Little Fe Configuration and GalaxSee Detection Code for Simulation the Planets with Pixels and Color Skin

Gerardo L. Figueroa Rosado
Master of Engineering in Electrical Engineering
Luis Vicente López, Ph.D.
Electrical Engineering Department
Polytechnic University of Puerto Rico

Abstract *— In this project we implemented a step by step procedure, of how to configure the portable cluster LittleFe, how to install the UNIX-LINUX operating system to the LittleFe (Server) and modifications to the GalaxSee Code. Also we will see a brief description of what is a LittleFe cluster and the associated hardware. The linux diestro used in this project is very powerful to complete tasks like this one. Galaxies are collections of billions of stars; our home galaxy, the Milky Way, is a typical example. New generations of stars are born out of gas that condenses within regions called giant molecular clouds, and the stars sometimes form into star clusters. When a star reaches the end of its evolution, it may return much of its gas back to the interstellar medium, which will be the source for a new generation of stars. Galaxies can be thought of as systems that turn gas into stars and back again. This is the reason why this project is so important to understand even more knowledge about galaxies.*

Key Terms *—BCCD, CMOS, CUDA, MPI.*

INTRODUCTION

LittleFe is a complete multi node Beowulf style portable computational cluster designed as an "educational appliance" for substantially reducing the friction associated with teaching high performance computing and computational science in a variety of settings. One of these units were donated to the Electrical Engineering. This project explains the setup of the LittleFe, install of the operating system for the LittleFe, and run a test program in the LittleFe cluster.

The LittleFe cluster is an affordable method for teaching computational science with innumerable applications that is accessible to everyone. It is an introductory level hardware into the world of parallel computing that is designed with the purpose of learning.

HISTORY

LittleFe began as an idea by Paul Gray (University of Northern Iowa), Dave Joiner (Kean University), Tom Murphy (Contra Costa College), and Charlie Peck (Earlham College) in 2005. [2]

LittleFe started as a cluster of eight small PC computers in separate cases. Unfortunately, the separate cases, power supplies, and hard drives made the system quite heavy, and was expensive to bring on an airplane. The first step, therefore, was to reduce weight. Then, they had to wrestle with what software we would provide on the cluster. They then started experimenting with running Paul Gray's Bootable Cluster CD (BCCD) on the cluster. This was much more maintainable than Debian, the first OS they are implementing, but its live CD nature meant it was difficult to have configuration and data persist between reboots of the cluster. [2]

Ideally we wanted a single head node with a hard drive, and have the rest of the nodes use the network to run off that drive. The process we settled on involved booting the node with the hard drive into live CD mode, and then running a shell script to copy the contents of the live CD to the hard drive, and then configure that system to boot the rest of the cluster nodes over the network. [2]

In November 2010 the LittleFe project was the recipient of a grant from Intel to build 25 clusters to be given to faculty across the United States who are involved in computational science education. These faculties will use their LittleFe clusters to improve or develop curricula for their students, and ultimately for such globally available resources as HPC University (HPCU) and the Computational

Science Education Reference Desk (CSERD). [1][2]

BRIEF EXPLANATION

As mentioned in the introduction this cluster was designed as an educational appliance for reducing the friction associated with teaching high performance computing (HPC) and computational science in a variety of settings.[1][2][3]

The principle design constraints for LittleFe are:

- $3,000USD total cost
- Less than 50lb (including the Pelican travel case)
- Less than 5 minutes to setup

Minimal power consumption; less than 100 Watts peak, 80 Watts average

The current production LittleFe design is composed of the following major components:

- (6) Mainboards (Mini-ITX, 1–2GHz CPU, 512MB–1GB RAM, 100Mb/1 GB Ethernet)
- (6) 12VDC-ATX power supplies
- 320 Watt 110VAC-12VDC switching power supply
- 40GB 7200RPM ATA disk drive (2.5" form factor)
- (1) 8 port 100Mb/1 GB Ethernet switch
- Fasteners, cabling, and mounting hardware

This portable cluster cost approximately $2500 and weighs less than 50 pounds and the setup can be done in less than 10 minutes. So basically, the LittleFe is a cluster of 6 nodes interconnected via Ethernet connection and with only one head node which is the only one with a hard drive. The LittleFe supports shared memory parallelism (OpenMP), distributed memory parallelism (MPI), and GPGPU parallelism (CUDA). LittleFe makes it possible to have a powerful ready-to-run computational science and HPC educational platform for less than three thousand dollars. Below is shown a figure 1 of the LittleFe assembled. [2]

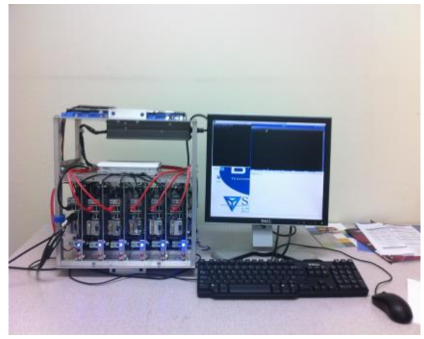

Figure 1
LittleFe Assembled

BEOWULF CLUSTERS

A Beowulf cluster is a computer cluster of what are identical computers networked into a small LAN with libraries and programs installed that allow processing to be shared among them. The result is a high-performance parallel computing cluster from inexpensive personal computer hardware. This same principle can be applied to virtual machines. You could set up a virtual Beowulf cluster using virtual machines. [2]

SETTING UP BCCD ON THE LITTLEFE

BCCD is a Debian based Linux distro specifically made for clusters. Bootable Cluster CD is intended for use as a pedagogical tool for high school and college computational science educators. Using the BCCD, one can quickly setup a computer cluster with all the tools needed to start teaching computational science or parallel programming. Optionally, a permanent cluster maybe automatically configured by copying the live CD image onto a hard drive. [2]

PROCEDURE TO BOOTING AND LIBERATE THE LITTLE FE

These instructions are for the LittleFe project with the BCCD software Version 3.3.1 (32 & 64) bit. This version can be download the image file (.iso) for your architecture (64-bit or 32-bit).

Extracted the (.iso) file to a USB for the installation in the LittleFe cluster. [4]

- First it is needed to setup the computer to accept a USB driver in the CMOS and also setup the other nodes to accept BootTP for later booting up from the main computer that is the main node.
- After setting up the CMOS in the main computer, put the USB Flash Drive with the BCCD image in the USB port. Restart the computer and continuously press the F11 key until a blue screen appears asking to select the boot device, choose to boot from USB Drive.
- After it begins the boot-up process from the USB drive, select in the booting screen the option Default.
- After several booting up processes and installations, it will ask for the password of the default user. The password needed for the setup is: bccd.
- After that a screen is brought up about the network showing "NO DHCP for eth0". Skip this option by selecting "YES".
- If all is functioning properly, a screen that says "WELCOME TO BCCD" will be displayed. Press "okay" to continue.
- To liberate the BCCD script, in other words to erase the installed BCCD program in the hard drive, write the following command in the black prompt:
  sudo perl /root/liberate.pl --libdev /dev/sda
- Restart the computer once the script finishes by running the following command:
  sudo shutdown -r now
  su
    Newpass: letmein
    reboot
- Then, remove the USB flash drive before the computer boots, to allow it to boot from the hard drive.
- After the LittleFe has restarted, execute the following command to setup the computer to send the IP address to the other nodes: bccd-nic-setup.
- Then press enter and write the following command: sudo /bin/bccd-reset-network.
- It will prompt for the password, it will be the one used at setup: bccd.
- Select in the panel that appears (No) DHCP after TURN PxE (YES)
- ETH1 (YES),
- Select to skip DHCP.
- Reboot the computer and turn on every node, one by one. First, turn on the first node and if a monitor is hooked up to it, the BootTP request will be visible from BCCD in a running script. When the node is given a name and number, do the same to the subsequent five nodes until they've all received a name and number

RUNNING THE TRIAL PROGRAM HELLO MPI

- After the system reboot, use the blue screen.
- Enter the command: (cd bccd)
- Enter the command: (cd home)
- Enter the command: (cd bccd)
- Enter the command: bccd-snarfhosts
- Enter the command: cat machines-openmpi
- Now we can test the setup by running a program.
- Enter the commands: cd Hello-world
- Enter the command: make clean
- Enter the command: make
- Now enter the command:
  mpirun -np 2 -machinefile ~/machines-openmpi ./hello-mpi.c-mpi

The number 2 in the previous command is the total number of nodes in our setup. You may have to adjust according to your setup. [3][4]

- If you would like to take a look at the code, then you can use the following command: cat hello-mpi.c

PROCESS TO MOUNT AND UNMOUNTS THE USB

Use these steps to Littler Fe read the USB. (This makes you the root user - aka the superuser, which allows you to do things like manage flash drives).

- Plug in the flash drive.
- Type: dmesg and hit enter. This will show a lot of text including some about a device like /dev/sdb. If it refers to sda instead of sdb, use sda in place of sdb for the remainder of the instructions.
- Type: mount -t vfat -o uid=bccd /dev/sdb1 /usr/local and hit enter. This says the directory /usr/local is the top-level directory on the flash drive. By copying files to /usr/local you are copying them or from the flash drive.
- When you are done using the flash drive, repeat step 1 above to become the root user (aka superuser).
- Then type: umount -l /usr/local and hit enter. Wait a couple seconds and then you can remove the flash drive.
- At this point, you do not need to be the root user anymore.
- You can return to being a normal user by typing: exit

PROCESS TO EXECUTE THE CODE OF GALAXSEE

Use this commands to execute the program of GalaxSee.

- cd GalaxSee
- ls –l (to see the directory with extensions)
- make clean
- make
- mpirun -np 2 -machinefile ~machinefile ~machines-openmpi GalaxSee.cxx-mpi 1000 200 1000 1
- Note: these commands assume a 2 node cluster running 1000 objects of 200 solar masses each for 1000Myears, displaying the results in an x Window.

GALAXSEE ORIGINAL CODE AND SIMULATION

In the line 176 to 182 this is the part to change the color of simulation of the galaxy and the line 209 to 250 is the part to create the loop over stars and put a pixel in for each star all this code are in the archive Gal.cpp of GalaxSee you can see the original simulation in the figure 2.

Code:

Line 176 to 182

```
theColormap = XCreateColormap(dpy,
DefaultRootWindow(dpy),
DefaultVisual(dpy,DefaultScreen(dpy)),
AllocNone);
    for (int i=0;i<numXGrayscale;i++) {
intcolor=(int)((double)i*35535.0/(double)numXGra
yscale)+30000;
            Xgrayscale[i].red=color;
            Xgrayscale[i].green=color;
            Xgrayscale[i].blue=color;
 XAllocColor(dpy,theColormap,&(Xgrayscale[i]));
    }
```

Line 209 to 250

```
    if (dispX < imwidth/2) {
XSetForeground(dpy,gc,Xgrayscale[depthZ].pixel);
//XDrawPoint(dpy,buffer,gc,dispX,dispY);
XFillRectangle(dpy,buffer,gc,dispX,dispY,3,3);
        }
    if (dispX > 0) {
XSetForeground(dpy,gc,Xgrayscale[depthY].pixel)
;
//XDrawPoint(dpy,buffer,gc,dispX+imwidth/2,disp
Z);
XFillRectangle(dpy,buffer,gc,dispX+imwidth/2,dis
pZ,3,3);
```

Simulation:

Figure 2
Original Simulation of GalaxSee

GalaxSee Modification and Simulation

Code

First Modification is to change the color using primary colors additive color system. Primary colors are sets of colors that can be combined to create the sensation of a range of colors. The primary colors are red, green, and blue. Additive mixing of red and green light produces shades of yellow, that show in Figure 3, Mixing green and blue produces shades of cyan, and mixing red and blue produces shades of purple and mixing nominally equal proportions of the additive primaries results in shades of grey or white. The color space that is generated is called an RGB color space.

The second modification is to change the stars and rectangles to circles with the color cyan as seen in the Figure 4 and the third Modification is to change the color to purple and change the stars and the circles to arcs, you can see this in Figure 5. Finally the last step would be to fill the arcs with any image. This final step will be achieved in another project.

Line 176 to 182

```
theColormap = XCreateColormap(dpy,
DefaultRootWindow(dpy),
DefaultVisual(dpy,DefaultScreen(dpy)),
AllocNone);
  for (int i=0;i<numXGrayscale;i++) {
intcolor=(int)((double)i*35535.0/(double)numXGra
yscale)+30000;
        Xgrayscale[i].red=color;
        Xgrayscale[i].green=color;
        Xgrayscale[i].blue=color;
XAllocColor(dpy,theColormap,&(Xgrayscale[i]));
              }
```

Line 209 to 250

```
  if (dispX < imwidth/2) {
XSetForeground(dpy,gc,Xgrayscale[depthZ].pixel);
//XDrawPoint(dpy,buffer,gc,dispX,dispY);
//XFillRectangle(dpy,buffer,gc,dispX,dispY,3,3);
XFillArc(dpy,buffer,gc,dispX,dispY,10,10,0*64,36
0*64);
XDrawArc(dpy,buffer,gc,dispX,dispY,10,10,0*64,
360*64);
      }
       if (dispX > 0) {
XSetForeground(dpy,gc,Xgrayscale[depthY].pixel)
;
//XDrawPoint(dpy,buffer,gc,dispX+imwidth/2,disp
Z);
//XFillRectangle(dpy,buffer,gc,dispX+imwidth/2,di
spZ,3,3);
XFillArc(dpy,buffer,gc,dispX+imwidth/2,dispZ,10,
10,0*64,360*64);
XDrawArc(dpy,buffer,gc,dispX+imwidth/2,dispZ,1
0,10,0*64,360*64);
```

Simulation:

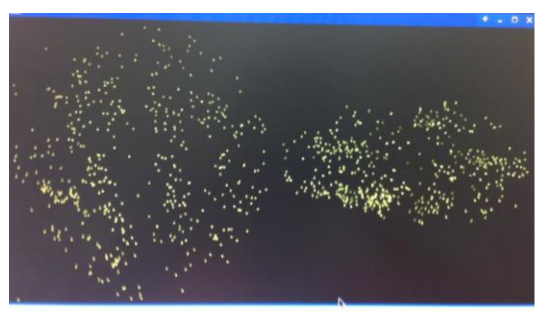

Figure 3
Original Simulation of GalaxSee with Color Yellow

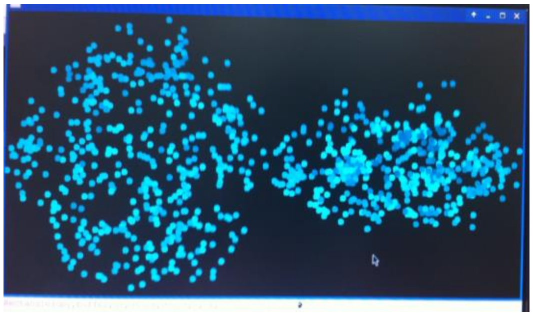

Figure 4
Simulation of GalaxSee with a Circle in Color Cyan

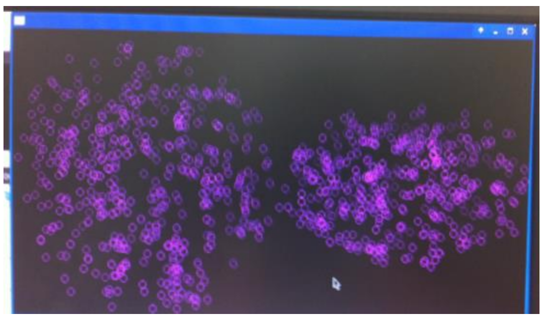

Figure 5
Simulation of GalaxSee with Arc in Color Purple

REFERENCES

[1] SHODOR, "LittleFe Overview by its Designers", 2014. Retrieved on March 18, 2014 from: http://www.shodor.org/media/content//petascale/materials/general/presentations/littlefe-overview_pdf.

[2] LittleFe, "Home Page of Little Fe with a History", July 16, 2013. Retrieved on April 1, 2014 from: https://littlefe.net/.

[3] Torres Andino, E. A., "Student of Parallel Computer Project about the Installation and Setup of the BCCD in the Little Fe", *Polytechnic University of Puerto Rico*, February 20, 2013.

[4] BCCD, "Teaching Parallelism Made Easy", February 18, 2014. Retrieved on March 20, 2014 from BCCD: http://bccd.net/downloads.