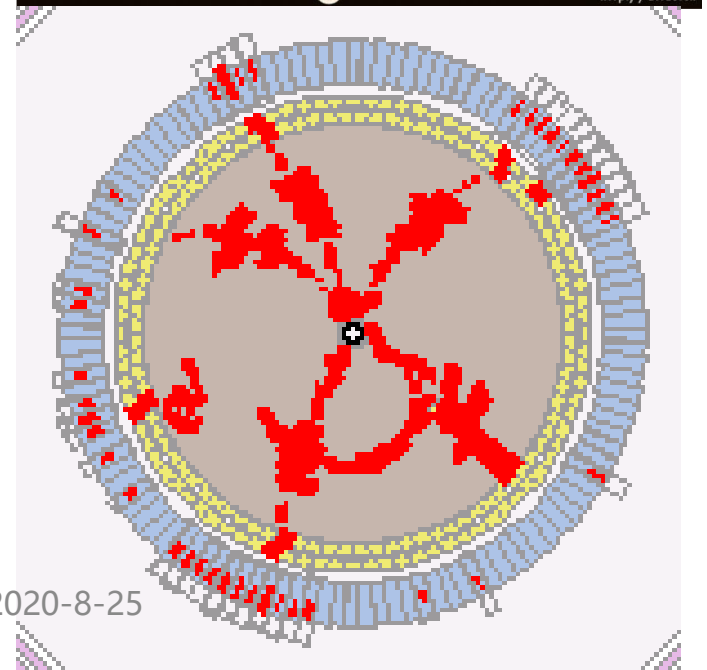


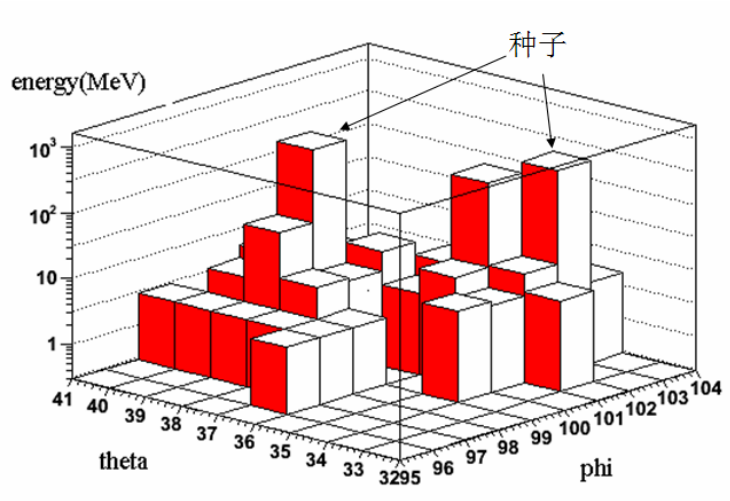

- 重要的模拟软件：GEANT4
- 基于‘已知‘物理理论模拟粒子在介质中的传输过程
  - 粒子衰变、光电效应、康普顿散射、粒子对产生、粒子输运过程中的能量损失、韧致辐射、强子相互作用

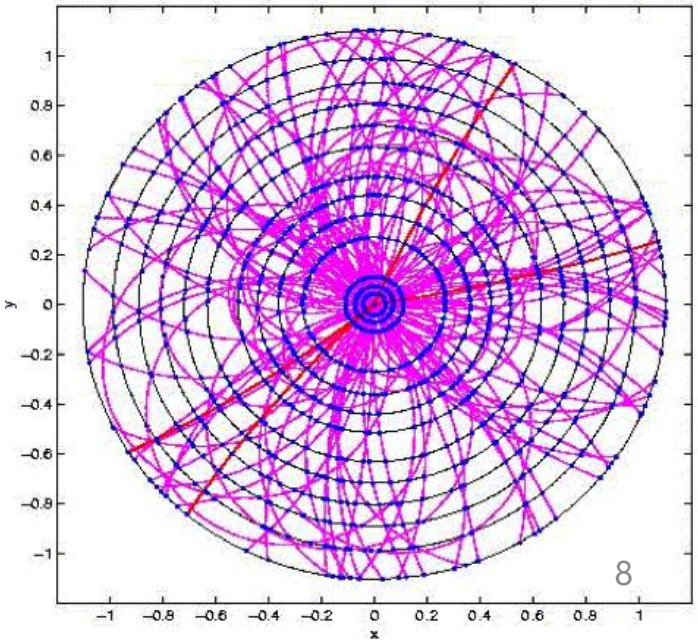
# 事例重建

- 径迹重建
- 簇射簇重建
- 粒子鉴别
- 顶点重建
- 动力学重建





# 物理分析

- 物理动机
- MC模拟
- 程序编写
- 程序调试
- 初步分析
- 深入分析
- 输入输出检查
- 系统误差

## 初步想法

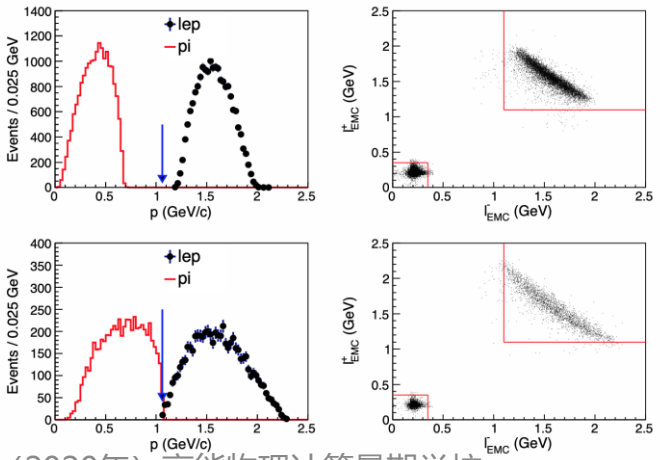
**B. For $\psi' \to \pi^+\pi^- J/\psi, J/\psi \to l^+l^-$ process**

- Track level cuts
  - $|V_z| < 10\text{cm}$
  - $|V_r| < 1\text{cm}$
  - $|\cos\theta| < 0.80$
  - $|\vec{p}| < 2.0\text{GeV}/c$
- nGood >= 4
- The charged tracks with $|\vec{p}| < 0.45$ GeV/c are assumed to be pion, select the $\pi^+\pi^-$ pair candidates by minimizing $|M^{rec.}_{\pi^+\pi^-} - M_{J/\psi}|$
- $\cos\theta_{\pi^+\pi^-} < 0.95$
- $3.05\text{GeV}/c^2 \le M^{rec.}_{\pi^+\pi^-} \le 3.15\text{GeV}/c^2$
- Take the two fastest positive and negative tracks as lepton candidates, identify the $e/\mu$ pair
  - $\mu^+\mu^-$:$[E/p]^+ < 0.26$ and $[E/p]^- < 0.26$
  - $e^+e^-$:$[E/p]^+ > 0.80$ or $[E/p]^- > 0.80$ or $\sqrt{([E/p]^+ - 1)^2 + ([E/p]^- - 1)^2} < 0.4$
- $\cos\theta_{l^+l^-} < -0.95$ in lab frame.
- $2.7\text{GeV}/c^2 < m_{l^+l^-} < 3.2\text{GeV}/c^2$ for $\pi^+\pi^-e^+e^-$ channel and $3.0\text{GeV}/c^2 < m_{l^+l^-} < 3.2\text{GeV}/c^2$ for $\pi^+\pi^-\mu^+\mu^-$ channel

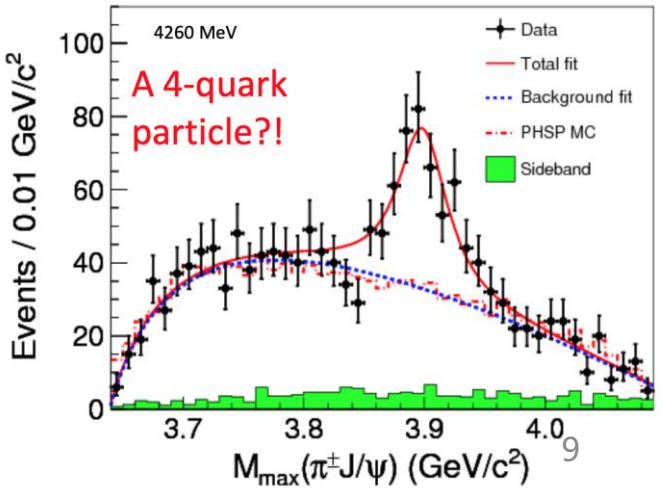
## 程序实现

```
// With momentum method calculate the invariant mass of Jpsi
// actually we use the recoil mass
HepLorentzVector m_lv_recoil, m_lv_jpsi;
m_lv_recoil = m_lv_lab - m_lv_pionp - m_lv_pionm;
m_lv_jpsi = m_lv_lepp + m_lv_lepm;

m_mass_twopi = (m_lv_pionp + m_lv_pionm).m();
m_mass_recoil = m_lv_recoil.m();
m_mass_jpsi = m_lv_jpsi.m();

// Jpsi mass cut
if( m_mass_recoil < 3.05 || m_mass_recoil > 3.15 ) return sc;
if( m_mass_jpsi < 3.0 || m_mass_jpsi > 3.2 ) return sc;
m_cout_recoil ++;

HepLorentzVector m_ttm(m_lv_jpsi + m_lv_pionp + m_lv_pionm);
if(m_ttm.m()>4 || m_ttm.m()<3) return sc;

// dangle between pions, suppress gamma convertion
m_pipi_dang = m_lv_pionp.vect().cosTheta(m_lv_pionm.vect());

m_mom_pionp = m_lv_pionp.vect().mag();
m_mom_pionm = m_lv_pionm.vect().mag();
m_mom_lepp = m_lv_lepp.vect().mag();
m_mom_lepm = m_lv_lepm.vect().mag();
m_trans_ratio_lepp = m_lv_lepp.vect().perp()/m_lv_lepp.vect().mag();
m_trans_ratio_lepm = m_lv_lepm.vect().perp()/m_lv_lepm.vect().mag();
m_trans_ratio_pionp = m_lv_pionp.vect().perp()/m_lv_pionp.vect().mag();
m_trans_ratio_pionm = m_lv_pionm.vect().perp()/m_lv_pionm.vect().mag();

Hep3Vector m_boost_jpsi(m_lv_recoil.boostVector());
HepLorentzVector m_lv_cms_lepp(boostOf(m_lv_lepp,-m_boost_jpsi));
HepLorentzVector m_lv_cms_lepm(boostOf(m_lv_lepm,-m_boost_jpsi));
m_cms_lepm = m_lv_cms_lepm.vect().mag();
m_cms_lepp = m_lv_cms_lepp.vect().mag();
log << MSG::DEBUG << "jpsi four momentum in cms " << m_lv_cms_lepp + m_lv_cms_lepm <<endreq;

m_inv_mass = m_ttm.m();
m_tot_e = m_ttm.e();
m_tot_px = m_ttm.px();
```

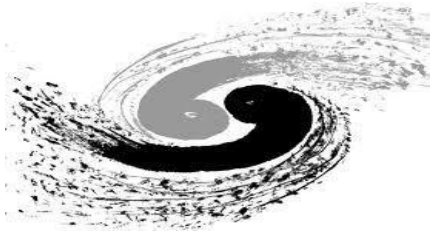
## 初步分析




## 深入分析



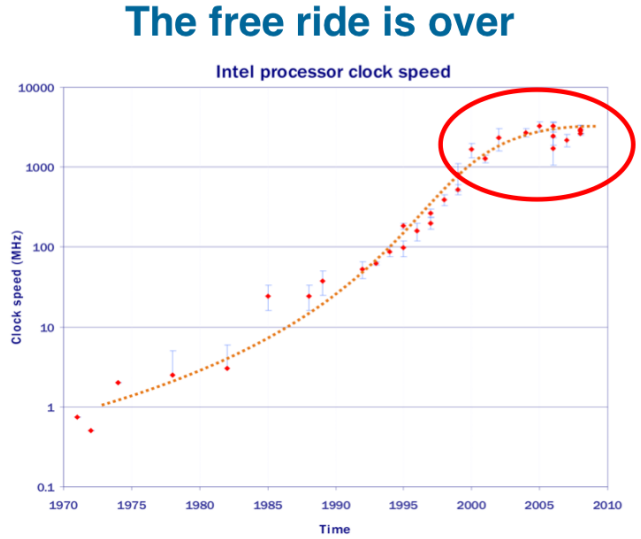
# 提纲

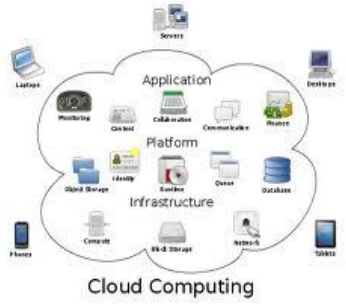


1. 高能物理离线数据处理过程
2. 高能物理离线数据处理特点
3. **高通量计算与高性能计算**
4. 网格计算与分布式计算
5. 高能所计算平台
6. 总结

# 计算技术的演变

- 过去40年：
  - 芯片性能增长一直符合摩尔定律：每个芯片的晶体管数每两年增加一倍
  - 但是计算能力并没有增长这么快：缓存，内存，网络...
  - 新问题：更高的能耗和制冷
- 科学计算的技术演变
  - 20年前： 本地计算集群 （今天仍是主流!）
  - 15年前：广域网上的分布式计算
  - 不断发展出的新技术：虚拟化计算，志愿计算，GPU加速，AI数据分析...

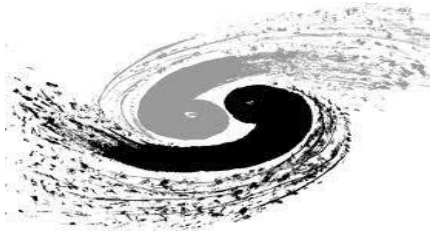



# 高通量计算 vs. 高性能计算

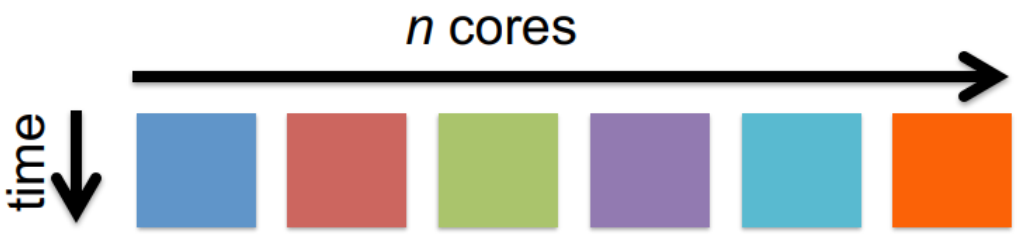
- 高通量计算：**High Throughput Computing**
  - 同时在多个处理器上运行大量相互独立的软件实例（作业）
  - 长时间（一年多年）
  - 大规模稳定的计算 -- 快速吞吐
- 高通量计算：**High Performance Computing**
  - 同时在多个处理器上运行大并行软件
  - 作业数量少，但是作业对CPU能力要求高
  - 短时间需求

# 高通量计算 vs. 高性能计算

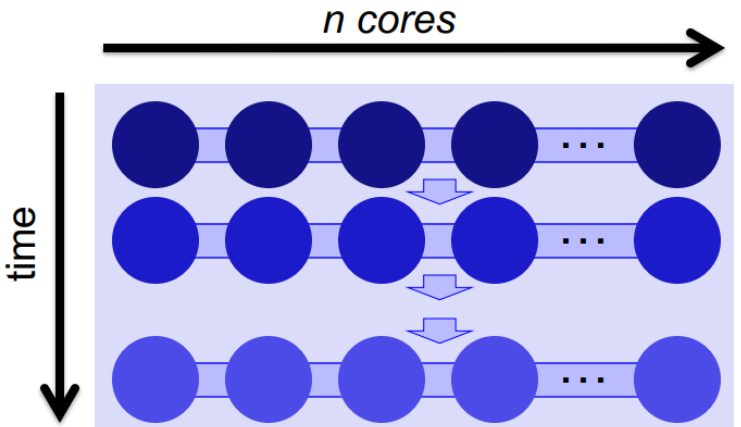


- HTC：在一个较长时间提供大量计算资源。HTC不关注每秒完成的操作，而是每月、每年完成的操作。其目标是在一个很长的周期内完成的尽可能多的作业量，而非可以多快时间内完成时间单个作业。

| | 作业槽数/作业量 | 单作业进程并行度 | 作业槽异构/同构 | 文件系统 | 节点网络速度 |
|---|---|---|---|---|---|
| HTC | 多 | 小 | 异构 | 共享文件系统/数据传输 | 不要求 |
| HPC | 少 | 多 | 同构 | 共享文件系统 | 快 |




单核、串行、无相关性

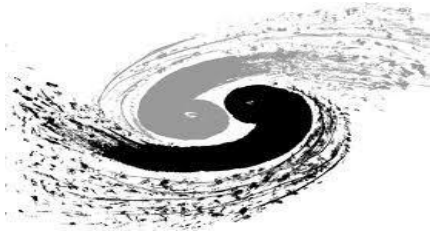



多核、并行、关联度高

# HTC的作业调度与资源管理

- HTCondor -- 一个示例
  - 买家/商家：购买/销售手机

# HTCondor的作业调度



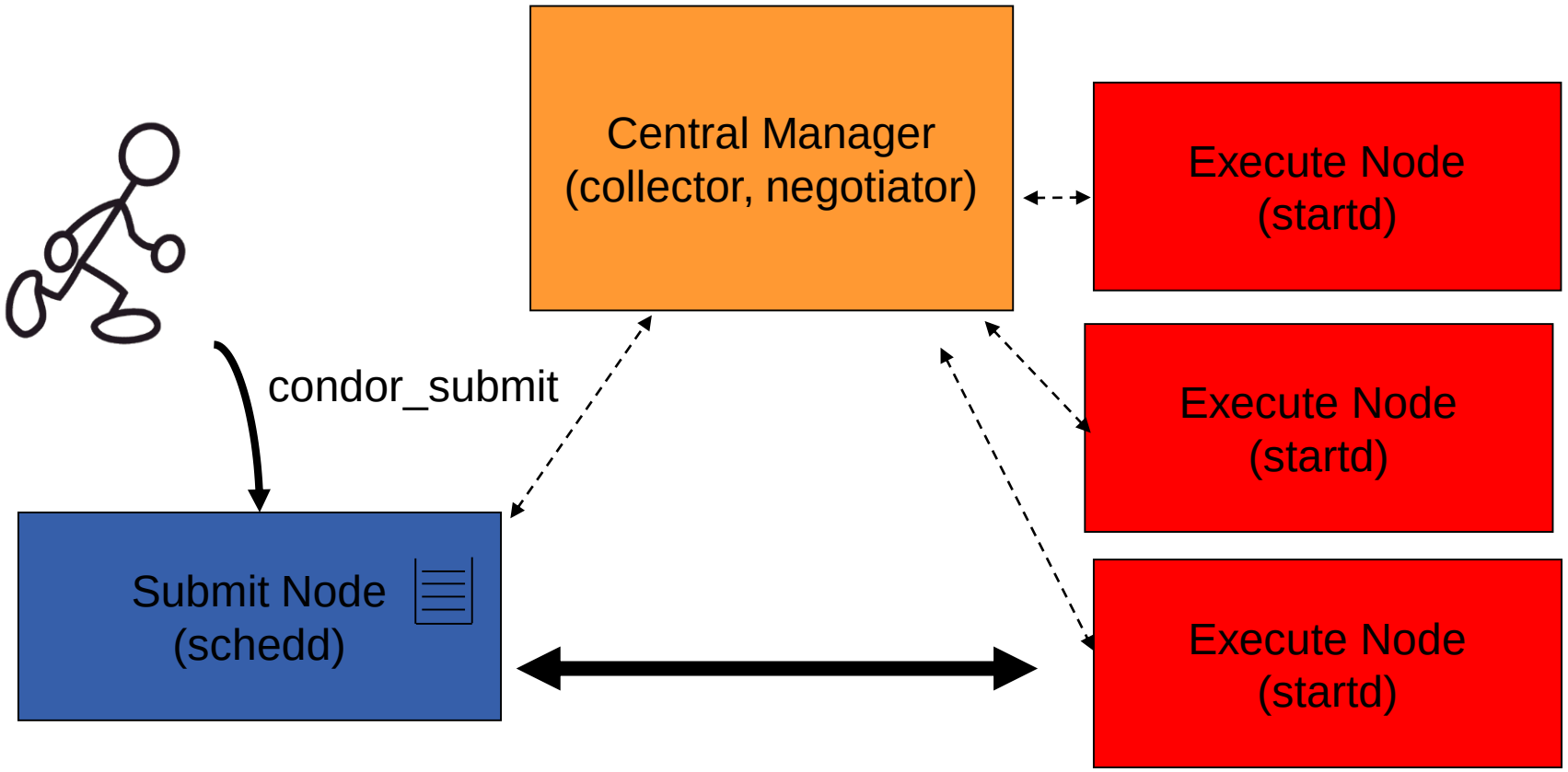
## Job Ad

```
Universe = "Vanilla"
AcctGroup = "physics"
RequestOS = "CentOS"
RequestMemory  = 8000
Requirements =
    (OPSYS == RequestOS) &&
    (Cpus >= 1) &&
    (Memory > RequestMemory)
```

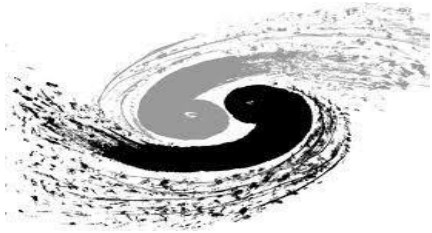
## Job Slot Ad

```
OpSysName = "CentOS"
Memory =10531
Requirements =
  (Universe == "vanilla")  &&
  (AcctGroup == "physics" ||
   AcctGroup=="lhaaso" ||
   AcctGroup=="juno")
Rank = physics
```

# HTCodor 资源调度器

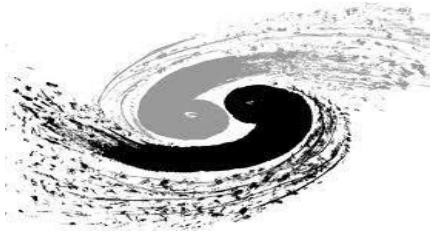


# HTCondor 调度算法的理念 – 公平共享



- 用户使用计算资源的公平性
  - 用户作业优先级 -- 动态调整
    - 基于最近时间过内用户使用资源的总和进行优先级排序
      - 跑得越多的用户，其作业优先级越低
      - 优先级按指定周期内作业累计运行时间总和半衰递减
- 不同用户组使用资源的公平性
  - 用户组的资源共享
  - 用户组可用资源受限于各组资源份额
    - Surplus模式:
      - 当有空闲资源时，可突破本组份额，占用更多资源
      - 份额未用足的用户组作业优先级最高
      - 共享池子越大→利用率越高，份额越容易保证

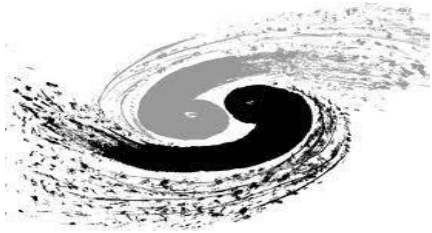
# HTCondor 调度算法的理念 – 公平共享



$$\frac{\sum Job\ Runtimes}{Wall\ Time} \Longrightarrow \frac{\sum Completed\ Job\ Runtime}{Wall\ Time} \Longrightarrow \left(\frac{\sum Completed\ Job\ Runtime}{Wall\ Time}\right)^{*}$$

* Subject to some notion of fairness

# HTCondor的运行模式特点 – 信任

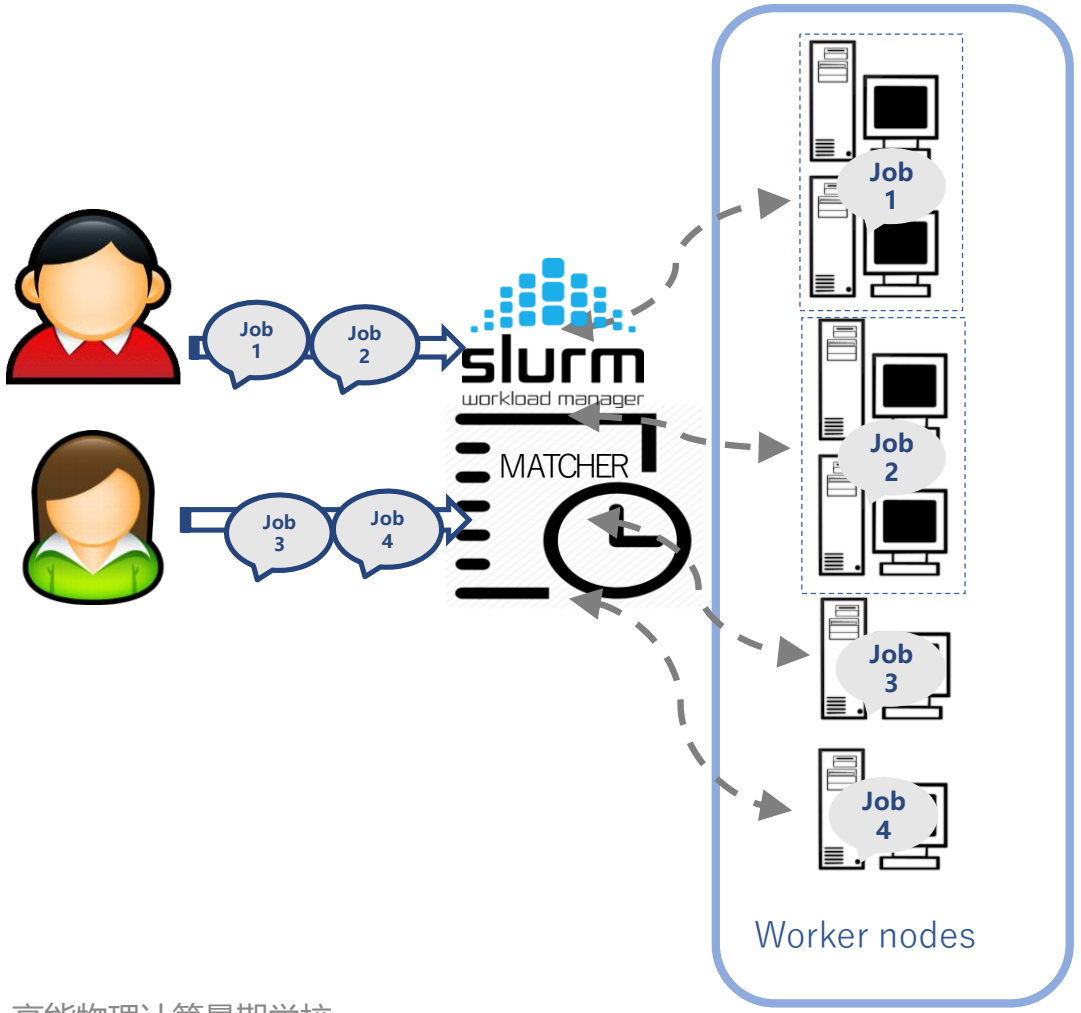


- HTCondor对计算资源的管理：计算节点自己汇报运行策略，作业结束状态，Condor不干涉（杀死节点），只收集节点拥有的资源情况
- HTCondor对作业的管理：作业自己向HTCondor汇报自身需求和具备条件
- HTCondor对数据的管理：
  - 对存储系统无要求
  - 支持 共享文件系统/sandbox传输/第三方传输协议
- HTCondor安全易扩展：轻量级、无状态的中央管理机制，支持SSL, Kerberos, GSI, HostIP, password, security session
- HTCondor 对作业的管理： 多种作业类型，详细作业生命周期日志

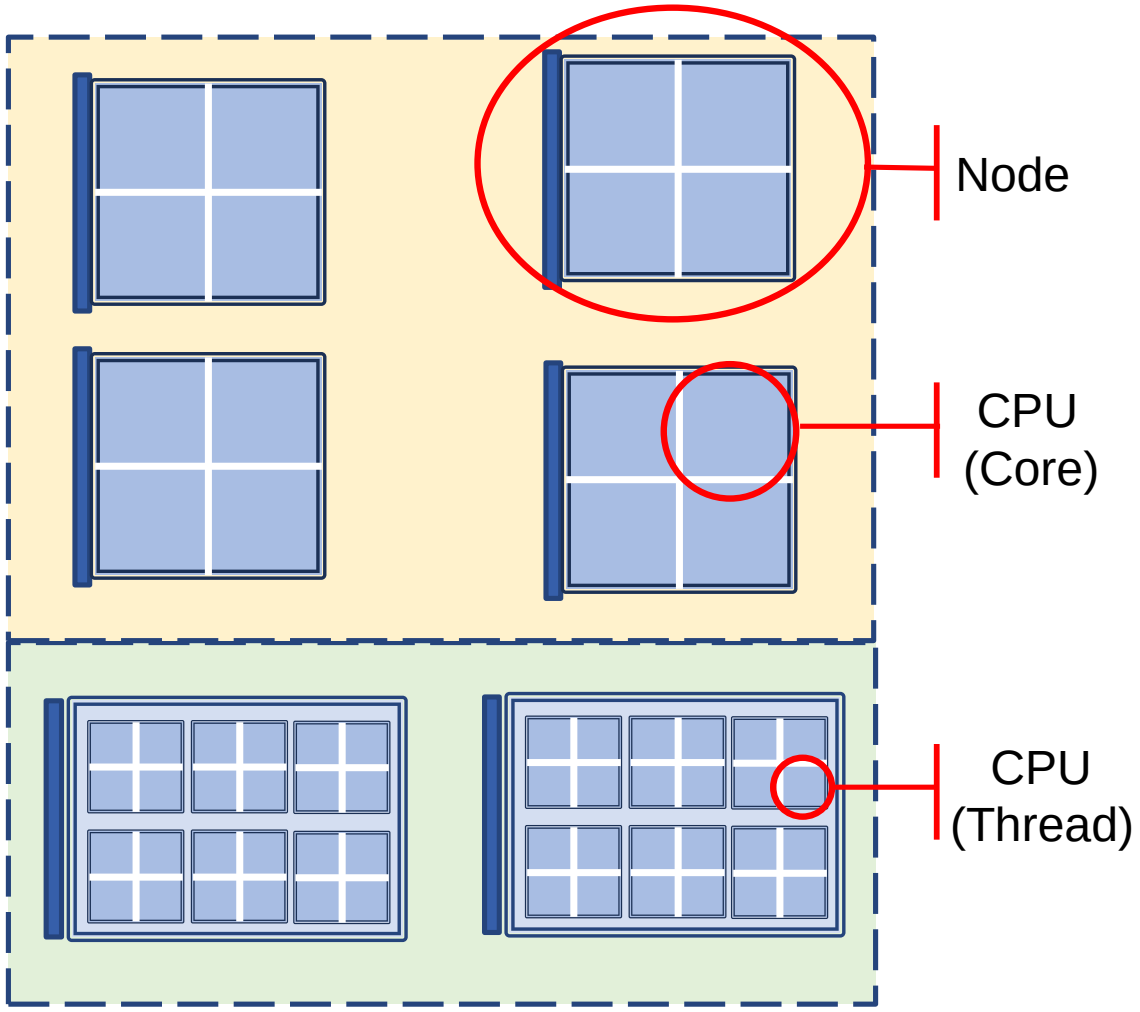
# HPC作业调度与资源管理 -- SLURM

## Slurm -- Simple Linux Utility for Resource Management

- 开源，容错，可扩展软件
- 提供作业的开始，执行，监视整个生命周期操作
- 连接计算资源和队列中的作业
- 对大并行作业支持良好
- 在超算中心得到广泛应用
  - 被应用于大量非高能物理领域
  - 高能物理领域使用越来越多



# SLURM资源管理




## 计算节点

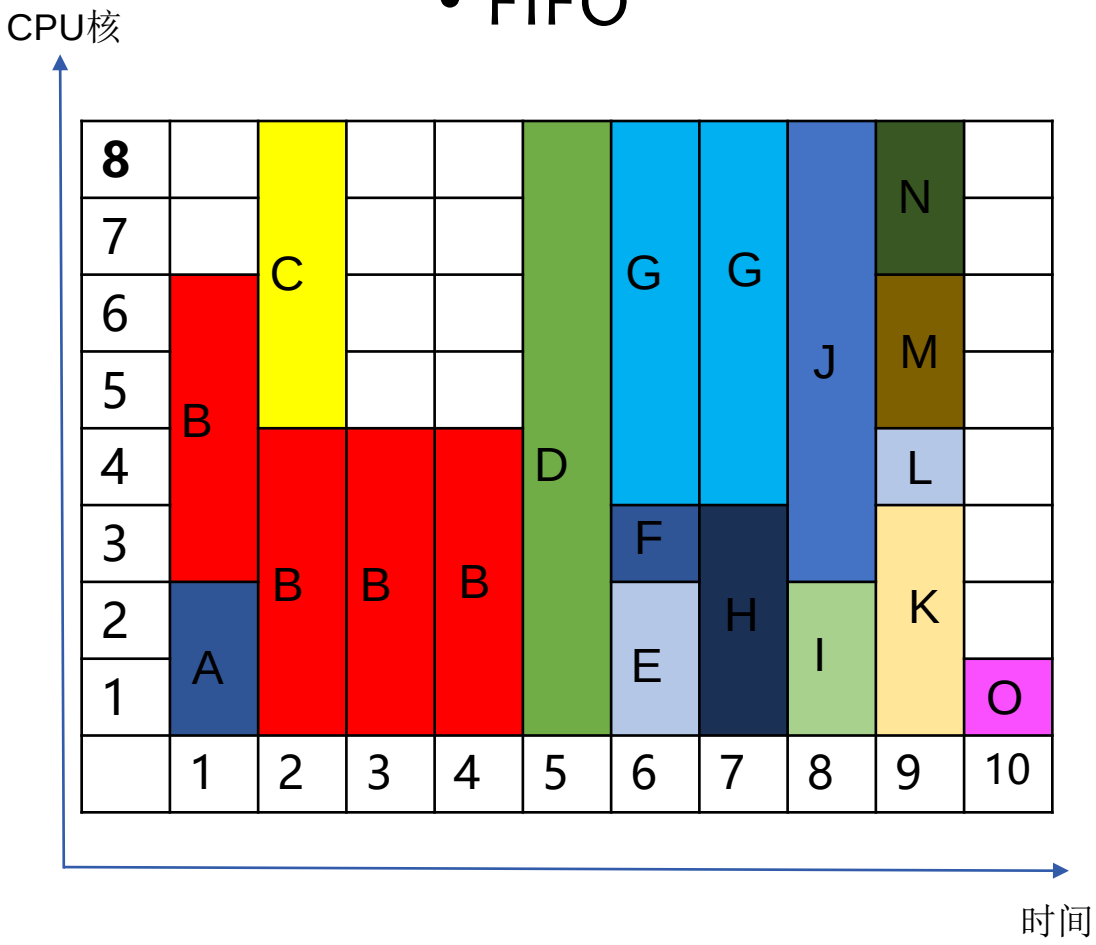
- CPUs
  - Cores
  - threads
- 内存
- 状态
  - Idle
  - Mix
  - Alloc
  - Completing
  - Drain/ing
  - Down

## Partition

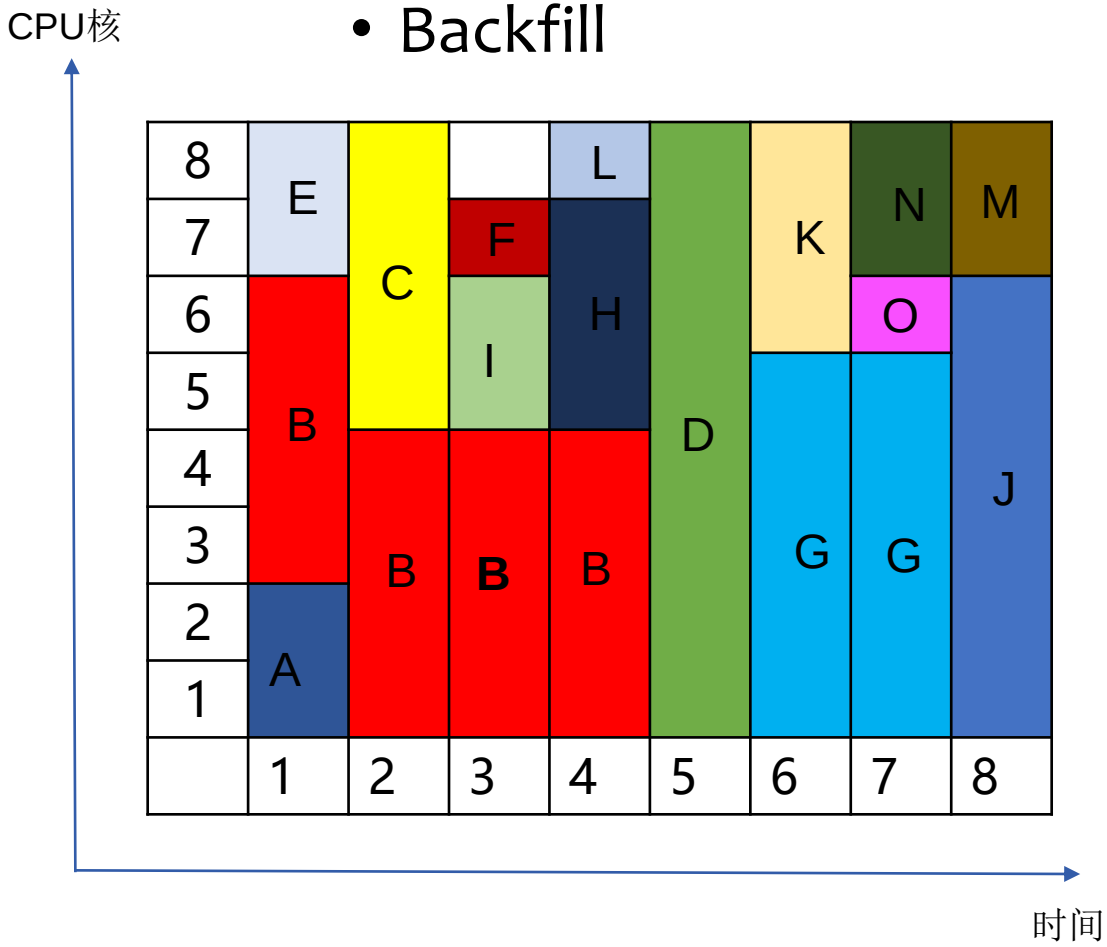
- 设定的节点集
  - 同构
- 状态
  - Idle
  - Mix
  - Alloc
  - Completing
  - Drain/ing
  - Down

# Slurm具有丰富的调度策略

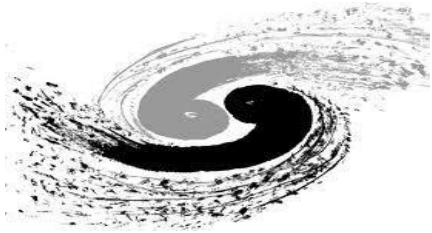
- FIFO




- Backfill

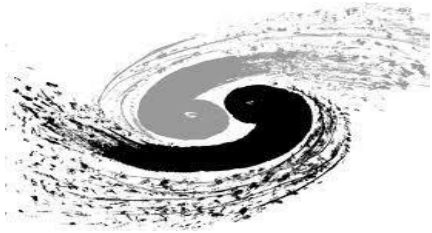


# 提纲



1. 高能物理离线数据处理过程
2. 高能物理离线数据处理特点
3. 高通量计算与高性能计算
4. **网格计算与分布式计算**
5. 高能所计算平台
6. 总结

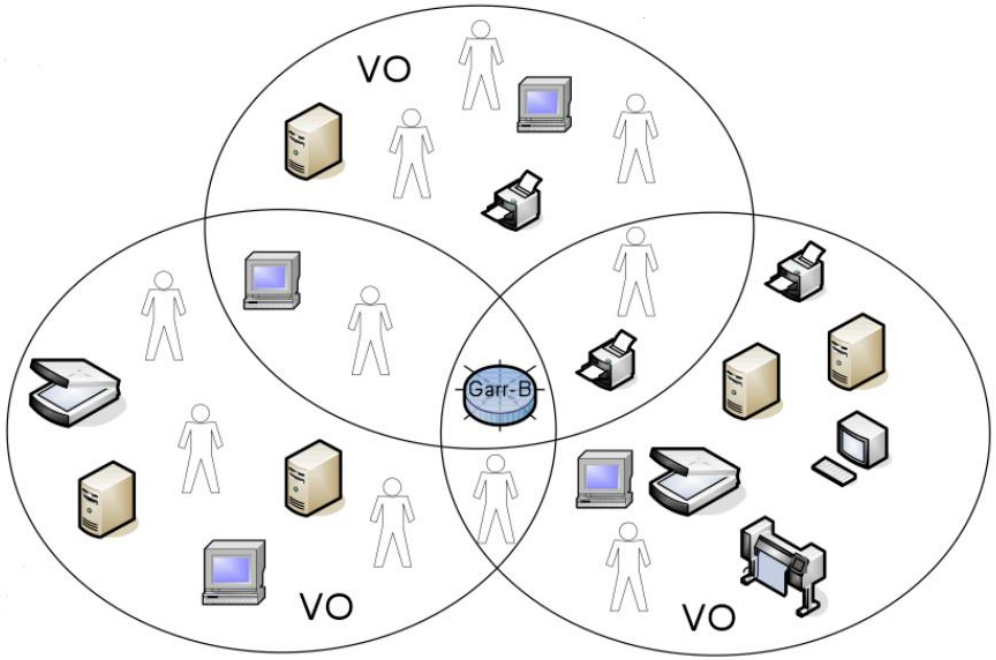
# 网格计算



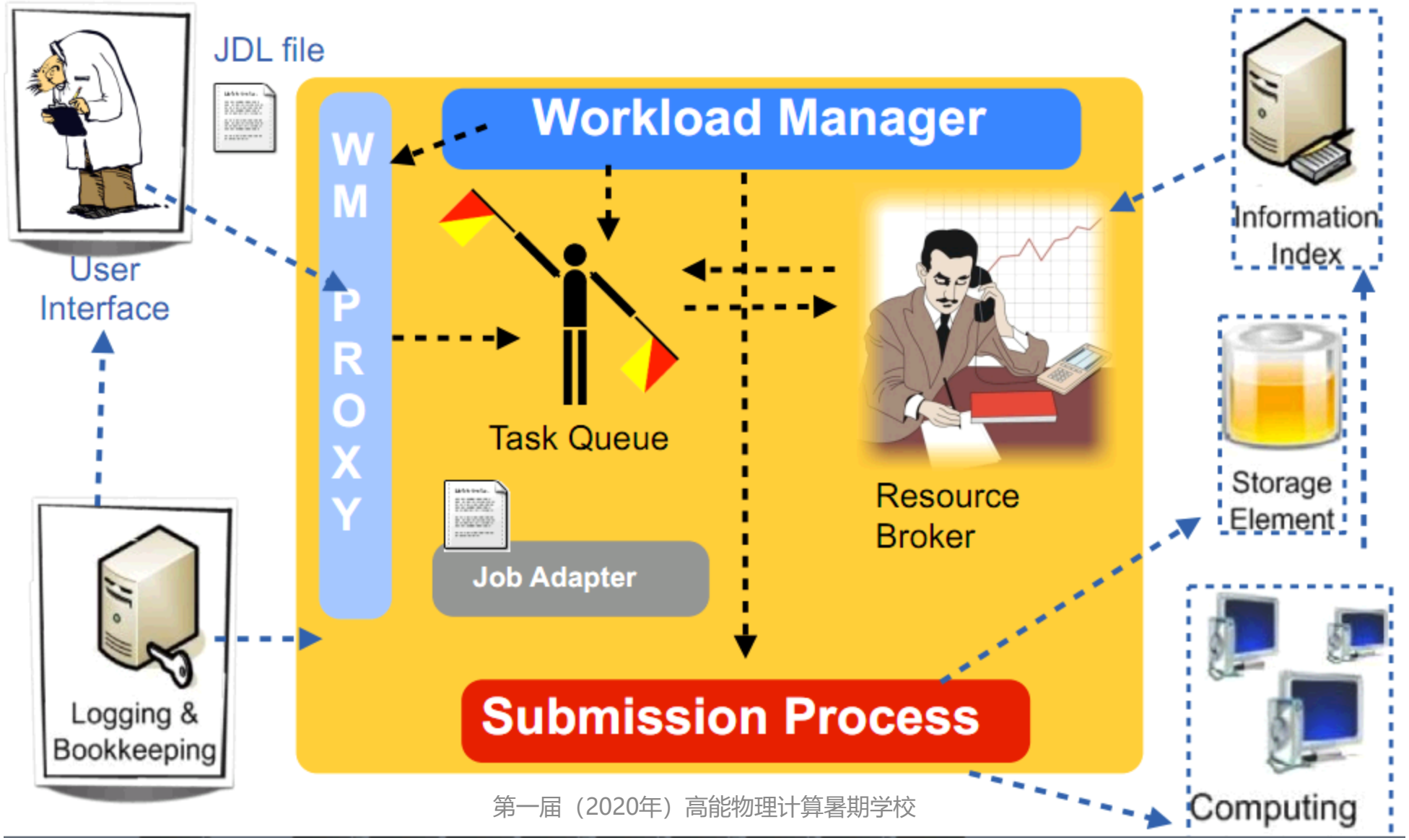
- 什么是网格?
  - 网格技术将全球地理上分散的计算资源有机的整合起来，协同工作，为大型科学实验研究提供计算支持，能够完成单个集群无法完成的大规模计算任务。
- 网格技术的主要体现
  - 网格是一台超级计算机，拥有大量的计算资源与存储资源，用户可以透明使用
  - 通过网格，最大程度的实现任务调度与资源共享，在全球范围内合理的分配资源
  - 在共享资源的同时最大程度的保证网格资源的安全性。
- 网格的现状
  - 网格技术是成熟技术，得到广泛应用。世界各国或者国际间合作启动了多个网格项目。网格已成为高能物理、生物医学等领域科学家日常使用的计算基础设施，发挥了极其重要的作用

# 虚拟组织

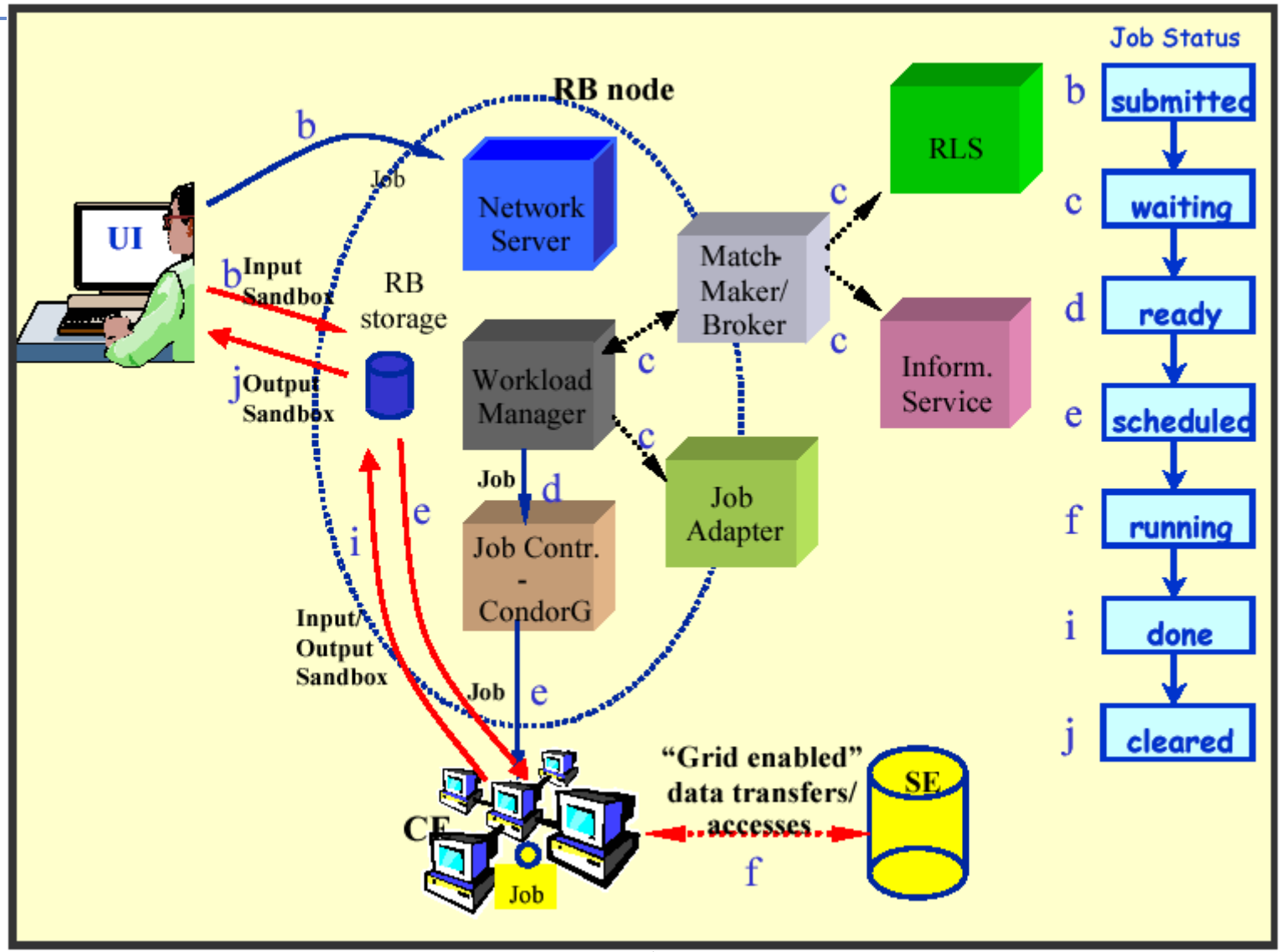
- 地理分布的不同单位人员可以按虚拟组织划分以共享计算，存储等资源 ---- 合作
- 网格成员可以同时跨多个虚拟组
- 从LCG开始的原型
  - 提供资源共享
  - 虚拟组织：合作与权限设定



# 网格重要组件

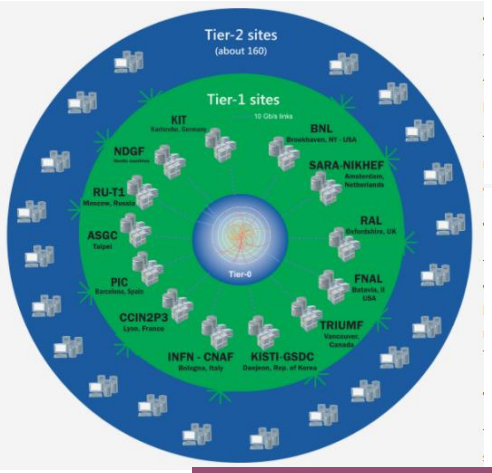

JDL file
User Interface
Workload Manager
WM PROXY
Task Queue
Job Adapter
Resource Broker
Information Index
Storage Element
Logging & Bookkeeping
Submission Process
Computing

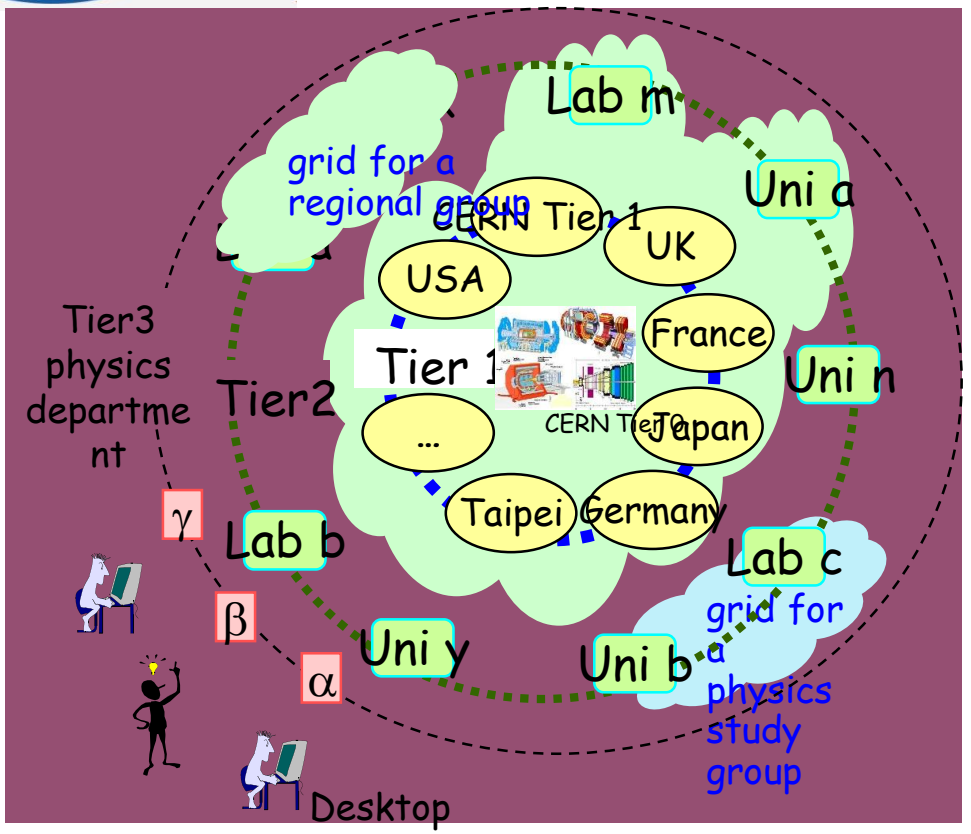
# 网格作业流


RB node
UI
Network Server
RB storage
Workload Manager
Match-Maker/ Broker
RLS
Inform. Service
Job Adapter
Job Contr. - CondorG
CE
SE
Job
Input Sandbox
Output Sandbox
Input/ Output Sandbox
"Grid enabled" data transfers/ accesses
Job Status
submitted
waiting
ready
scheduled
running
done
cleared

# LHC网格现状

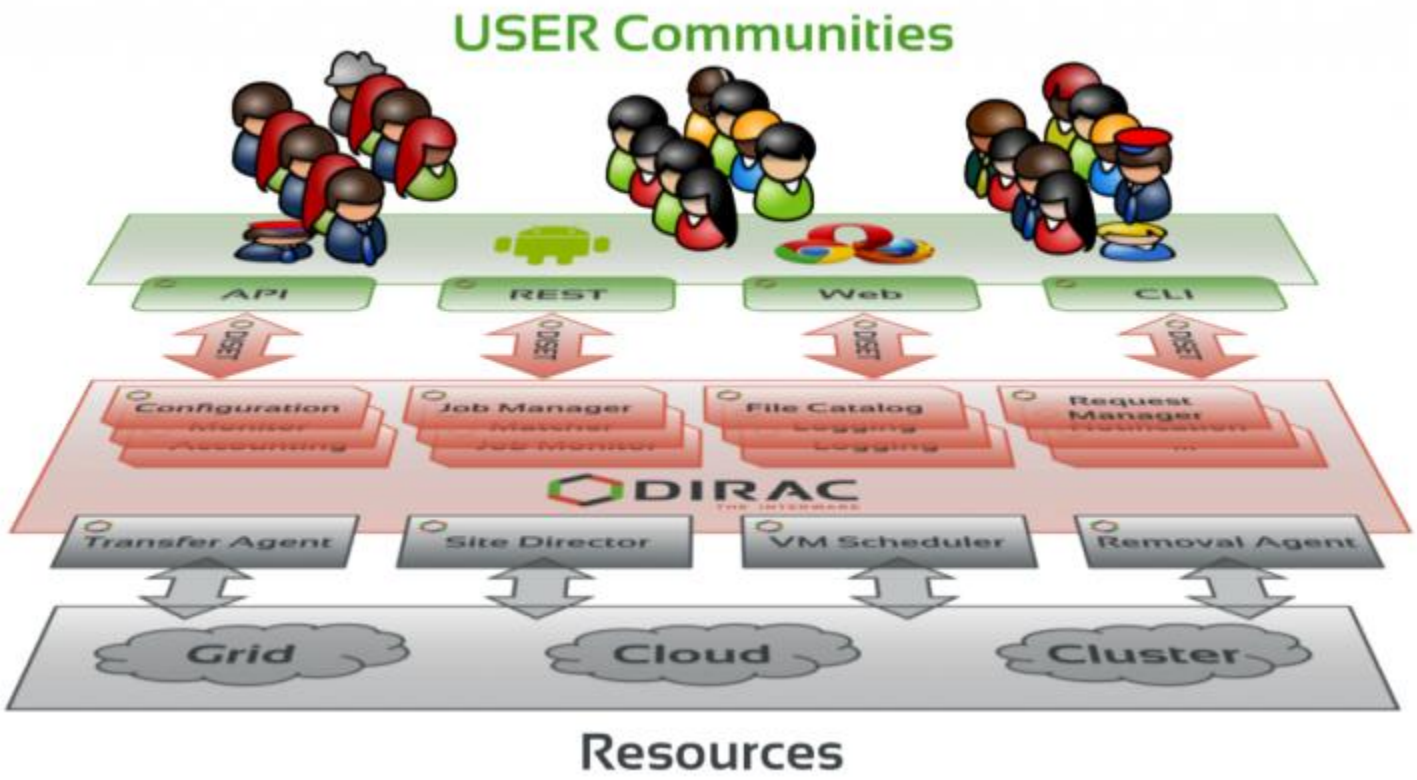
- 世界最大的网格项目
- 42个国家，170个计算中心
- 1M CPU 核每天运行2M计算任务
- 1EB的数据存储
- 分层
  - Tier 0: Cern 数据中心
    - Custodial 存储
    - First-pass 重建
  - Tier 1：13个大型计算中心
    - Custodial 存储
    - Reprocessing
  - Tier2：遍布全球的160个计算中心
    - 事例模拟
    - 最终用户（物理学家）分析
  - Tier 3：小型计算中心
    - 小型计算设施



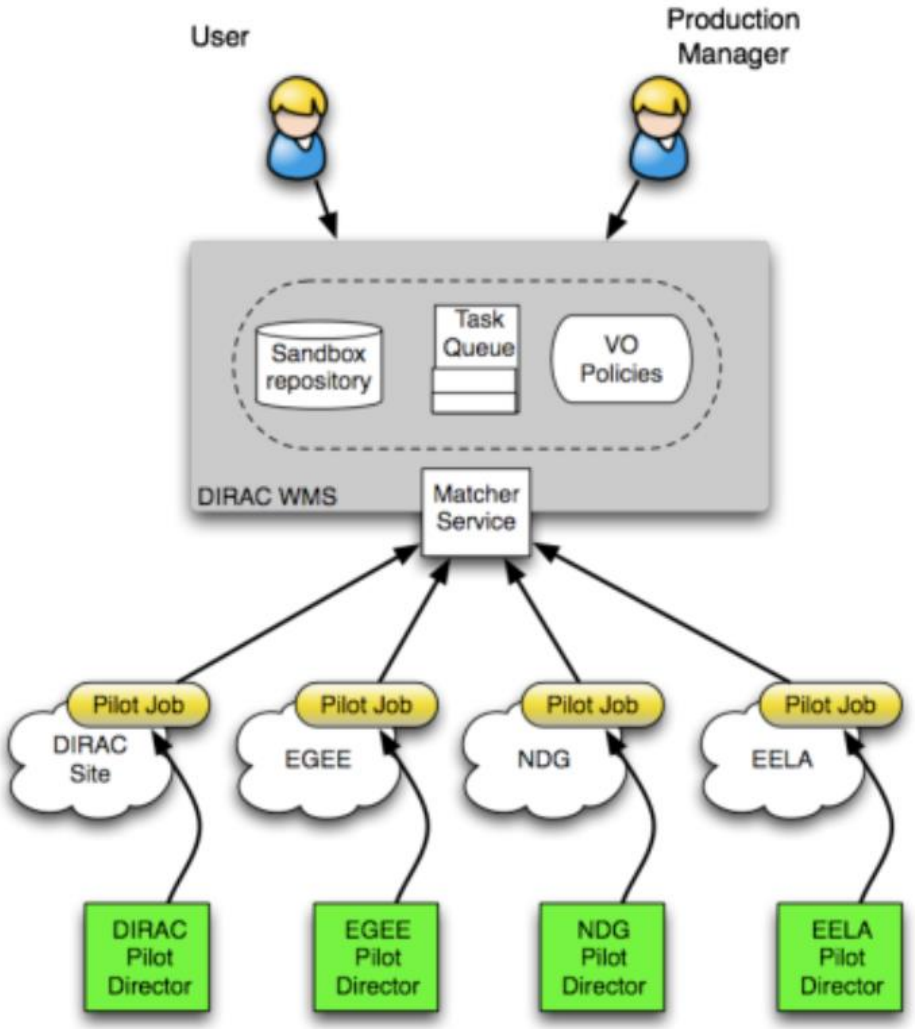
# 分布式计算 -- Dirac

- 开源分布式软件框架
- 在用户和资源之间的中间层
- 与实验无关
  - 可扩展，灵活性高
- 可整合多种资源
  - 集群，网格，商业云
- 功能完备
  - 基于面向服务架构
  - 资源调度、数据管理、文件传输、监视记账。。。。

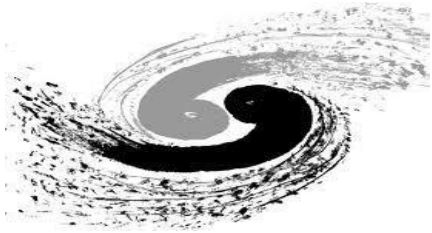


# Dirac作业调度 -- 拉作业

- “拉作业”的pilot机制
  - Pilot用于为作业准备所有运行环境及相关配置
  - 用户以证书及VO将作业提交到 Central Task Queue
  - Directors向Grid WMS提交指定用户角色的pilot作业
  - Pilot作业从TaskQueue里拉取用户作业
  - 作业被pilot监管下执行。



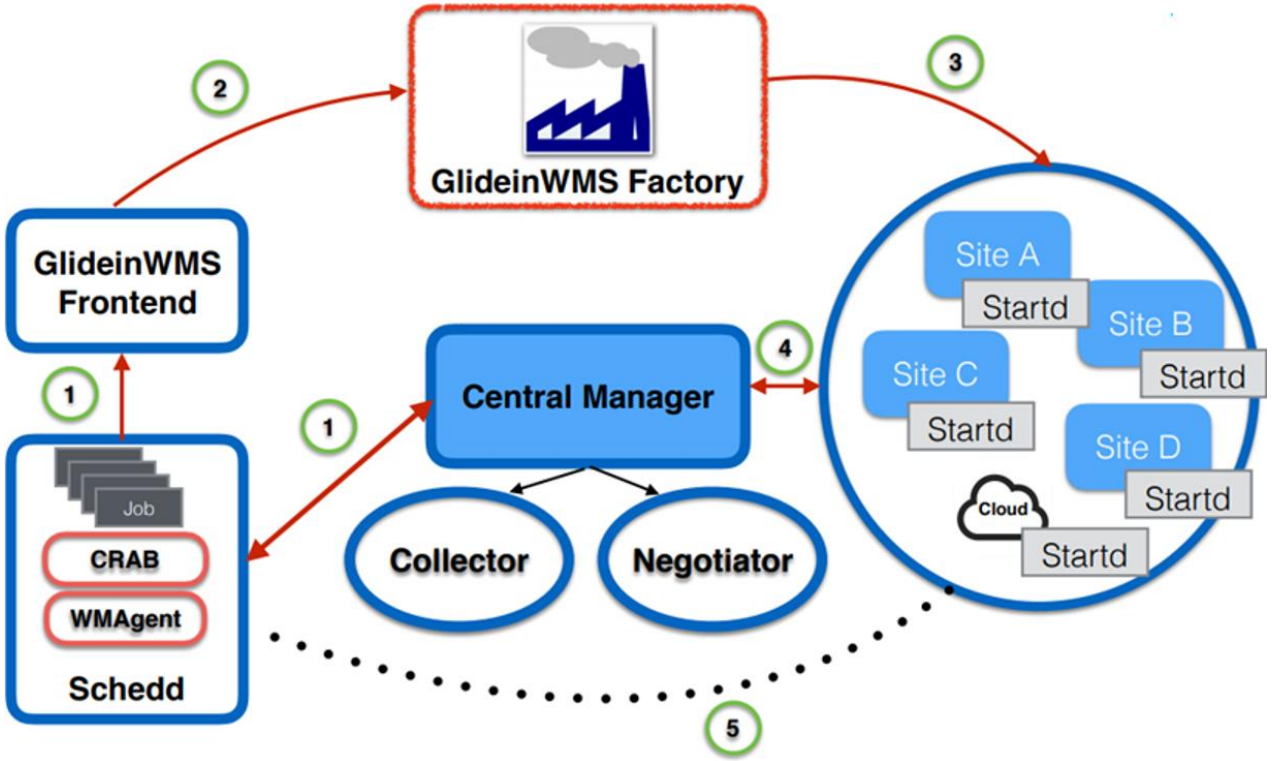
# Dirac 的应用

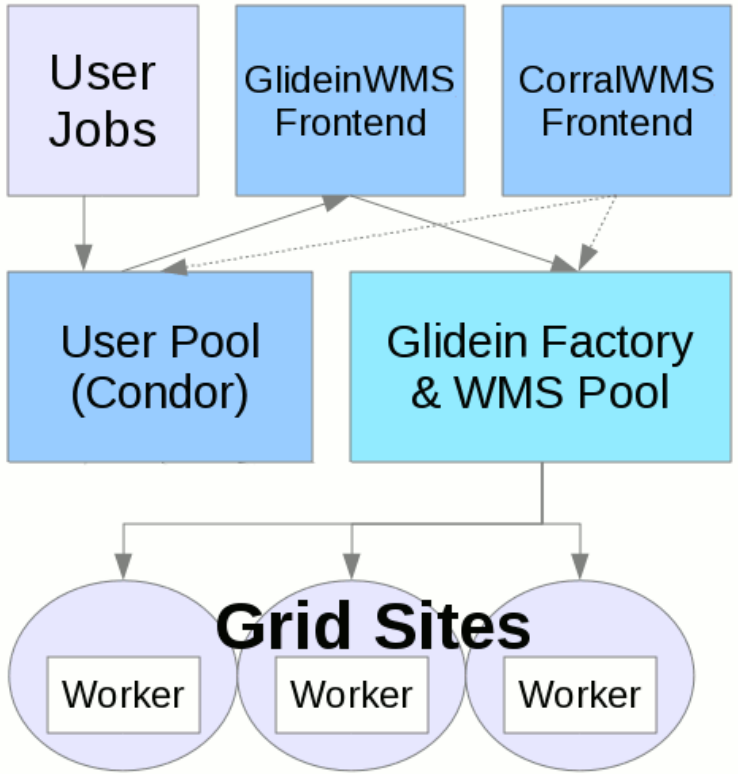


- Dirac由LHCb实验开发，用于WLCG资源整合，成功应用于LHCb, BelleII等多个高能物理实验，用户群体包括LHCb, ILC, Belle II 等实验，还拓展应用于天文，生物领域，包括T2K, CTA, Pierre Auger Observatory, Eiscat 3D, BioMed 等
- 2014年，高能所自主建成基于DIRAC的分布式计算系统，为 BESIII、JUNO、CEPC 等中国高能实验服务，现已整合包括美国、俄罗斯、意大利、英国、土耳其、法国、台湾、北大、北航、国科大、中科大、上海交大、武汉大学、等十多个分布式站点，资源类型覆盖集群、云、网格以及志愿者计算等

# 分布式计算 -- Global Pool

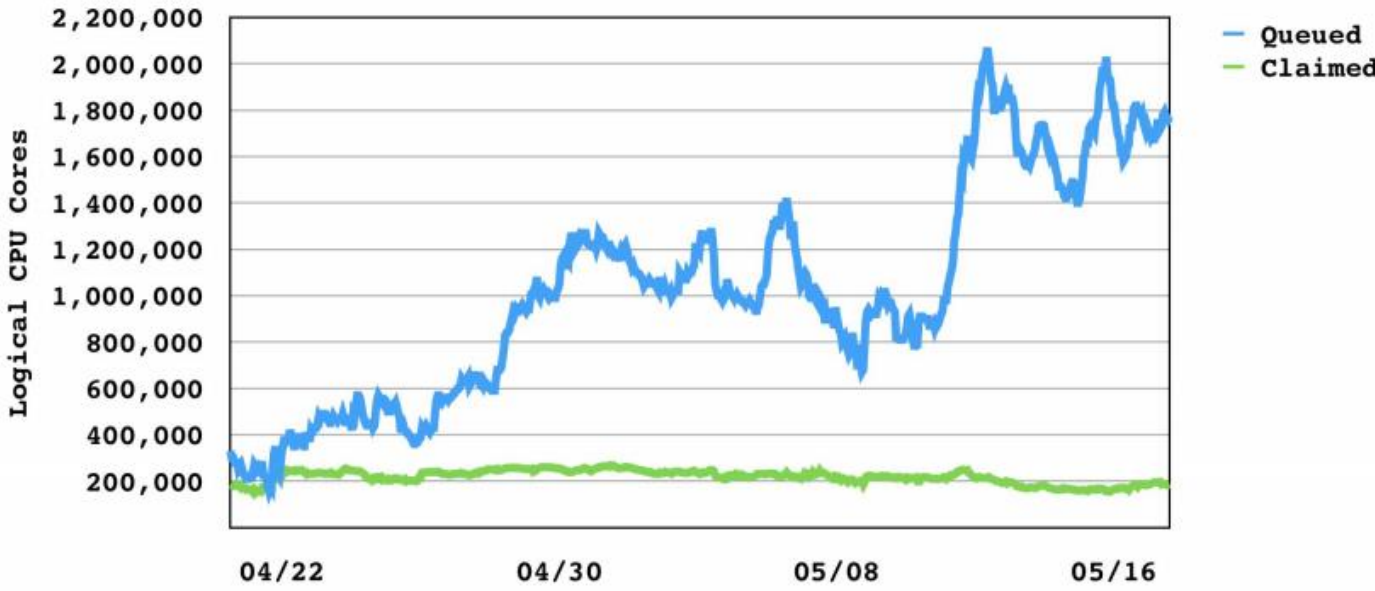
- 基于HTCondor的资源整合，成功应用于CMS大型实验

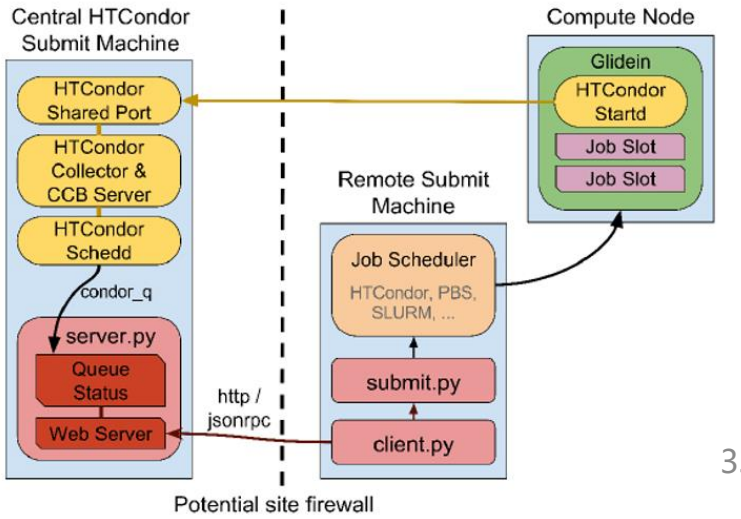


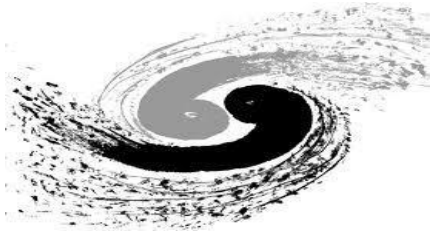
# 分布式Global Pool

- 可以整合多类资源
  - 集群，网格，云资源
- 支持大规模作业--成功应用于CMS
  - **20万**动态作业槽运行在**30万**逻辑CPU 核
  - ~700分析用户
  - 集成了30+ sched
  - 作业量达到1-2M
- 在一些相对小型实验也有非常好的应用
  - IceCube： 简化的HTCondor glidein service



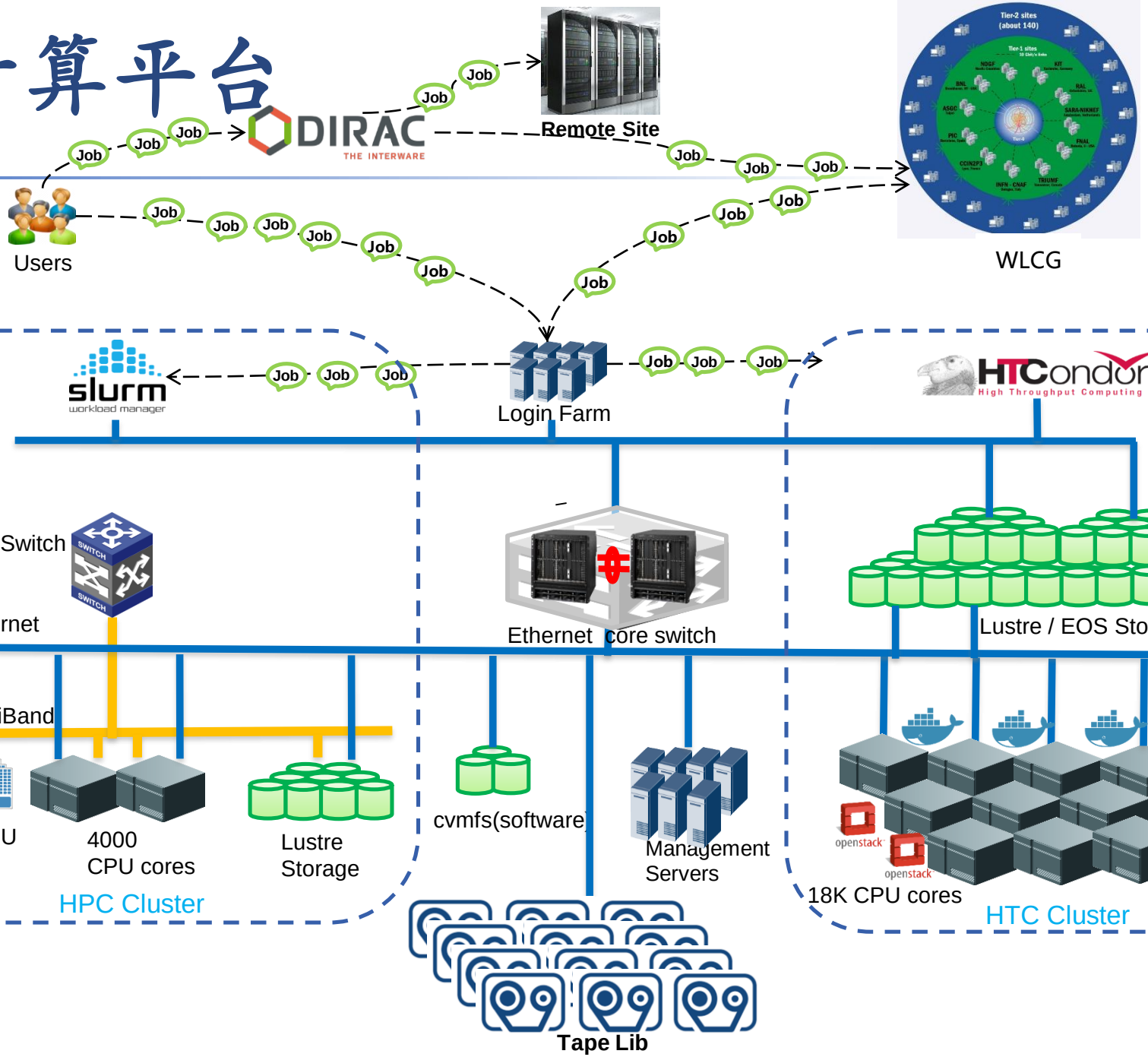



# 提纲



1. 高能物理离线数据处理过程
2. 高能物理离线数据处理特点
3. 高通量计算与高性能计算
4. 网格计算与分布式计算
5. **高能所计算平台**
6. 总结

# 高能所计算平台


Job
DIRAC
THE INTERWARE
Remote Site
Users
WLCG
slurm
workload manager
Login Farm
HTCondor
High Throughput Computing
IB Switch
10Gb Ethernet
Ethernet core switch
Lustre / EOS Storage
EDR InfiniBand
180 GPU cards
4000 CPU cores
Lustre Storage
HPC Cluster
cvmfs(software)
Management Servers
18K CPU cores
HTC Cluster
Tape Lib

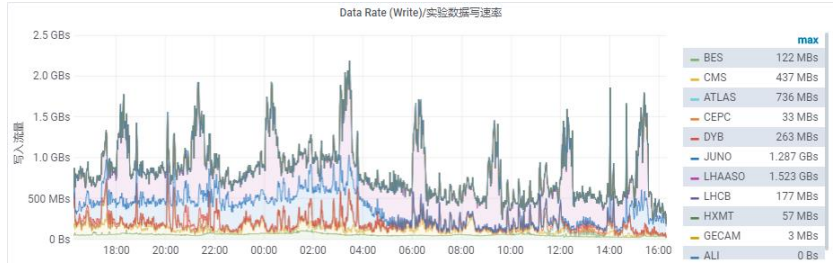
# 计算及存储资源

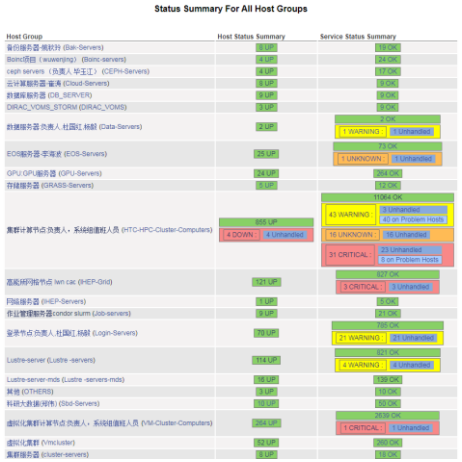
- HTC集群：~18K CPU核
  - 资源共享，实验定制化服务
  - 特殊需求：长短队列、大内存等
  - 基于虚拟化的多操作系统支持
- HPC集群：180块GPU卡+4K CPU核
  - 并行计算，GPU计算
- Grid Tier2 站点：1.9K CPU核
  - 国际合作站点
- Hadoop 集群：
  - 特点计算模式支持
- Dirac 分布式计算：3.5K CPU核
- IHEPCloud based on openstack: 2K CPU cores
  - IHEP私有云
- Lustre存储： 20PB
- EOS: 10PB
- 磁带： 8PB
- AFS, cvmfs, home， 备份： 百TB

# 计算平台的智能化管理


FrameWork
综合监控
报表管理
流程管理
运维管理
资源管理
用户门户
标准化
一体化
开发平台
智能运维
可定制
易集成
保障
知识管理
信息安全
流程管理
服务级别管理
事件
服务台
故障/问题
日常流程
服务请求
变更/发布
资源
容量管理
资产管理
可用性管理
服务管理
流程配置
流程定义引擎
人员、权限管理
基础组织信息
配置管理
CMDB
业务模型
自动化运维
服务器自动化
网络自动化
桌面自动化
应用发布自动化
软件安装、配置
作业调度
日常巡检、安全检查
脚本执行
OS、补丁安装
信息收集
智能IT运维
监控平台
主机监控
安全监控
系统监控
存储监控
网络监控
环境监控
主机
CPU
内存
磁盘
存储
存储数据设备
存储网络设备
网络
路由器
交换机
防火墙
负载均衡
网络接口
网络链路
网络流量
通用资源
数据库
中间件
操作系统
应用监控
业务应用
消费应用
网络应用
计算应用
资源监控

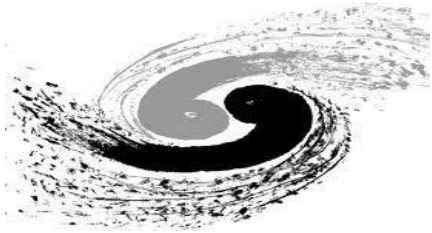

Running jobs / 各实验作业运行数统计
Job count
8.0 K
7.0 K
6.0 K
5.0 K
4.0 K
3.0 K
2.0 K
1.0 K
0
18:00
20:00
22:00
00:00
02:00
04:00
06:00
08:00
10:00
12:00
14:00
16:00
max
lhaaso 5.598 K
juno 3.890 K
dyw 1.278 K
cms 667
atlas 50
hxmt 29


Data Rate (Write)/实验数据写速率
写入流量
2.5 GBs
2.0 GBs
1.5 GBs
1.0 GBs
500 MBs
0 Bs
18:00
20:00
22:00
00:00
02:00
04:00
06:00
08:00
10:00
12:00
14:00
16:00
max
BES 122 MBs
CMS 437 MBs
ATLAS 736 MBs
CEPC 33 MBs
DYB 263 MBs
JUNO 1.287 GBs
LHAASO 1.523 GBs
LHCB 177 MBs
HXMT 57 MBs
GECAM 3 MBs
ALI 0 Bs


Status Summary For All Host Groups
Host Group
Host Status Summary
Service Status Summary

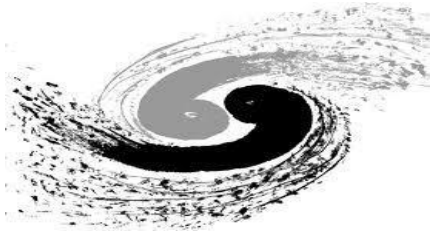

NMS报警
NMS 报警
类型: PROBLEM
主机: lhoss02.ihep.ac.cn
IP: 192.168.50.123
服务: check_load_lustre
状 态: CRITICAL
时间: 2020-08-14 22:59:45
输出:CRITICAL - load average: 242.81, 112.95, 44.47
NMS 报警
类型: PROBLEM
主机: lhoss01.ihep.ac.cn
IP: 192.168.50.118
服务: check_load_lustre
状 态: CRITICAL
时间: 2020-08-14 22:59:57
输出:CRITICAL - load average: 239.72, 116.44, 46.25

# 提纲



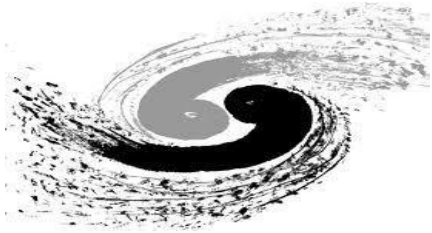
1. 高能物理离线数据处理过程
2. 高能物理离线数据处理特点
3. 高通量计算与高性能计算
4. 网格计算与分布式计算
5. 高能所计算平台
6. 总结

# 总结



- 高能物理离线处理需要有力的计算服务支持
- HTC, HPC计算是高能物理计算的基础
- 分布式计算整合更多高能物理资源协同工作
- 计算平台的建构与运维已成为高能物理实验密不可分的一部分，需要不断完善与改进

# Backup -- HTCondor USER PRIRIOTY



- The RUP of a user u at time t, $\pi_r(u,t)$, is calculated every time interval $\delta t$ using the formula

$$\pi_r(u,t)=\beta \times \pi_r(u,t-\delta t)+(1-\beta)\times \rho(u,t)$$

where $\rho(u,t)$ is the number of resources used by user uu at time t, and $\beta=0.5^{\delta t/h}$. h is the half life period set by PRIORITY_HALFLIFE .

The EUP of user u at time t, $\pi_e(u,t)$ is calculated by

$$\pi_e(u,t)=\pi_r(u,t)\times f(u,t)$$

where f(u,t) is the priority boost factor for user uu at time t.