基于命令行客户端的网格软件 SCE 设计与实现①

龙　斌[1,2]　迟学斌[1]　肖海力[1]　(1.中国科学院 计算机网络信息中心 超级计算中心 北京 100190;
2.中国科学院 研究生院 北京 100190)

摘　要：如何高效和方便的使用计算资源是网格计算里迫切需要解决的问题之一。为了解决该问题，本文基于 Linux 命令行，开发一种超级计算环境（SCE）。在 SCE 中，用户可以完成作业的提交、编译、查询和文件下载等工作。SCE 的部署配置操作简单，并具备高度可扩展的特性。通过屏蔽底层计算节点的异构性，使得其在网格计算中适用环境更加广泛。

关键词：网格计算；中间件；命令行客户端

Design and Implementation of Grid Software SCE Based on Command Client

LONG Bin[1,2], CHI Xue-Bin[1], XIAO Hai-Li[1]

(1. Super Computing Center, Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China; 2.Graduate University, the Chinese Academy of Sciences, Beijing 100190, China)

Abstract: How to make use of computing resource effectively and conveniently is one of urgent topics in grid computing. In order to solve the problem, this paper introduces Super Computing Environment (SCE) based on Linux command line. SCE can do job process such as submission, compiling, query, download file and so on. It has characteristics of deploy simple configuration and high extensibility. Through shielding the bottom of heterogeneous computing nodes, it can be used in more grid computing environments.

Keywords: grid computing; middleware; command client

经过多年的发展和应用，高性能计算技术已经在计算机仿真、石油与天然气、气象、生命科学、计算化学和金融业的突破性研究和日常工作中发挥了重要的作用。网格计算是目前高性能计算技术中最为热门的一个分支，网格计算强调通过整合网格中分散的异构计算资源节点，来提供与地理位置无关、计算设施无关的通用一致性计算能力[1]。

目前网格软件中较为成熟的是 Globus 项目为代表的 Globus Toolkits，该工具包软件是一个构筑网格计算环境的中间件，能够提供基本的资源管理、数据管理、通信和安全等服务[2,3]。

Globus Toolkits 功能全面，但缺乏灵活性和可扩展性，部署起来也相对庞大和繁琐。针对自身的特点和需求，我们自主设计开发了轻量级面向科学计算的超级计算环境(SCE)网格软件。

SCE 是一款基于命令行的网格软件，通过命令行的输入，可以完成作业提交、编译、查询、下载作业文件等工作。采用命令行的操作方式，具有以下优点：①跟传统的 Linux 操作习惯相同，使用起来方便简单；②具有较强的交互性，并可交叉使用 Linux 系统的命令；③对于编译、调试作业等操作具有较好的支持。

1　SCE 的框架设计

1.1 SCE 项目介绍

中国科学院“十一五”信息化规划中部署了建设中国科学院超级计算环境的任务，并据此在院“十一

① 基金项目:国家高技术研究发展计划(863)(2006AA01A117);中国科学院信息化专项(INFO-115-B01)
收稿时间:2009-12-25;收到修改稿时间:2010-01-25

五”信息化专项中设立了“超级计算环境建设与应用”项目，借此形成由院超级计算主中心、分中心、所级计算中心构成、具有总计 200 万亿次/秒以上计算能力的分布式高性能超级计算环境。

SCE 的开发源于建设“中国科学院超级计算环境建设与应用”，希望建立一个能够把各学科计算应用集成到统一的网格环境。

1.2 SCE 的设计思想

根据中科院各研究院所的计算能力和资源需求的差异性，把科学计算网格自上而下分为主中心（顶层）、分中心（中间层）、所级中心（基层）的三层结构。主中心管理整个网格系统，调度全局资源；分中心调度所管辖范围内的资源，服从主节点全局调度；基层的计算资源就近接入所级中心，服从分中心调度。

采用三层结构的设计是对传统计算网格的拓展，有利于以下几个方面：

(1) 环境稳定，单个计算节点或网格服务器出现问题，不影响计算。

(2) 功能划分明确，每一级定位、分工明确。

(3) 区域自治，分中心具有一定管理权限，便于合理安排区域内的共享资源。

(4) 扩展性好，多叉树结构，允许更多节点加入，规模扩大对网格软件承受的负载影响很小，便于软件稳定、可靠运行。

为了达到这样一个设计目标，SCE 采用了 OGSA 的体系架构[4]。OGSA 最基本的思想就是以“服务”为中心，在 OGSA 框架中，将一切抽象为服务，包括各种计算资源、存储资源、网络、程序、数据库等。这种观念，有利于通过统一的标准接口来管理和使用网格[5]。

在 OGSA 框架下，SCE 结合自身特点,把安全服务、信息服务、数据服务、作业管理等各个部分有机地结合在一起,形成完整的网格系统,并根据 I. Foster 等人提出的五层沙漏结构[5,6]组成相适应的 3 层网格基本功能模块。

1.3 SCE 的功能模块

SCE 的功能模块主要分为 3 个层次，客户端、服务层、资源层(图 1)。

客户端：主要执行与网格用户的交互操作，用户在命令行输入命令进行作业提交、编译、查询、下载等操作。命令包括 SCE 作业命令 bsub（作业提交）、scecc(作业编译)、bjobs（作业状态查询）、bqueues（队列查询）、bkill（终止作业）等[7]；还包括 Linux 下使用的系统命令，如 top、ps、mkdir、vim 等。

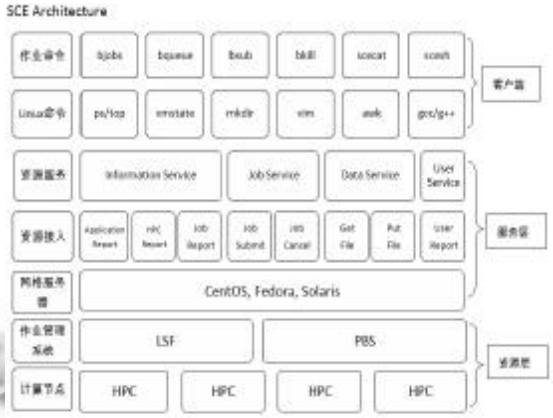

图 1 SCE 层次结构

服务层：通过服务层把客户端层的用户命令与底层的资源相连接。服务层包括资源服务和资源接入两个部分。资源服务主要是响应用户的查询作业、提交作业等情况，通过查询 SCE 数据库完成；资源接入则通过与底层的 HPC 建立连接，把相关的资源信息汇报给 SCE 数据库。

资源层：把用户提交的作业通过服务层转发给资源层的作业管理系统，作业管理系统则负责分发作业到网格内的计算节点。目前支持的作业管理系统有 Platform 公司的 LSF(Load Share Facility)，Cluster Resources 公司的 Torque PBS。

1.4 SCE 的安装部署

SCE 网格内的节点可分为：登录节点、服务节点、计算节点。登录节点，负责网格用户的 SCE 命令请求，检查操作请求是否合法并检查命令参数。为了提高安全性和稳定性，服务节点则分为中央服务器和前端服务器两部分。中央服务器负责与网格登录节点的连接，前端服务器负责与后台的计算节点的连接。计算节点则通过作业管理系统来处理计算请求。SCE 根据节点的不同功能和特点，将部署模块分为 4 个层次的对应模块：Client、Center Server(CS)、Front Server(FS)和 HPC。

由于部署 Client、CS、FS、HPC 模块的机器之间都通过 SSH 建立连接，因此在不同机器机器之间需要先建立基于密钥的安全认证方式连接[8]。

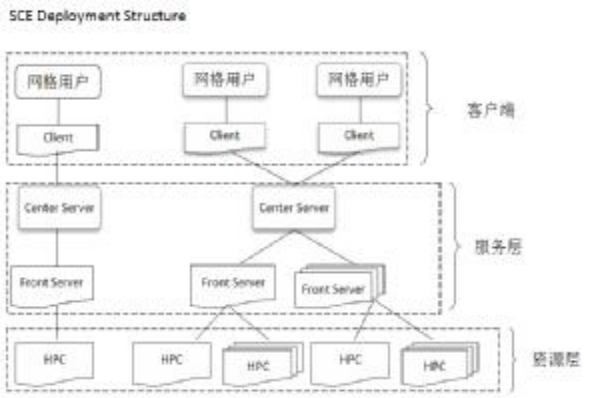

图 2　SCE 部署结构

如图 2 所示，这 4 个类型的模块可以灵活配置到不同的机器，也可以安装在同一台机器。图左半部分代表在一台机器内安装所有 SCE 套件的情形，右半部分代表分别部署不同套件到多台机器的情况。当一台机器部署安装所有套件时，相当于一台机器构建了一个自己的 SCE 网格；同理，当不同套件安装在多台机器时，相当于这些机器集合构建了一个 SCE 网格。

针对科学计算网格的三层结构，SCE 通常会部署到主中心、分中心、所级中心的三层网格环境。其中主中心部署 CS,分中心部署 FS 和 Client，所级中心部署 HPC。网格用户可以就近选择一台 Client 登录进入科学计算网格。由于采取的是模块化部署，单个计算节点或网格服务器出现问题，不影响网格中其它节点的计算。

1.5 SCE 的目录

SCE 的安装目录结构（图 3）和部署方式相同，区别只是将 HPC 模块移入 FS 模块内部。Client、CS、FS 三个目录都包含 bin 和 etc 两部分内容，其中 bin 文件存放具体的可执行文件，etc 文件存放相关安装配置文件；FS 目录包含一个 hpc 文件，存放与计算资源相关的文件，其中包括封装的应用、用户权限表、队列信息等；Portal 目录存放通过浏览器方式登录 SCE 的相关文件内容；Log 目录存放 CS、FS 产生的日志记录。

Client、CS、FS、Portal、Log 几个部分可以分别安装在不同机器，也可集成安装在同一台机器。对于 SCE 的安装部署，主要都是通过修改 etc 中的.conf 文件来完成。

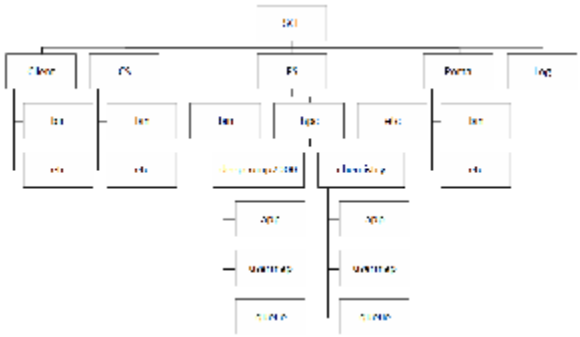

图 3　SCE 目录结构

2　SCE 的作业提交

2.1 bsub 介绍

SCE 的开发是为了能够让网格内的用户方便和高效的使用网格中的各种计算资源。所谓使用计算资源，主要通过提交作业命令来完成。

为了保持与用户的使用习惯相一致，SCE 采用目前广泛使用的 LSF 作业管理系统命令做为 SCE 的命令集。在 LSF 中作业提交命令由 bsub 来完成，故而 SCE 提交作业也命名为 bsub。

2.2 作业提交的流程

一个作业从网格用户提交到最终的完成作业可以划分为 4 个阶段。划分可按照 SCE 部署的模块不同纵向划分，以及按照作业执行流程的顺序横向划分。具体的作业提交处理流程如图 4 所示。

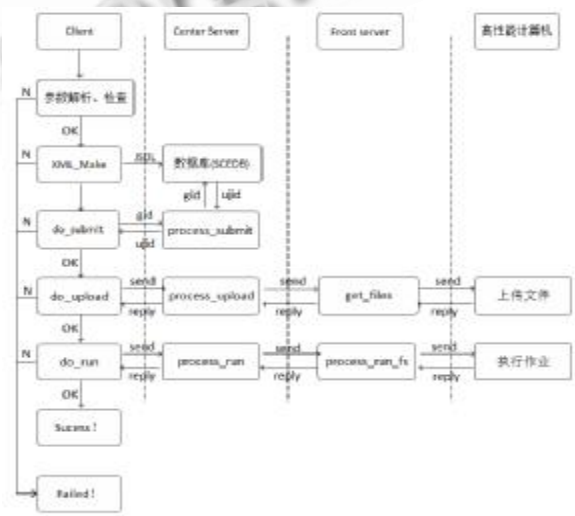

图 4　作业提交框架

根据部署模块的不同，纵向划分可分为 4 个步骤：

(1) Client 处理作业提交请求

(2) CS 接收提交的作业

(3) FS 收集作业信息并转发到相应的计算节点

(4) 作业在计算节点完成计算

根据作业的执行顺序来横向划分，则可分为生成作业 XML、提交作业、上传作业文件、作业运行 4 个环节。

首先，生成作业的 XML 文件。作业命令在 Client 进行参数解析，解析生成的参数传递到 opts_bsub 结构体中，之后进行相关参数检查。参数解析出错会给出相应的错误码，并返回 Failed。符合要求的作业会调用 XML_Make 函数，该函数会把作业转化为标准 JSDL[9]（Job Submit Define Language）格式的 XML 文件，里面包括作业基本信息（作业名），所请求的应用信息（应用名），所需要的资源（CPU 个数、运行时间），以及其它资源信息（如计算节点名、队列名等）。

其次，生成作业的全局作业号 gid(Global ID)和用户作业号 ujid（User Job ID）。do_submit 函数中生成 gid，并把 gid 传入到 Center Server 的 process_submit 函数，该函数会把 gid 插入到 SCEDB 数据库，并从数据库中取出和用户相关联的 ujid 作业号。gid 是 19 位长度的全局唯一标识作业号，ujid 则根据每个用户提交的作业数量情况累加起来得到的作业号。ujid 号相比 gid 短小，方便记忆和作业查询。

然后，上传指定的输入文件。用户提交的作业可能包含有指定的输入文件，此时则调用 do_upload 函数完成上传文件的过程。Client 把上传文件传送到 CS，CS 再把文件传送到 FS，最后由 FS 把该文件再送达最后的高性能计算机。采用层层传递的方式，而不采用点对点的直接传输方式，主要基于安全性的考虑。以上传输的文件通常是指传输小文件（不大于 10M 的文件），对于大文件则通过其它加密方式直接上传到高性能计算机。

最后，在计算节点运行作业。Client 通过 do_run 函数发起运行作业的请求，CS 端接收运行作业请求，在数据库中读取 JSDL 格式的 XML 文件，并解析得到作业基本信息和作业运行使用的资源。CS 端把相关信息和资源传送给 FS，FS 则把相关信息传送给计算节点，计算节点通过调用系统命令完成作业计算。

3 SCE操作演示

3.1 listres 查看科学应用

SCE 网格中，使用 listres 命令来列出网格内所有的资源应用。listres + [应用程序名]，则可以查看一个指定应用程序的详细情况，包括所在的网格、能够提交的 HPC 队列等。gaussian 是进行半经验计算和从头计算使用最广泛的量子化学计算应用，图 5 是查看计算化学的 gaussian 应用程序在 SCE 当前节点中的详细情况。

图 5 listres 查看科学应用

3.2 bqueues 查看队列信息

在提交作业前，先查看各节点的队列状态情况。使用 bqueues 命令来进行操作。图 6 中可以看到合肥 ISSP 节点拥有 serique/paraque/fuque 的作业队列。其中 GRIDNODE 代表网格节点，HPCNAME 代表某一台具体的机器，QUEUE 代表该机器中所包含的队列，WALLTIME 代表作业在该队列内允许运行的最长时间（以分钟为单位），NJOBS 代表队列中所有的作业数，PENDS 代表正在等待的作业数，RUN 代表正在运行的作业数目。

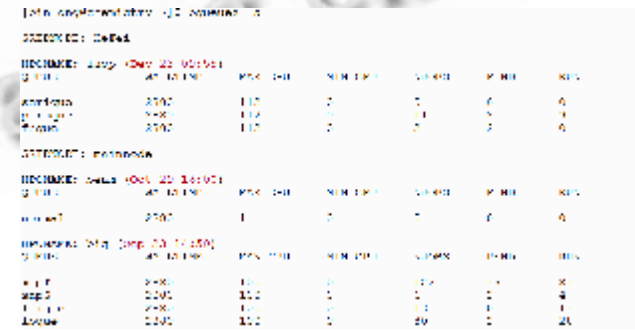

图 6 bqueues 查看队列信息

3.3 bsub 提交作业

bsub 是提交作业的命令，通过指定不同参数指定提交的计算节点、作业队列、CPU 个数等信息。图 7 提交一个 hostname 的应用到 deepcomp7000 机器，只使用 1 个 CPU。该作业提交之后，调度到 scgrid 队列，并给出作业对应的 gid/ujid 号。可以通过

ujid/gid 来查询该作业的相关信息。

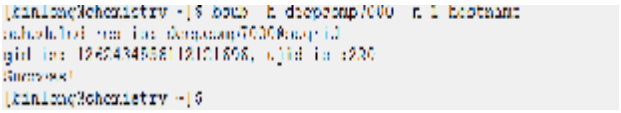

图 7　bsub 提交作业

3.4 bjobs 查看历史作业

bjobs 用来查看历史作业信息。图 8 反映了当前用户提交历史作业，以及各作业提交时间、更新时间、所属队列、完成情况等作业的信息。

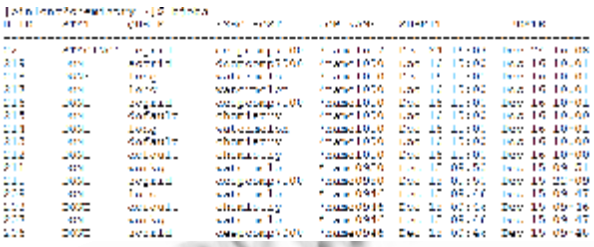

图 8　bjobs 查看历史作业

4　结束语

目前 SCE 已成功部署到中国科学院科学计算网格，通过 SCE 网格中间件可以访问包括中科院百万亿次超级计算机深腾 7000 在内的网格资源，并完成相应的作业提交、编译、查询、下载作业文件等工作。

本文着重介绍了超级计算网格环境中间件 SCE 的设计与实现，并通过 bsub 提交作业的框架来展示 Client、CS、FS、HPC 模块之间的处理流程。SCE 的主要特点是轻量级和可扩展性，部署配置操作简单，可扩展性高。通过屏蔽底层的计算节点的异构性，使得其在网格计算的适用环境更加广泛。

参考文献

1 都志辉,陈渝,刘鹏.网格计算.北京:清华大学出版社, 2002.2－5.

2 Globus Overview of the Grid Security Infrastructure. [2009-7-10]. http://www.globus.org/security/overview.html.

3 Dai ZH, Wu LJ, Xiao HL, eds. A Light weight Grid Middleware Based on OPEN SSH – SCE Grid and Cooperative Computing. GCC 2007. Sixth International Conference on, 2007,16-18:387－394.

4 Ian Foster, Hiro Kishimoto, Andreas Savva, eds. The open grid services architecture,version1.0. informational document.[2005-01-29]. http://www. gridforum. org/documents/GWD－I－E/GFD－I.030.pdf

5 樊宁.网格体系结构概述. [2006-9-11]. http://www.ibm. com/developerworks/cn/grid/gr-fann/.

6 Foster I, Kesselman C. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 1998. Chapter 3.

7 龙斌.SCE Client 用户手册 1.0. [2008-03-20]. http://wiki.scgrid.cn/.

8 Ylonen T, Lonvick C. The Secure Shell (SSH) Protocol Architecture RFC 4251.[2009-6-11]. http://www.snailbook.com/docs /architecture.txt.

9 Anjomshoaa A, Brisard F, Drescher M. Job Submission Description Language (JSDL) Specification, Version 1.0. [2005-11-7]. http://forge.gridforum.org/projects/JSDL-wg.