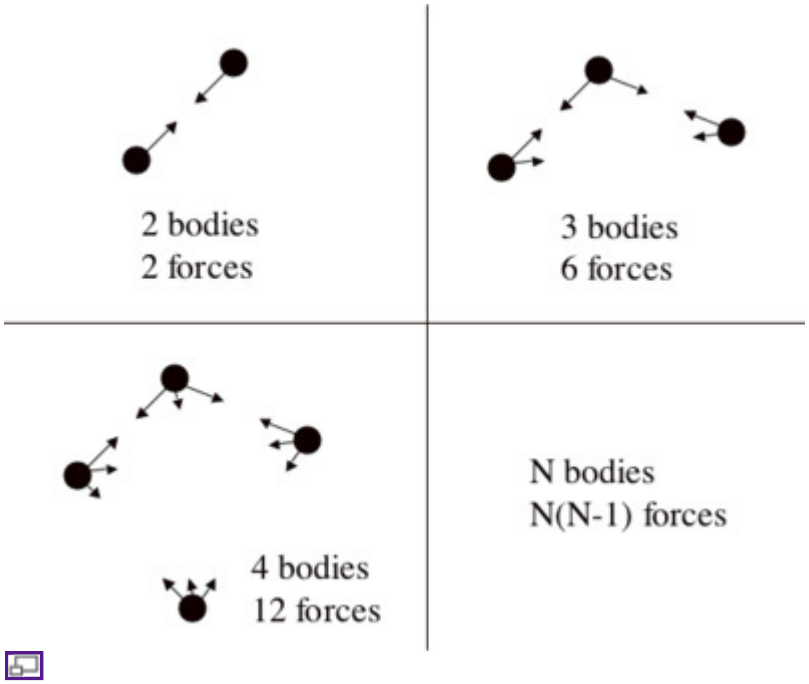
GalaxSee

From BCCD 2.2

Jump to: navigation, search

This tutorial will walk you through an investigation of n-body parallel computing problems using the program GalaxSee. For instructions on how to compile and run MPI programs, please see Compiling and Running. This tutorial assumes that you have booted up the BCCD and have X running (to do this, enter `startx`). It uses mpirun, which means you need an implementation of MPI installed; the BCCD comes with both LAM MPI and MPICH but with MPICH running by default. For more information, see Running MPICH and Running LAM MPI).

GalaxSee: N-Body Physics



One of the grand challenge problems in astronomy is the evolution and structure of the universe and galaxies. The universe is seen to have a structure of sheets and voids on a large scale. Galaxies are seen to often have a spiral structure that is difficult to explain. Space is not occupied by a homogeneous fluid, but by discrete particles that interact through gravity over long ranges.

This is often modeled as discrete bodies interacting through gravity. The gravitational force is given by $F_g = G M_1 M_2 / D_2$, where M_1 and M_2 are two objects masses and D is the distance between them. The acceleration of an object is given by the sum of the forces acting on that object divided by its mass $a = \Sigma F/M$. If you know the acceleration of each mass, you can calculate the change in velocity $a = \Delta v/\Delta t$. If you know the velocity, you can calculate the change in position $v = \Delta x/\Delta t$.

The algorithm can be loosely described as **NEW = OLD + CHANGE**

This is applied to each particle. To apply this to each particle, you need to know the acceleration, but the

acceleration is determined by the sum of all of the forces.

What this means is that the more objects you have, the more forces you need to calculate. What's worse, every object needs to know about every other object.

The GalaxSee code is a simple implementation of parallelism. Since most of the time in a given N-Body model is spent calculating the forces, we only parallelize that part of the code. "Client" programs that just calculate accelerations are fed every particle's information, and a list of which particles that client should compute. A "server" runs the main program, and sends out requests and collects results during the force calculation.

You could think of the total running time in the following way:

- A: It takes some time to send out information on N particles to P-1 processors each of S time steps
- B: It takes some time for each processor to calculate N/P * N interactions each of every S time steps

As long as you run the model for enough time steps that not much time is spent "setting up" the program, a reasonable model for how long the GalaxSee program will take to run on a given cluster is

```
time = A*N*(P-1) + B*N*N/P
```

For your cluster, what would be the coefficients of this equation?

To run the program, first move into the GalaxSee directory by executing `cd ~/Gal`. Next the executable needs to be "made" by running `make`. This will create the executable `Gal`. (If you're using the most recent version of the BCCD, this will show up as a green-colored file, meaning it's executable.)

Next we need to copy this executable to all the nodes that will be running it (If you have not yet set up your nodes for remote access, make sure you have logged into each machine as bccd, started the heartbeat (pkbcast) program, run `bccd-allowall`, and run `bccd-snarfhosts`. You must do this before continuing.) A new automated script now exists to successfully copy executables across BCCD nodes without compromising other user's runs. It is called 'bccd-syncdir', and for GalaxSee, it is run with the following command:

```
bccd-syncdir ~/Gal ~/machines
```

where `~/Gal` is the directory which holds the executable, and `~/machines` is the machinefile created previously with 'bccd-snarfhosts', which contains a list of all the nodes in your cluster. This creates a unique directory in /tmp which holds your executable directory across all nodes. The name of this directory is unique and chosen by the first 8 characters of your current host's public key. Make note of this directory and move into with `cd <your directory>`.

To run the program on one node of your cluster, enter the following command

```
time mpirun -np 1 -machinefile ~/machines ./GalaxSee  500      400      1000.0
            #cpus                                    #bodies  #mass    #final time
```

(Run the following models without a display, and record the wall time for each model.)

- Run the model with 1 processor, 500 bodies, and 1000 Myrs.
- Run the model with 2 processors, 500 bodies, and 1000 Myrs.
- Run the model with all processors, 500 bodies, and 1000 Myrs.
- Run the model with 1 processor, 2000 bodies, and 1000 Myrs.
- Run the model with 2 processors, 2000 bodies, and 1000 Myrs.
- Run the model with all processors, 2000 bodies, and 1000 Myrs.

“Guestimate” the coefficients, and try it out for a few different runs other than the ones above. Does this “model of the model” work?

What happens to your efficiency as you add processors? What would happen if you went from 4 to 8 CPUs? 8 to 16? 16 to 32? 32 to 64?

What if the model were bigger? Real “production” N-Body codes use millions of particles. How long would it take to run a million particle code for the lifetime of the universe (14,000 Myrs with an 8 Myr timestep) on your cluster? What if you had 16 CPUs? 32? 256? 512? 2048?

- Disclaimers