

嵌入式系统的集群计算

中期报告

2018年4月15日

一 项目信息

项目名称：嵌入式系统的集群计算

院系：电子科学与工程学院

项目成员：高博文 吴沛聪 余加辉 陈建

指导教师：方元

二 项目目标及开发环境

1 项目目标

在嵌入式开发版上搭建可扩展的计算集群，进行集群的功能测试和性能测试。

2 开发环境

开发平台：Windows x64, Linux x86/64, Linux ARM

变成语言及C编译工具链：Shell Script, Python, C, C++; GNU ARM, GNU

Linux

目标平台：搭载s5pv210的x210ii开发板以及其他支持posix协议的系统

软件环境：Linux内核（版本3.0.8），Python 2.7，mpicc/mpi4py

三 研究过程及项目进展

1 集群节点配置

在嵌入式开发板上载入Linux操作系统内核，挂载文件系统，配置相关环境变量。配置远程连接和控制台服务，安装（或者移植）相关软件如mpi，python以便进行集群搭建。为每个节点分配唯一的hostname以便进行整个网络的配置。

在单个开发版上进行单应用多线程测试，以便将来在搭建多机测试时直接分配计算任务。因为需要进行性能测试，我们用python设计了一个时间复杂度为 2^n 的算法，将计算任务设计到单机运行需要80s左右，以便观察将计算任务部署到整个集群后计算速度的提升效果。

进行启动脚本的编辑。我们学习了linux系统的启动过程，在/etc/init.d/中增加启动任务，从/proc/cmdline读入uboot传入的启动参数，从而进行包括hostname在内的一系列配置。

2 网络环境配置和集群搭建

将每个节点接入网络环境，尝试使用telnet/ssh进行多机通信。这个部分十分简单，因为我们使用的开发板上已经有dm9000以太网芯片，我们直接将它们接入网络，然后挂载nfs作为启动系统，执行系统中的/sbin/init即可。

解决每个节点的mac地址相同导致的网络通信阻塞问题。在集群构建过程中，我们发现我们使用的开发板所有的mac地址都是完全相同的，这导致了非常严重的网络通信阻塞。为了解决这个问题，我们首先了解了系统获取mac地址的方法。尽管在uboot中配置ethaddr环境变量可以改变uboot环境下的mac地址，但是系统启动后mac地址又会恢复到原来的值。考虑到这可能是内核在

启动过程中读取一个eeprom中存储的值，而在启动后修改mac地址又要先断开网络连接，从而不能从nfs直接启动。我们首先考虑构建ramdisk作为启动文件系统，配置相关环境，然后再将根文件系统转换到nfs上。然而由于这个开发板的特殊性，我们无法使用switch_root或者pivot_root进行相关配置。最终我们将设计修改为将系统烧写到开发板的mmc上，从而可以实现离线配置。

交换每个节点的ssh公钥等节点信息。为了实现节点的互相连接，每个节点都需要存储其他节点的信息，为了防止这种去中心化的配置导致的非常大量的数据交换，我们将集群的节点表和相关配置文件存储到了服务器上，集群节点会在启动后从服务器获取这些配置文件，从而简化增加新节点的配置流程。

3 集群监控系统的构建

为了能够实时观测集群的运行状态，我们还构建了一套集群监控系统。监控系统的任务是在集群节点从/proc/中获取系统的相关运行信息，通过socket协议发送到处理服务器，在服务器端开放一个tcp端口，接收集群发送的信息，同时完成信息的处理和显示。在实验中我们发现这对单个节点的计算性能影响几乎是微乎其微的。

在节点上我们使用python实现了这一功能，在服务器端我们用C语言编写了服务端程序，使用JavaScript和html构建了整个网页以实现实时显示。除此之外，我们正在尝试构建能够绘制节点运行状态图表的程序。

4 进行集群的性能测试

使用单个节点的测试算法，将原来分配给每个线程的任务改为分配给每个节点，观察在集群中增加节点对算法的加速效果，完成集群的性能测试。

四 目前的项目成果

1 实现了节点的相关配置和自动化扩展，在服务端配置完成后一个新的节点可以在启动后立刻加入集群而不必进行更多的配置。节点在配置好启动项后会自动从服务器下载配置文件并进行配置

```
Server://cluster/mpiconfig -> /etc/init.d/mpiconfig
```

```
Server://cluster/config* -> /root/* (ln -> /etc/*)
```

2 实现了集群的构建，完成了集群的性能测试。使用一个需要进行非常多次迭代计算的算法对集群进行了性能测试，得到了集群对计算速度的提升效果数据。

3 实现了节点的性能监控系统的构建。综合考虑了性能监控对节点性能和网络负载的影响，使用本地Socket发送数据到网络端实现了性能监控。服务器使用了在公网的阿里云服务器以保证稳定性。

五 下一步的研究目标

1 优化节点的启动过程，重新尝试使用ramdisk启动，然后挂载nfs文件系统，从而实现集群节点的无硬盘化，进一步降低集群的构建和部署成本。

2 进行集群的能耗测试。比较集群和同等算力的其他计算设备的功耗，测试嵌入式集群对功耗的优化效果。

- 3 实现算法的统一部署。将算法和配置文件部署到服务器上，使节点从服务器获取有关文件，从而简化集群的部署流程。
- 4 尝试优化集群的部署方式，给出集群算力提升的瓶颈，尝试给出解决方法。
- 5 尝试实现集群任务的自动分配，亦即在一个节点开始工作后自动将任务动态地分配给其他节点。

六 项目地址和相关图片等附件

本项目的相关配置文件和测试用程序代码已经开源在 **GitHub** 上，其他程序文件会在确定稳定后加入 **GitHub** 仓库。

相关测试时的照片和文件目录截图见附件。

相关服务器地址：

项目地址：<https://github.com/Shaloc/Cluster-computing>

讨论/贡献：<https://github.com/Shaloc/Cluster-computing/issues>

集群配置服务器：<https://www.shaloc.site/cluster>

集群监控服务器：<https://www.shaloc.site/status>

项目成员 **GitHub** 主页：

<https://github.com/Shaloc>

<https://github.com/happy0day>

<https://github.com/110101001>

<https://github.com/metooChenXiaoJian>