



竞赛手册

Competition Manual

健雄科协出品

Produced by ASTC

ASC竞赛

体验算法的极值

世界最大规模的超算竞赛

Latest Computing Architecture 	Fastest Supercomputer for Competition Platform 	 Toughest Application Tasks
8,500+ Young Talents Universities 350+ 6 Continents	 ASC Intelligent Computing	\$40,000 Amazing Rewards 
1 Week Transcultural Experience 	 Global Media Echo	Supercomputing Competition Books Published 2

竞赛介绍

Tournament Overview

ASC世界大学生超级计算机竞赛（ASC Student Supercomputer Challenge，简称ASC超算竞赛），肇始发轫于2012年，是世界最大规模的超算竞赛，与德国ISC、美国SC并称世界三大超算竞赛。

该由中国倡议成立，与日本、俄罗斯、韩国、新加坡、泰国、中国台湾、中国香港等国家和地区的超算专家和机构共同发起并组织，并得到美国、欧洲等国家地区超算学者和组织的积极响应支持。ASC旨在通过大赛的平台，推动各国及地区间超算青年人才交流和培养，提升超算应用水平和研发能力，发挥超算的科技驱动力，促进科技与产业创新。

竞赛规则

Introduction to the rules

对于ASC这样的范围广、耗时长的大型比赛，了解清楚竞赛规则可以帮助我们在竞赛中游刃有余。



健雄科协



竞赛手册

Competition Manual

健雄科协出品

Produced by ASTC

竞赛流程

Competition Process

1. 报名及培训

报名：校内赛以个人报名形式参赛。

培训：协等单位组织将针对高性能计算组织若干次培训，时间地点请关注科协等组织发布的相关内容。团队需要使用脚注、尾注或内联文档记录任何外部信息源，并在这些来源的参考列表或书目中包含适当的引用。

（即任何形式的参考文献都需要进行标注）

2. 校内赛暨国际赛选拔

组委会将根据ASC大赛组委会发布的通知，选择部分初赛题目作为校赛题目，参赛学生于指定时间前提交竞赛文档，校赛组委会将结合培训表现和竞赛文档质量，评选校赛获奖名单，同时组建国际赛初赛队伍成员及后备队员。

3. 国际赛初赛（以国际赛组委会实际发布通知为准）

参考以往经验，国际赛初赛参赛队（5名队员）在规定时间内按命题要求提交文档和测试优化结果，ASC竞赛评审委员会评定，在全球队伍中选拔20支左右队伍进入决赛。

4. 国际赛决赛（以国际赛组委会实际发布通知为准）

入围决赛的队伍集中在决赛地点，现场完成给定要求下的超级计算机系统设计，完成软硬件组装、调试及决赛指定题目，并进行答辩与展示。ASC主办方专家组委会根据各队全部题目的表现，最终决出冠军、亚军、e-prize、一等奖和各单项奖。

赛题类型

Question types

对于任何类型的考试或竞赛，尽可能多的了解考题或赛题的类型都可以帮助我们在考试或比赛前做好充分的准备。ASC采用2+N模式，即两道固定题目，N道实际应用题，一般为2道。

【固定题目方向】

1. 集群的设计与搭建
2. HPL和HPCG标准测试

【应用题的相关方向】

1. 脉冲星搜索
2. 量子计算
3. 自然语言处理
4. 人工智能
5. 云计算
6. 一系列前沿CS知识.....

时间安排（以2021年为例）

Time Schedule (take 2021's schedule as an example)

- 2021年11月-2022年1月为报名阶段，参赛高校需以小组形式报名，每支参赛队包含1名指导老师和5名在校本科生。
- 2022年1-3月为预赛阶段，参赛队按要求提交超算应用优化提案。
- 决赛将于2022年4月到5月间举行，经预赛选拔的中国大陆队伍将参加现场总决赛，中国大陆之外的晋级队伍将通过在线方式参与总决赛，他们将比拼基准测试、人工智能和高性能计算应用赛题，并参加英文答辩，角逐冠亚军、e Prize计算挑战奖、最高计算性能奖等诸项大奖。



竞赛手册

Competition Manual

健雄科协出品

Produced by ASTC

竞赛提示

Competition Tips

咱们的同学可不会打没把握的仗，那么接下来就一起来看看竞赛要做哪些准备、有哪些要注意的地方吧。

竞赛准备工作

Preparations

1. 明确阶段任务

ASC竞赛不同阶段的任务核心和难点有所不同，充分了解不同阶段任务核心和难点，有利于我们更好的应对在竞赛时出现的各种问题。

1.1 校内赛

校内赛的任务核心是独立完成往年旧题、今年新题，并提前练习撰写报告的技能，比如：LaTeX、软件绘图、编排表格等。校内赛虽然是选拔赛，但其难度并不低，尤其是对刚接触像ASC这类入门困难的竞赛的萌新们，环境配置将会成为第一大阻碍；由于校内赛是个人赛，最终队伍尚未组建，缺乏队员交流解惑是需要面对的第二道难关；再加上ASC竞赛的学习曲线陡峭的问题，共同构成了ASC竞赛校内赛的较高难度。

1.2 正式赛初赛

初赛的时间跨度为三个月，任务核心是通过校内赛已有的基础独立完成今年新题，并继续练习撰写报告，此时要注重报告的逻辑，并在报告中附加详细的实验。此时我们的竞争对手就不再是校内的同学，而是来自全球的参赛者，因此初赛的难点主要在于竞争压力大，尤其是在DDL前，备赛强度会达到一定高度，可能需要熬夜备考。但通过校内赛也意味着我们有了更多的资源支持：备赛场地、设备可协商解决、产生的合理费用可以报销、队友扶持并进、专业老师指导，都将成为我们比赛的助力。

1.3 线下赛

线下赛是ASC竞赛的决赛，时间跨度为一周。任务核心就是独立组装超算，并完成全新题目。线下赛的竞争压力会更加巨大，来自其他名校的同学与我们同台竞技，时间紧，任务重，给我们带来极大的挑战，但也同时意味着更加丰富的资源，你会得到足够的经费支持，并且受到全球媒体的采访曝光。

2. 收集整理资料

收集整理建议使用搜索引擎，搜索引擎有一定的推荐顺序，建议优先使用Google、GitHub收集资料，如果在Google、GitHub不能找到你所需的内容，可以尝试知乎、Quora，当然，CSDN、百度也可以作为资料的来源，但建议在其他搜索引擎都找不到你所需的内容时再使用。

3. 竞赛设备准备

复现实验对于我们大多数只有笔记本的同学来说可能略显困难，浪费大量的时间，因此推荐大家购买/租用一个云服务器，可以节省很多时间。

4. 计算软件安装

Configure科学计算软件一般安装过程:

Configure -> make -> make install

4.1 Configure 通常是指定特性，比如—with-thread多线程，是否加入某些库—with-cuda;编译器以及选项通常也是在这里指定，with-icc;使用—help仔细阅读一下提供的options帮助很大。

4.2 Configure 会检查系统的具体配置，比如使用的编译器版本，int型具体是多少字节，然后生成makefile或make.in

4.3 Make install 是把编译生成的二进制包和依赖库复制到指定路径。注意:编译的地方是<workdir>，里面有源码和编译的中间文件。而安装的目的地是自己指定的prefix，默认通常是/usr/bin，没有根目录权限又不修改安装路径的话会报错哦。



竞赛手册

Competition Manual

健雄科协出品

Produced by ASTC

知识入门

Basic Knowledge

1. 资源与教程

Latex

Latex总体自学笔记可参考: LaTeX自学笔记_小v小
白冷的博客-CSDN博客_latex自学

(https://blog.csdn.net/weixin_43589465/article/details/107115951)

模板网站: LaTeX Templates - Home
(<http://www.latextemplates.com/>)

LaTeX 工作室 (latexstudio.net)
(<http://www.latexstudio.net/>)

Latex插图排版可参考: LATEX2e插图指南 - 搜索结果 - 知乎(zhihu.com)

LaTeX 各种命令, 符号可参考: LaTeX 各种命令,
符号_GarfieldEr007的专栏-CSDN博客_latex符号
(<https://blog.csdn.net/garfielder007/article/details/51646604>)

超级计算机

中山大学超算队GitHub
(<https://github.com/SYSU-SCC>)

OpenMP

OpenMP 官方教程: OpenMP in Visual C++ |
Microsoft Docs (<https://docs.microsoft.com/en-us/cpp/parallel/openmp/openmp-in-visual-cpp?redirectedfrom=MSDN&view=msvc-170>)

Linux

Linux自学:<https://www.runoob.com/linux/linux-tutorial.html>

Linux yum 命令: <https://www.runoob.com/linux/linux-yum.html>

Linux apt 命令:<https://www.runoob.com/linux/linux-comm-apt.html>

Linux vi/vim: <https://www.runoob.com/linux/linux-vim.html>

Linux命令大全:<https://www.runoob.com/linux/linux-command-manual.html>

Linux下集群的搭建:Linux下集群的搭建_服务器应用_Linux公社-Linux系统门户网站

(<https://www.linuxidc.com/Linux/2014-09/106963.htm>)

Keepalived高可用集群搭建: Keepalived高可用集群搭建_服务器应用_Linux公社-Linux系统门户网站
(linuxidc.com)(<http://www.linuxidc.com/Linux/2014-09/106965.htm>)

Linux下搭建SVN服务器:

Linux下搭建SVN服务器_sunnyqkl 的博客-CSDN博客
(<https://blog.csdn.net/sunnyqkl/article/details/90714763>)

Linux环境变量配置全攻略:

Linux环境变量配置全攻略 - 悠悠i - 博客园
(<https://www.cnblogs.com/youyoui/p/10680329.html>)

Linux常见问题及解决方案 - fengMisaka - 博客园
(<https://www.cnblogs.com/linuxAndMcu/p/10767351.html>)

linux基础知识的总结: linux基础知识的总结-linux _ player_c-ChinaUnix博客

(<http://blog.chinaunix.net/uid-29622064-id-4523016.html>)

其余相关问题以及高级技巧: Linux杂谈 (一): 软链接与硬链接_linux_player_c-ChinaUnix博客

(<http://blog.chinaunix.net/uid/29622064.html>)



竞赛手册

Competition Manual

健雄科协出品

Produced by ASTC

其他

ASC21的初赛资料:

<https://github.com/huangyifan0501/20-21-SEU-ASC>

集群资源: 学校会统一发布

2. 自学提示

2.1 目标设置清晰

在准备竞赛的过程中, 清晰的学习目标可以使我们的学习效率大大提高。根据学长学姐的分享, 以下列出准备几个相关内容时的学习目标和要思考的点, 仅供大家参考。

(1) 学习OpenMP:

什么是OpenMP? 有什么作用?

如何使用OpenMP来加速我的代码? 在Linux环境下, 在Windows环境下如何配置?

我该在哪并行? 如何选择并行区域?

什么是数据依赖? 什么是数据冲突? 如何解决?

什么是原子操作? 为什么需要原子操作?

该选多少线程来运行呢? 线程数量越多越好吗?

代码运行正确吗? 如何检验优化后代码运行正确性?

代码优化成功了吗? 加速比如何计算?

OpenMP实战: 多线程矩阵乘法, 矩阵分块乘法。

(2) 学习MPI:

什么是MPI?

进程和线程的区别是什么?

什么是主进程master, 什么是从进程slave?

什么是进程间通信? 为什么进程间需要通信? 如何进行进程间通信?

如何发送数据和接收数据? 都有哪些方法? 每种方

法之间又何不同?

什么是同步? 为什么需要同步?

什么是死锁? 如何避免死锁?

我该如何编译和运行MPI程序?

为什么我在一个节点上运行这么多进程和我在多个节点上运行这么多进程时间不一样?

MPI实战: 矩阵乘法, 矩阵分块乘法。

(3) 学会使用高性能集群 (Linux命令行):

如何远程登录 (SSH)? 节点之间如何无密码访问?

集群怎样运作? 什么是管理节点? 什么是计算节点?

为何所有节点都能看到同样的目录? 什么是共享存储?

了解你的集群。集群拓扑结构是怎样的? 配置是什么? 使用的是什么网络?

什么是环境变量? 有什么作用? 什么是程序入口参数?

Linux基本命令。如何上传和下载文件? 为何不能用apt-get或yum安装软件? 什么是Linux权限?

如何从源码编译? 什么是软件依赖? 什么是动态链接库? 什么是动态链接? 什么是静态链接?

什么是编译器? 什么是编译工具链? 什么是make和cmake? 如何编写MakeFile和CMakeList?

编译报错怎么办? 系统自带的软件版本太旧怎么办?

如何提交作业? PBS作业调度系统如何使用?

什么是脚本? 掌握Shell脚本的使用。

(4) 学习程序性能分析:

IntelParallel Studio XE的使用

如何测试程序性能? 什么是热点分析?

程序性能分析报告怎么看?



竞赛手册

Competition Manual

健雄科协出品

Produced by ASTC

(5) 性能优化策略:

并行度优化; 内存管理优化; 数据传输优化;
存储器访问优化; 向量化优化; 负载均衡优化;
多线程扩展性优化

2.2 重视代码阅读

阅读代码可以帮助我们训练思维, 熟悉代码, 在我们自己写代码时就可以做到事半功倍。而阅读代码本身是一件枯燥乏味的事情, 所以建议大家借助可视化工具, 如Doxygen+Graphviz, 通过思维导图的形式理解代码的逻辑。Doxygen是一种开源跨平台的, 以类似JavaDoc风格描述的文档系统, 完全支持C、C++、Java、Objective-C和IDL语言, 部分支持PHP、C#。注释的语法与Qt-Doc、KDoc和JavaDoc兼容。Doxygen可以从一套归档源文件开始, 生成HTML格式的在线类浏览器, 或离线的LATEX、RTF参考手册。Graphviz的是AT&T Labs Research开发的图形绘制工具, 他可以很方便的用来绘制结构化的图形网络, 支持多种格式输出, 生成图片的质量和速度都不错。Graphviz本身是开源的产品, 下载可以到 [这里](#), 以及他的演示界面 Graphviz在windows上和Linux上都可以顺利运行。

2.3 学长的其他建议

完全没有接触过Linux的话, 只能说你就上手去做, 然后在做的过程, 你可以看各种各样的博客, 可以看各种各样别人的教程, 然后边做边去学, 然后把这些就是这种非常基础的东西, 自己不断摸索。

可以学习一下华为的内部的Linux教学手册。

上手的早期就直接把图形界面关掉, 直接上在终端或者是直接去文字化的这种方式去操作, 这样可能对你后期比较有好处, 不然这个长时间在图形界面下, 对后期的操作可能会造成影响。

ASC是竞赛不是课堂回答问题, 没有标准答案,

需要自己根据分析上手操作寻找方向。

相关知识

Further Related Knowledge

高性能计算平台

High Performance Computing

1. 含义

高性能计算平台又称高性能计算集群, 简称HPC (High Performance Computing), 是一套计算性能强大, 数据传输率超高, 具有大规模存储空间和完整软件的系统, 通常包含计算处理单元、并行存储单元、网络单元、管理调度软件单元, 主要采用集群架构, 通过网络将大量的服务器连接起来, 让所有服务器协调工作来完成一系列计算任务。

2. 主要作用

主要应对的问题是缩短超大规模复杂计算任务的时间和扩大求解规模;

3. 应用方向

(1) 计算密集型应用: 大型科学与工程计算, 数值模拟等; 应用领域包括石油、气象、CAD/CAE、生命科学、核能、制药、环境监测分析

(2) 数据密集型应用: 数字图书馆, 数据仓库, 数据挖掘, 计算可视化等; 应用领域包括图书馆、银行、证券、税务、决策支持系统等

(3) 通信密集型应用: 协同工作、网络计算、遥控和远程诊断等; 应用领域包括网站、信息中心、搜索引擎、电信、流媒体等



4. 典型的案例

石油勘探行业已经由原来基于地质学方法的钻井探索方式改变为更先进的地震波法，所谓地震波法就是在地面用炸药激起人工地震波，碰到不同岩层会形成不同的反射波，反射波经接收器收集转变为电子信号数据，经过数据分析模拟还原地下地质构造。

5. 性能衡量主要指标

每秒钟双精度浮点计算次数，XX万亿次/s，XX Tflops的写法意思就是XX万亿次/s。

6. 基本组成构架

(1) 计算节点：一个高性能、高读写率、高传输率、高可靠性、高稳定性的计算单元，通常为节约空间常选择刀片服务器，整个计算节点采用Cluster集群架构，通过高速InfiniBand计算网络连接服务器计算资源，通过千兆以太网协同调度、管理整个系统。

(2) 网络系统：一般分为计算网络和管理网络，计算网络采用InfiniBand网络，高带宽、低延迟，可达到100G/s以上的传输率；管理网络采用万兆或千兆以太网网络，用户通过资源调度管理软件管理整个高性能计算平台，用户通过网络远程访问高性能计算平台，获取高性能计算平台的计算资源进行求解运算。

(3) 数据并行存储系统：主要是用来提供高带宽的存储容量，缓解计算任务对临时文件的高速存储和读取需求。

(4) 集群调度管理系统：通常采用B/S架构，通过浏览器进行操作。可以管理集群系统中的软硬件资源和用户提交的作业，根据集群中的资源使用情况来合理的调度用户提交的作业，从而达到提高资源利用率和作业执行效率的目的。

7. GPU在高性能计算中的优势

GPU设计初充为高性能三维图形应用，具有强大的计算能力和很高的存储带宽，而这两点对于高性能三维图形应用是至关重要的。GPU通常具有128位或256位的内存位宽，直接表明在一个时钟周期内，传输的数据要远大于CPU，CPU的内存位宽目前通常为32位和64位。此外GPU包含大量的流处理器（核），在并行计算能力上更远大于CPU。

Linux

Linux Operating System

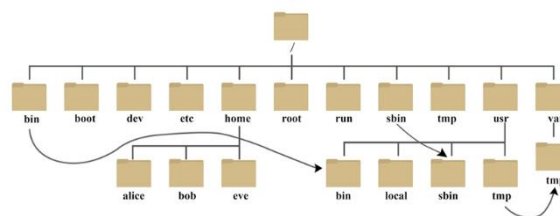
1. 简介：Linux 内核最初只是由芬兰人林纳斯·托瓦兹（Linus Torvalds）在赫尔辛基大学上学时出于个人爱好而编写的。Linux 是一套免费使用和自由传播的类 Unix 操作系统，是一个基于 POSIX 和 UNIX 的多用户、多任务、支持多线程和多 CPU 的操作系统。Linux 能运行主要的 UNIX 工具软件、应用程序和网络协议。它支持 32 位和 64 位硬件。Linux 继承了 Unix 以网络为核心的设计思想，是一个性能稳定的多用户网络操作系统。

2. 系统目录：登录系统后，在当前命令窗口下输入命令：ls /

你会看到如下图所示：

```
[root@localhost ~]# ls /
bin  dev  home  lost+found  mnt  proc  sbin  srv  tmp  var
boot  etc  lib  media  opt  root  selinux  sys  usr
```

树状目录结构：





竞赛手册

Competition Manual

健雄科协出品

Produced by ASTC



以下是对这些目录的解释：

/bin：bin 是 Binaries (二进制文件) 的缩写, 这个目录存放着最经常使用的命令。

/boot：这里存放的是启动 Linux 时使用的一些核心文件, 包括一些连接文件以及镜像文件。

/dev：dev 是 Device(设备) 的缩写, 该目录下存放的是 Linux 的外部设备, 在 Linux 中访问设备的方式和访问文件的方式是相同的。

/etc：etc 是 Etcetera(等等) 的缩写, 这个目录用来存放所有的系统管理所需要的配置文件和子目录。

/home：用户的主目录, 在 Linux 中, 每个用户都有一个自己的目录, 一般该目录名是以用户的账号命名的, 如上图中的 alice、bob 和 eve。

/lib：lib 是 Library(库) 的缩写, 这个目录里存放着系统最基本的动态连接共享库, 其作用类似于 Windows 里的 DLL 文件。几乎所有的应用程序都需要用到这些共享库。

/lost+found：这个目录一般情况下是空的, 当系统非法关机后, 这里就存放了一些文件。

/media：linux 系统会自动识别一些设备, 例如U盘、光驱等等, 当识别后, Linux 会把识别的设备挂载到这个目录下。

/proc：proc 是 Processes(进程) 的缩写, /proc 是一种伪文件系统 (也即虚拟文件系统), 存储的是当前内核运行状态的一系列特殊文件, 这个目录是一个虚拟的目录, 它是系统内存的映射, 我们可以通过直接访问这个目录来获取系统信息。这个目录的内容不在硬盘上而是在内存里, 我们也可以直接修改里面的某些文件, 比如可以通过下面的命令来屏蔽主机的ping命令, 使别人无法ping你的机器: `echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all`

/mnt：系统提供该目录是为了让用户临时挂载别的文件系统的, 我们可以将光驱挂载在 /mnt/ 上, 然后进入该目录就可以查看光驱里的内容了。

/opt：opt 是 optional(可选) 的缩写, 这是给主机额外安装软件所摆放的目录。比如你安装一个ORACLE数据库则就可以放到这个目录下。默认是空的。

/root：该目录为系统管理员, 也称作超级权限者的用户主目录。

/sbin：s 就是 Super User 的意思, 是 Superuser Binaries (超级用户的二进制文件) 的缩写, 这里存放的是系统管理员使用的系统管理程序。**/selinux**：这个目录是 Redhat/CentOS 所特有的目录, Selinux 是一个安全机制, 类似于 windows 的防火墙, 但是这套机制比较复杂, 这个目录就是存放selinux相关的文件的。

/srv：该目录存放一些服务启动之后需要提取的数据。

/sys：这是 Linux2.6 内核的一个很大的变化。该目录下安装了 2.6 内核中新出现的一个文件系统 sysfs。sysfs 文件系统集成了下面3种文件系统的信息：针对进程信息的 proc 文件系统、针对设备的 devfs 文件系统以及针对伪终端的 devpts 文件系统。

该文件系统是内核设备树的一个直观反映。当一个内核对象被创建的时候, 对应的文件和目录也在内核对象子系统中被创建。

/tmp：tmp 是 temporary(临时) 的缩写, 这个目录是用来存放一些临时文件的。

/usr：usr 是 unix shared resources(共享资源) 的缩写, 这是一个非常重要的目录, 用户的很多应用程序和文件都放在这个目录下, 类似于 windows 下的 program files 目录。

/usr/bin：系统用户使用的应用程序。

/usr/sbin：超级用户使用的比较高级的管理程序和系统守护程序。



竞赛手册

Competition Manual

健雄科协出品

Produced by ASTC



/usr/src: 内核源代码默认的放置目录。

/var: var 是 variable(变量) 的缩写, 这个目录中存放着在不断扩充着的东西, 我们习惯将那些经常被修改的目录放在这个目录下。包括各种日志文件。

/run: 是一个临时文件系统, 存储系统启动以来的信息。当系统重启时, 这个目录下的文件应该被删掉或清除。如果你的系统上有 /var/run 目录, 应该让它指向 run。

其他相关名词

Others

MPI:

Message Passing Interface是一个并行计算的API, 适合超级电脑, 大规模集群。

OpenMPI:

OpenMPI是一种高性能消息传递库, 可以很方便的把串行程序, 改为多线程并行程序, 适合多核心电脑, 可以和MPI搭配使用, 对C语言和Fortran高性能计算支持很好。

TBB:

Intel Threading Building Blocks 线程构建模块, 是Intel公司开发的并行编程开发的工具, 能很好的支持C++并行计算编程。

CUDA:

Compute Unified Device Architecture, 显卡厂商NVIDIA推出的运算平台, 使用显卡和处理器进行异构并行计算, 是比较高级的并行开发工具, 不需要太多硬件知识。

OpenCL:

Open Computing Language 是一个为异构平台编写程序的框架, 属于API, 和OpenGL架构类似, 此异构平台可由CPU, GPU或其他类型的处理器组成。

