凯习信息
CACHEINFO

HTCondor与跨域资源整合

杜治高　2019.11

目录 / CONTENTS

01

HTCondor简介

1.1 HTCondor概况

- HTCondor是一个开源的分布式作业调度软件，由UW-Madison Miron Livny教授1994年发布，广泛应用于科学计算领域。
- HTCondor更适合HTC应用场景。
- HTC vs HPC

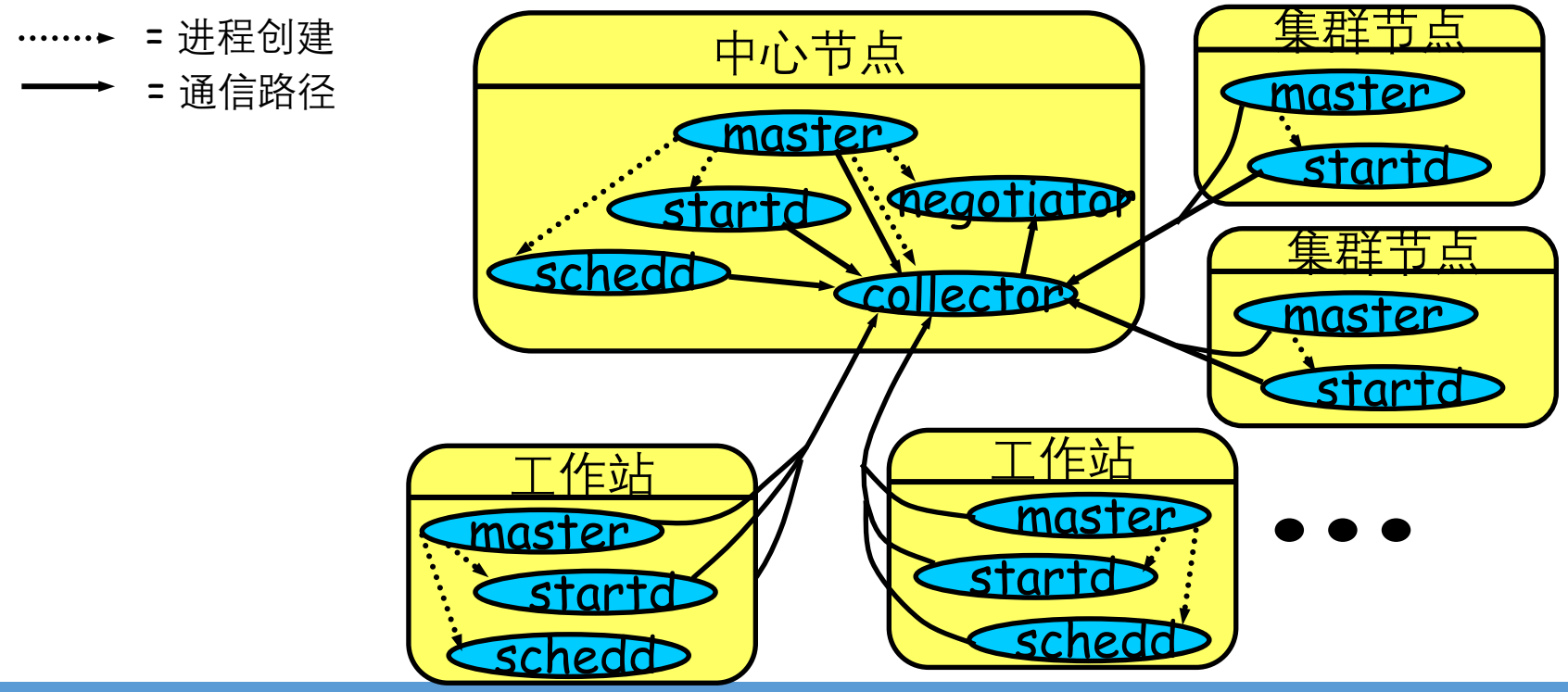
HTC is about many jobs, many users, many servers, many sites and long running workflows

FLOPY ≠ *(60*60*24*7*52)*FLOPS*

100K Hours*1 Job ≠ 1 H*100K J

1.2 HTCondor软件结构

- HTCondor管理的一个集群称为一个Pool，Pool中的节点分成三种类型：中心节点、提交节点、计算节点
- 三种类型不同角色的节点运行同一套软件但启动不同的进程。



1.3 关键特性

高效整合离散分布式计算资源

- 支持集群、工作站、台式机等
- 支持Windows、Linux、Unix、Mac OS X各种操作系统

支持多种作业类型

- 串行作业、MPI作业、Java作业、虚拟机作业、docker作业等

提供不同粒度的安全和可靠性机制

- 检查点、重启机制、证书/密钥等安全通讯协议

支持灵活的作业调度和资源匹配

- 资源、作业双向匹配（广告机制）
- 以核、处理器和节点等为基本资源单元
- 与网格、云资源互操作

支持复杂科学工作流

- 通过有向无环图（DAG）实现作业间依赖关系

1.4 作业和资源双向匹配

作业属性

```
Type = “job”
TargetType = “machine”
Cmd = “sim.exe”
Owner = “thain”
Requirements =
(OpSys== “linux” )
```

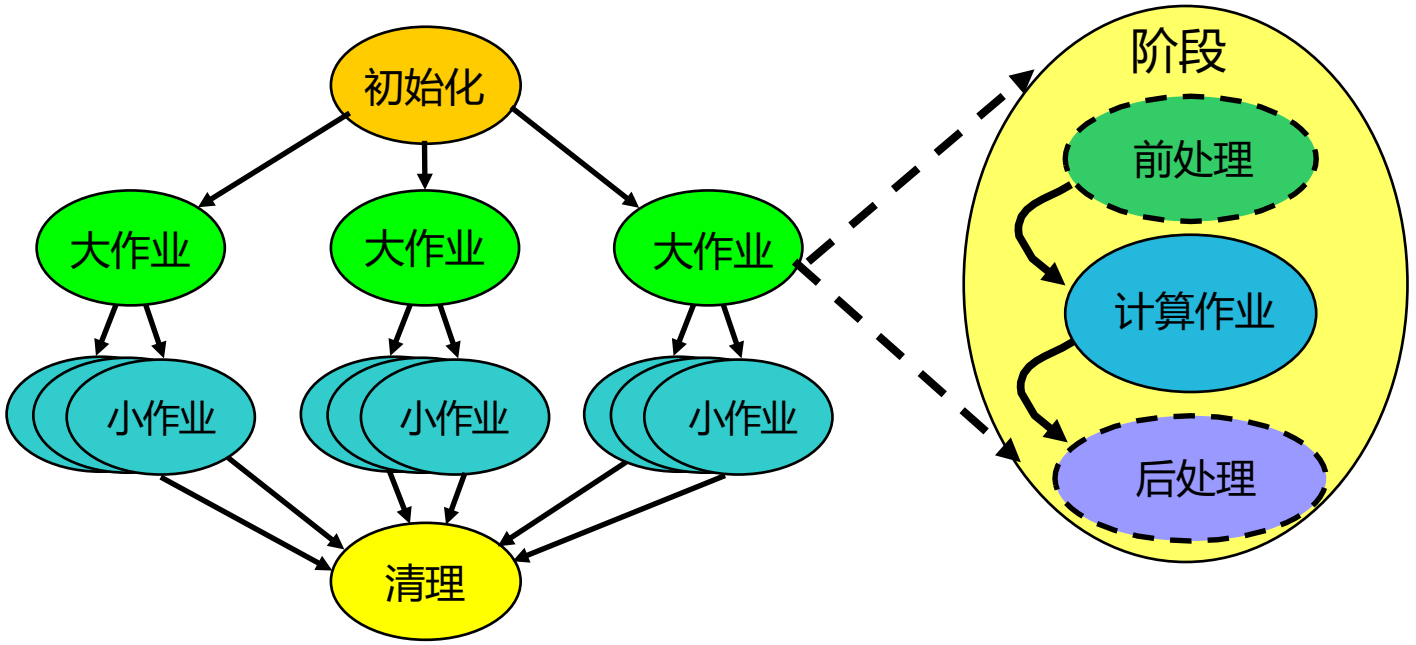
资源属性

```
Type = “machine”
TargetType = “job”
OpSys = “linux”
Requirements =
(Owner== “thain” )
```

Requirements = Memory >= 256 && Disk > 10000
Rank = (KFLOPS*10000) + Memory

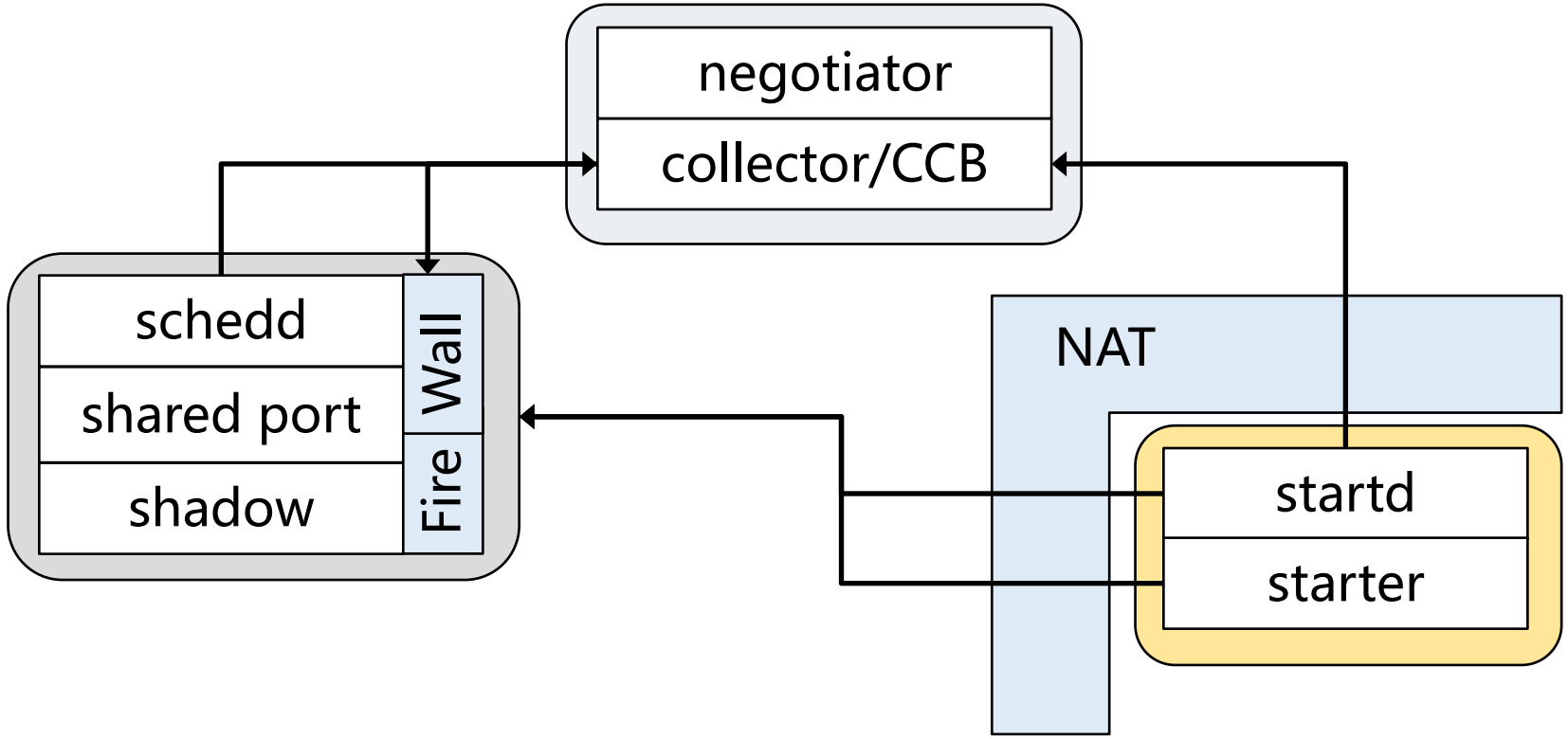
- 支持作业属性和资源属性的双向匹配
  - 支持匹配偏好表达式，定义丰富的优先机制
- 可以灵活扩展作业和资源的属性
- 支持灵活的资源管理策略
  - 优先执行owner或者某个用户组的作业；优先选择上次成功执行作业的节点

1.5 支持复杂的工作流定义



- 支持大规模科学工作流的构建和运行
- 每个工作流节点内可以支持前处理和后处理

1.6 支持IPv4，IPv6，端口共享，NAT



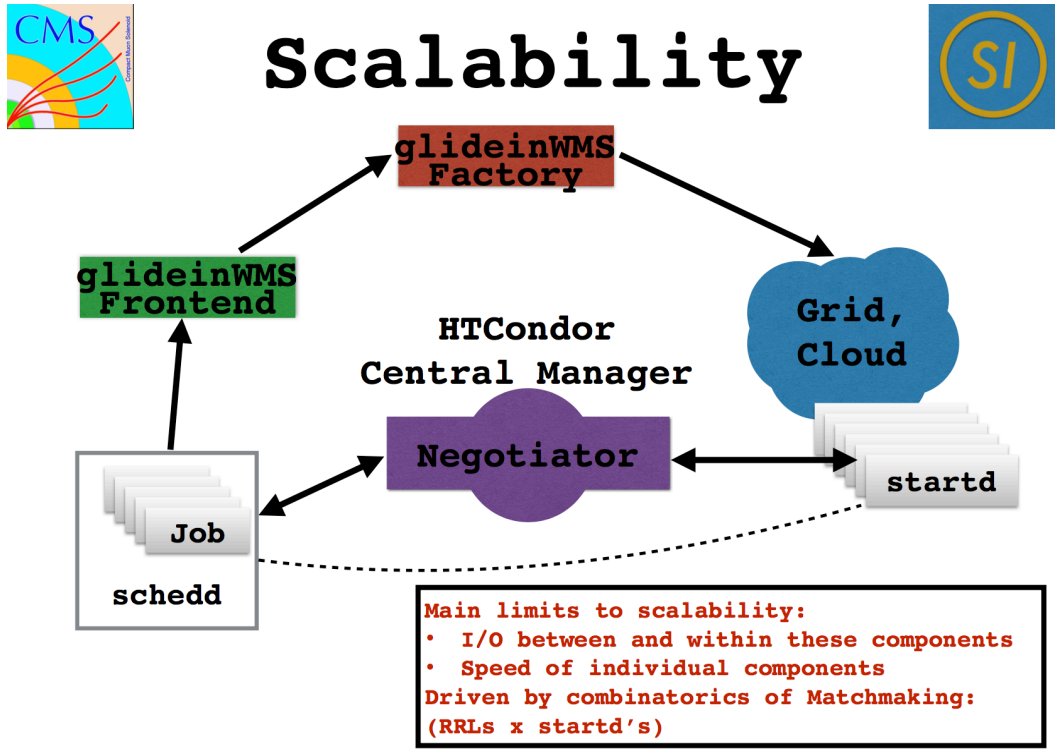
1.7 应用: Open Science Grid (OSG)

National HTComputing

1.7 应用: OSG资源使用统计

1.7 应用: HTCondor帮助发现上帝粒子

- CERN欧洲核子研究组织使用HTCondor替代了LSF
- 在2013年发现上帝粒子的实验中，计算平台的调度系统就是HTCondor



1.7 应用: 2013 Nobel Prize in Physics

Armed with 5σ significance delivered by more than 6K scientists from the ATLAS and CMS LHC experiments, the Director General of CERN, Rolf Heuer, asked on July 4, 2012:

"I think we have it, do you agree?"

"We have now found the missing cornerstone of particle physics. We have a discovery. We have observed a new particle that is consistent with a Higgs boson."

"only possible because of the extraordinary performance of the accelerators, experiments and the

computing grid."

02

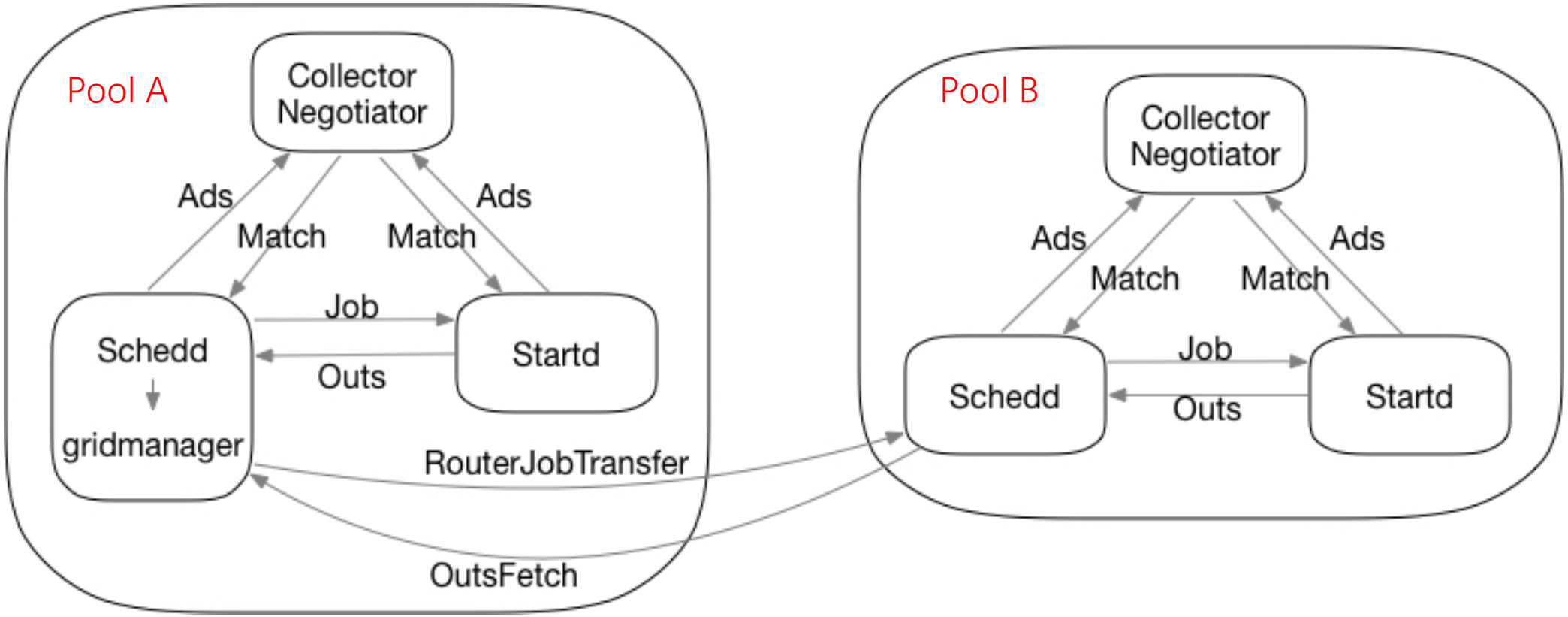
HTCondor整合跨域资源

资源整合

- 2.1 整合多个Condor Pool
- 2.2 软件资源接入
  - ✓分布式文件系统、容器、商业软件云
- 2.3 第三方平台资源
  - ✓国家网格、天津超算工业云平台等

2.1 整合多个Condor Pool

- 通过JobRouter整合多个跨域Condor Pool



2.1 整合多个Condor Pool-RouteTable配置

```
[
  GridResource = "condor job@aliyun217 47.105.54.217";
  name = "site_beihang";
  set_remote_jobuniverse = 5;
  requirements = target.LhaasoNoSFS is True;
  ##delete_WantJobRouter = true;
]
[
  GridResource = "condor ZJUT_HTCONDOR_CM@zjut 139.196.212.96";
  name = "site_zjut";
  set_remote_jobuniverse = 5;
  requirements = target.TOZJUT is True;
  ##delete_WantJobRouter = true;
]
[
  GridResource = "condor my_condor_schedd@ali-cloud 39.107.94.153";
  name = "site_uestc";
  set_remote_jobuniverse = 5;
  requirements = target.TOUESTC is True;
  ##delete_WantJobRouter = true;
]
```

北航Pool

浙工大Pool

成电Pool

2.1 整合多个Condor Pool-网络配置

- Server端
  - ✓对有防火墙或局域网的condor pool，配置CCB Server
  - ✓CCB Server可以和Collector相同
  - ✓MAX_FILE_DESCRIPTORS=10000
- Client端
  - ✓CCB_ADDRESS=$(COLLECTOR_HOST)
  - ✓PRIVATE_NETWORK_NAME=your.private.domian(此配置可以使在相同局域网内的通信不通过CCBserver）

2.1 整合多个Condor Pool-作业属性配置

- 作业转发时匹配合适的CondorPool
  - ✓ Requirements= (TARGET.HAS_GPU is True) AND (TARGET.HAS_CVMFS is True)
- CondorPool有选择的接受自己可以执行的作业
  - ✓ HAS_GPU = true
  - ✓ HAS_CVMFS = true
  - ✓ STARTD_ATTRS = HAS_CVMFS,HAS_GPU,$(STARTD_ATTRS)

2.1 整合多个Condor Pool-安全配置

- 在Daemon进行通信时，配置是否认证
  - ✓SEC_DAEMONNAME_NEGOTIATION = REQUIRED|OPTIONAL|NEVER
- 进行认证采用何种认证方式
  - ✓SEC_DAEMON_AUTHORIZE_AUTHENTICATION_METHODS=CLAIMTOBE, SSL, KERBEROS
    - SSL：使用证书认证
    - KERBEROS ：通过Kerberos协议认证
    - CLAIMTOBE ：不做认证，用于测试

2.1 整合多个Condor Pool-安全配置（续）

❑SSL认证方式

- ✓制作CA证书
- ✓配置SSL认证证书
  - AUTH_SSL_CLIENT_CAFILE = /home/condor/signing-ca-1.pem
  - AUTH_SSL_CLIENT_CERTFILE = /home/condor/host_omega.pem
  - AUTH_SSL_CLIENT_KEYFILE = /home/condor/host_omega.key
  - AUTH_SSL_SERVER_CAFILE = /home/condor/signing-ca-1.pem
  - AUTH_SSL_SERVER_CERTFILE = /home/condor/host_omega.pem
  - AUTH_SSL_SERVER_KEYFILE = /home/condor/host_omega.ke

2.1 整合多个Condor Pool-用户映射

- Unified User Map用户映射文件
  - ✓配置CERTIFICATE_MAPFILE指向一个文件，文件中配置各类认证方式下用户的映射方式，
    - SSL (.*) ssl@unmapped
    - KERBEROS ([^/]*)/?[^@]*@(.*) \1@\2
    - CLAIMTOBE (.*) \1

2.2 软件资源接入

- 科学计算的工具、软件、计算库种类繁多，部署困难
- 容器技术
  - ✓容器可以将复杂的应用环境及程序打包，然后移植到其他的平台上无缝运行
  - ✓对于不同应用背景的Condor Pool，通过容器的使用，能方便的运行其他类的应用
  - ✓HTCondor支持Docker、Singularity容器

2.2 软件资源接入：Docker配置

- 计算节点
  - ✓配置Docker路径
    - DOCKER = /usr/bin/docker
- 作业文件
  - ✓表示此作业要在docker容器中运行
    - +WantDocker = True
  - ✓指定使用的docker镜像
    - +DockerImage = “/cvmfs/containers.ihep.ac.cn/docker/scientificlinux/sl:7”
  - ✓universe=docker

2.2 软件资源接入： Singularity配置

❑计算节点

- ✓配置Singularity路径
  - SINGULARITY = /usr/local/bin/singularity

❑作业文件

- ✓表示此作业要在有Singularity的节点执行
  - requirements=TARGET.HASSINGULARITY==true
- ✓指定使用的镜像
  - +SingularityImage = “/cvmfs/containers.ihep.ac.cn/singularity/scientificlinux/s1.sif”

2.2 软件资源接入：文件共享

❑分布式文件系统CVMFS

- ✓镜像文件发布在CVMFS上
- ✓计算节点安装CVMFS
  - cd /cvmfs/containers.ihep.ac.cn/
- ✓计算节点配置相关属性属性，通知Condor已安装CVMFS
  - HAS_CVMFS = true
  - STARTD_ATTRS = HAS_CVMFS, $(STARTD_ATTRS)
- ✓作业classAd及job router要求目标节点可访问/cvmfs
  - requirements=Target.HAS_CVMFS==true
- ✓选择镜像文件时，通过/cvmfs路径，执行节点可以获取镜像

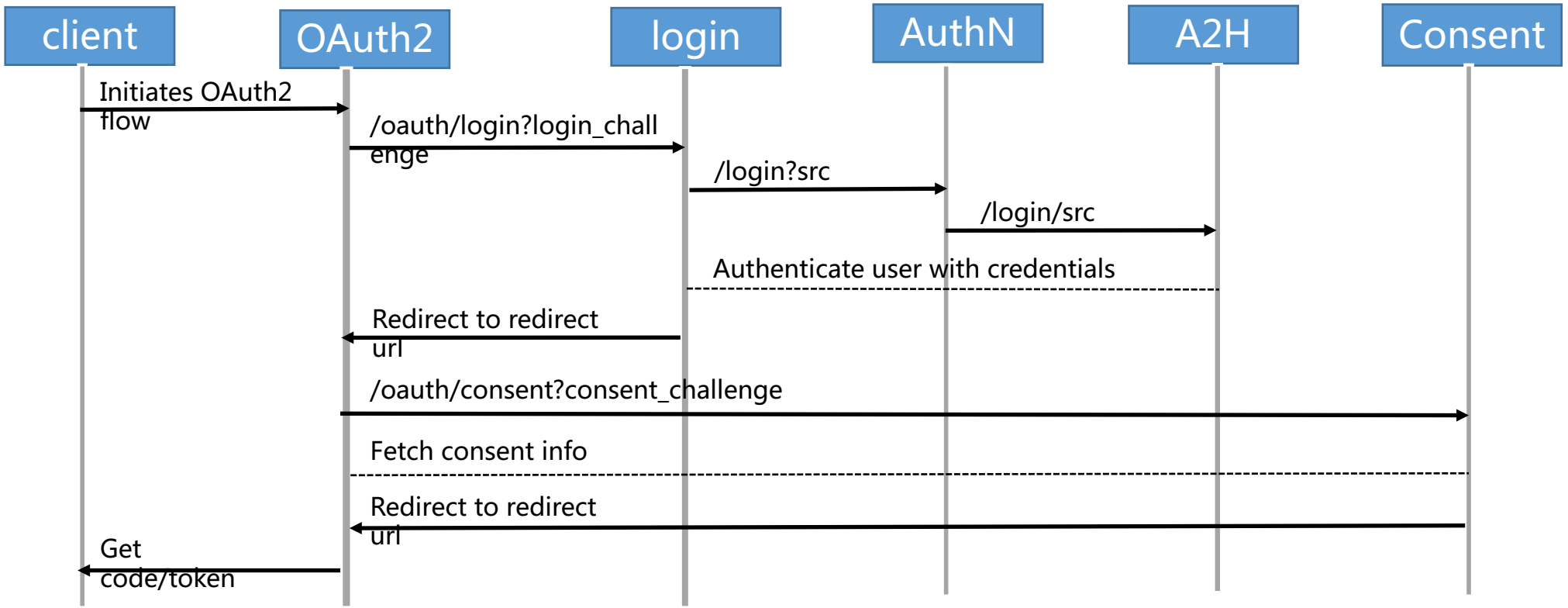
2.3 第三方计算平台接入

❑CNGI国家网格

- ❑向第三方平台或社区提供对应的appid和app secret，以及redirect url
- ❑通过联盟系统访问第三方平台时，使用oauth协议，提供本系统认证的token供平台或社区验证
- ❑验证通过后，第三方平台对账号分配权限

2.3 第三方平台接入（续）

❑CNGI认证流程

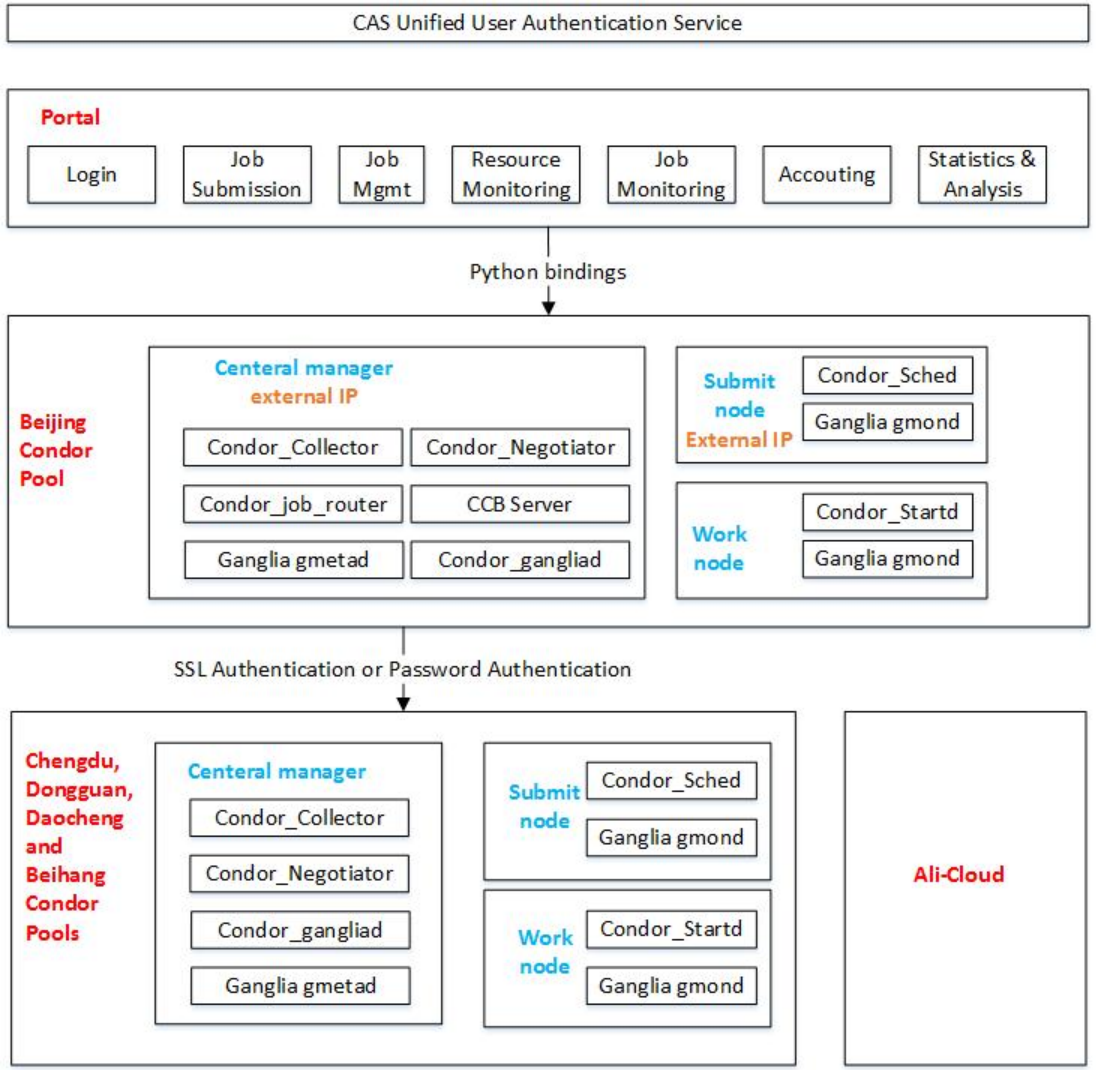


03

LHAASO计算平台实践

3.1 LHAASO计算平台-架构

- 采用HTCondor技术将分布在各地的计算资源联起来，统一管理使用
- 统一在中科院登录节点向平台提交任务
- 根据各pool的负载、软件能力自动分发作业到各地site
- 通过CCB技术解决内网pool的互联互通
- 连接阿里云计算资源
- 通过Graphite和Grafana搭建监控系统，监控和展示平台的硬件资源、计算任务、记时信息。



3.2 LHAASO计算平台-资源和作业情况

- 目前已经接入高能所北京Pool、高能所成都Pool、高能所稻城Pool、北航Pool、成都电子科技大学、浙江工业大学六个站点资源，共计约1500核
- 通过Docker和Singularity容器规避了高能所软件部署的复杂度
- 通过CVMFS分布式文件系统实现了文件的跨域共享
- 完成测试、实际作业共计200多个

3.3 应用案例：分波计算作业

- 通过singularity容器，运行一个GPU程序，通过不同的输入，输出波形文件及数据文件
- 镜像大小2.8G，发布在/cvmfs文件系统上

```
-rw-r--r-- 1 cvmfs cvmfs 2806980608 Nov  2 14:16 gpuwa_v1.2.sif
```

- 程序要求节点使用GPU
- 在执行节点以-writeable，--nv参数运行镜像

```
singularity -d exec --writable  --home `pwd`:/srv --pwd /srv --nv $image /bin/sh run.sh
```

3.3 应用案例：输入文件

Sub文件

```
universe = vanilla
executable = setupenv.sh
transfer_input_files = run.sh
should_transfer_files = YES
when_to_transfer_output = ON_EXIT
output = $(cluster).$(Process).out
error = $(cluster).$(Process).err
log = $(cluster).$(Process).log
queue
```

容器内部执行文件

```
echo "======START RUN FENBO ANALYZE PROGRAMM====="
echo $GPUPWA_BINFILE
ln -sf /data/gputest_epp_1/GammaKK/binfiles $(pwd)/binfiles
cp /data/test/files.txt .
cp /data/test/para.inp .
cp /data/test/res.inp .
ls -la
gammakk files.txt
cat .job.ad
cluster=`cat .job.ad | grep GlobalJobId | cut -d '#' -f 2 | cu
echo $cluster
touch $cluster.ps
mv testout1.ps $cluster.ps
echo "======END ANALYZE====="
```

3.3 应用案例：运行日志及文件输出


```
Active waves: 4
Startup: 3770000 ticks = 3.77 s
MC: 130000 ticks = 0.13 s
Lookup                                    = 0.03 s
Number                                    le : 4
Reading MC Integral File :0 ticks = 0 s
Memory Allocation :0 ticks = 0 s


**********************************************************
Gamma KK Analysis
With     100000 Data Events,
         500000 Generated MC Events and
         500000 Accepted MC Events.
**********************************************************
      Fit coverged

      Minimum Likelihood: -38180.1

      Estimated Distance to Minimum: -2.25393e-06

**********************************************************
Gamma KK Analysis
Using the following partial waves:
          Wave Name  Magn. in Magn. out  Phase in Phase outDynamic inDynamic outDynamic inDynamic out
-----------------------------------------------------------------------------------------------------
                f20         1  0.957243         1   (fixed)     2.001   (fixed)     0.133   (fixed)
                f21      0.03 0.0303597         1   (fixed)     2.001   (fixed)     0.133   (fixed)
                f22       0.2  0.202576         1   (fixed)     2.001   (fixed)     0.133   (fixed)
                 f0      2.26   1.99345      1.11  0.955816      2.15   (fixed)    0.0486   (fixed)
-----------------------------------------------------------------------------------------------------
```

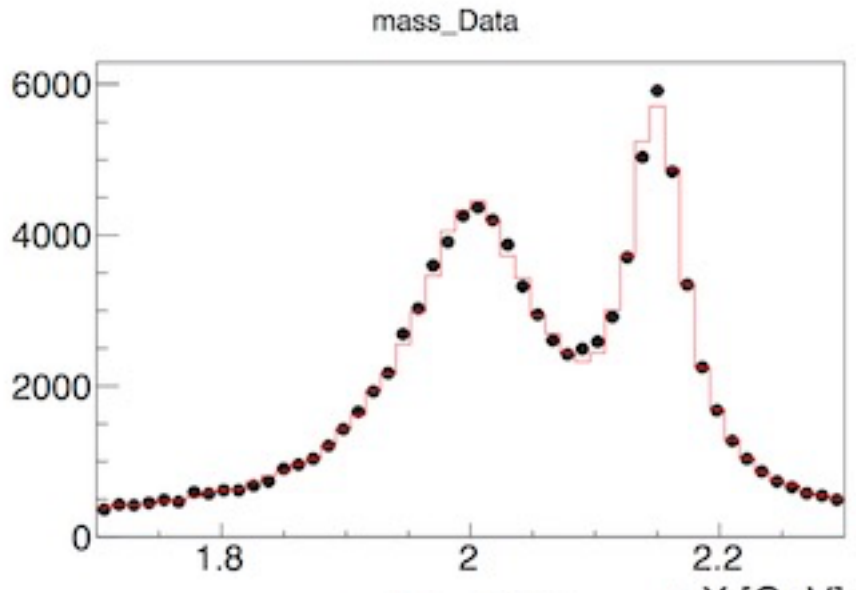
Customize and control Google Chrome. Update is available.

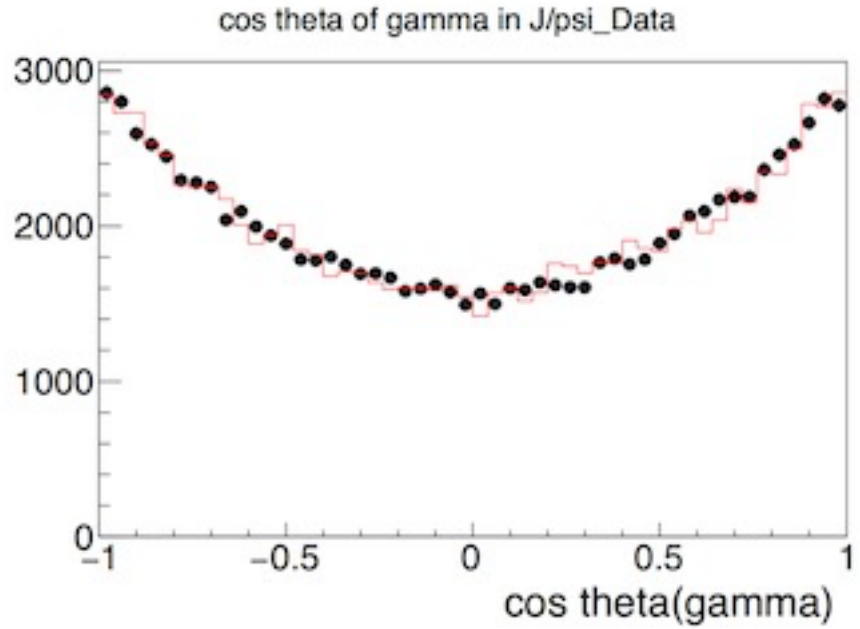
运行日志

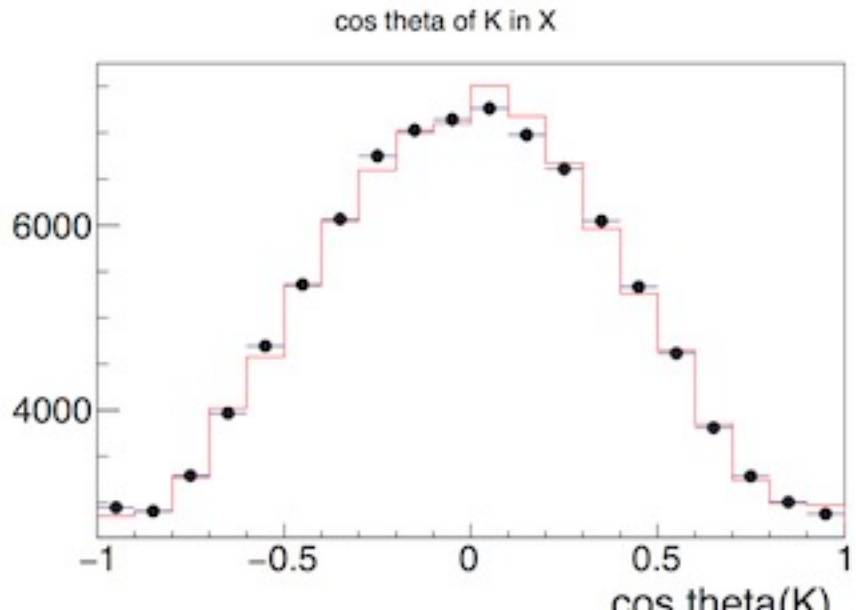
```
1930.0.log
1930.0.out
1930.ps
files.txt
fitresult_GammaKKAnalysis_0001.txt
GammaKKAnalysis_Amplitude_MC_Integral
GammaKKAnalysis_counter.cnt
gmon.out
para.inp
para_input_GammaKKAnalysis_0001.txt
para_output_GammaKKAnalysis_0001.txt
res.inp
```

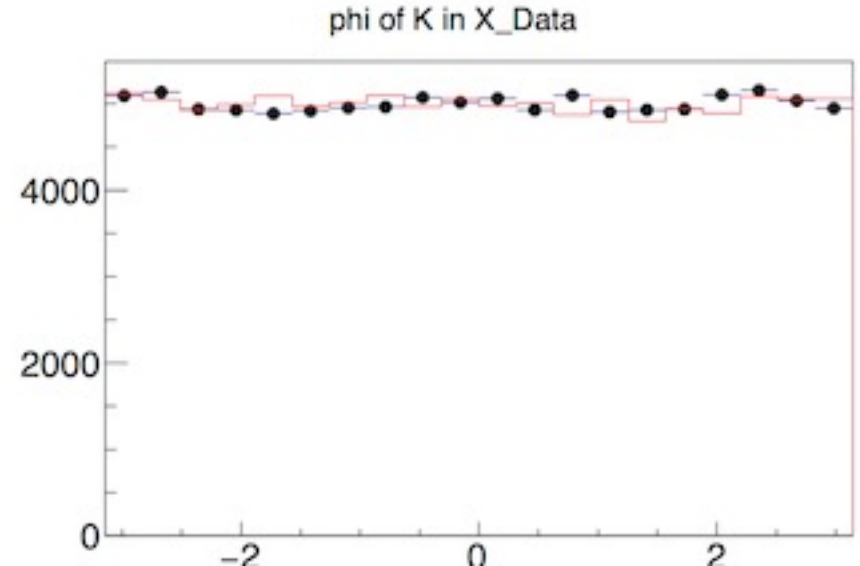
输出文件

3.3 应用案例：波形输出


mass_Data
6000
4000
2000
0
1.8
2
2.2

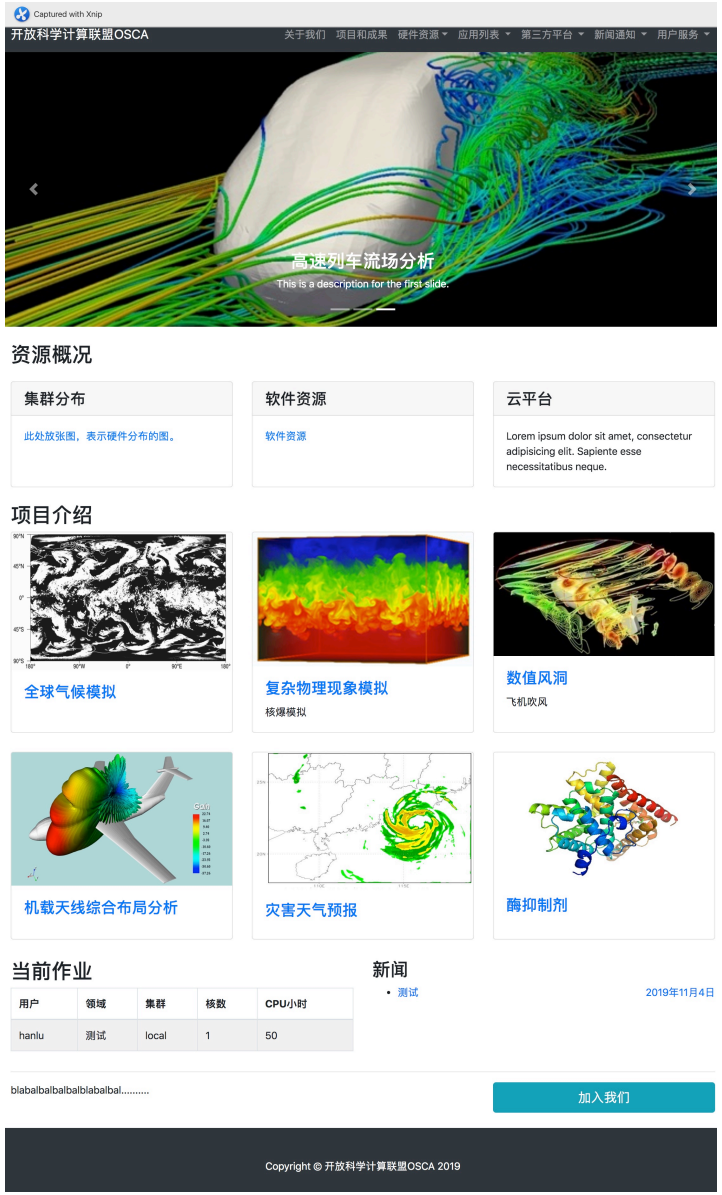

cos theta of gamma in J/psi_Data
3000
2000
1000
0
−1
−0.5
0
0.5
1
cos theta(gamma)


cos theta of K in X
6000
4000
−1
−0.5
0
0.5
1
cos theta(K)


phi of K in X_Data
4000
2000
0
−2
0
2

3.4 Portal建设

- ❑资源统计
- ❑用量统计
- ❑应用服务信息
- ❑项目信息
- ❑第三方平台认证
- ❑用户注册登录
- ❑用户Portal提交作业



3.4 Portal建设：展示硬件资源信息

- 资源概况统计
  - ✓ 各类硬件指标总量：GPU核数、CPU核数、Memory，Disk等
  - ✓ 按地区分布的集群情况
- 资源使用统计
  - ✓ 当前及历史用量
    - 各Pool的节点、核数运行状况
    - 各Pool的作业队列状态
  - ✓ 累计用量
    - 各Pool的作业运行的CPU小时统计

3.4 Portal建设： Portal用户后台

- 应用详情页
  - ✓介绍应用的基本信息
  - ✓提供应用使用接口

3.4 Portal建设： 作业提交表单

- Case by Case
- 对每一种作业提供对用的提交form
- 不是所有作业都能通过portal提交

04 下一步工作

4 下一步工作

- 接入更多的计算资源以及复杂Pool路由机制
- 接入更多类型的计算资源：更多超算平台，阿里云，Slurm，PBS等各类资源
- 执行更多领域和类型的复杂计算作业，例如DAG，MPI等作业
- 完善平台的监控系统和Portal建设
  - ✓ 加强Portal提交作业的通用性和安全性
  - ✓ 统计用户在不同pool的资源用量
  - ✓ 开发计费功能

谢谢！请批评指正！

凯习信息
CACHEINFO

杜治高　2019.11