Tests Conducted:

The tests consists of the following jobs to be submitted to Condor:

1. **host_job.sh**

This is a simple shell script which will execute on the target machine. It will sleep for some amount of time (passed as command line argument), and then terminate. It will for the purpose of establishing the identity of the machine it is executing upon will print the name of the host it is executing upon and the process id it has got. It will additionally also describe the command line arguments passed to it, and the exact location of the binary on the executing machine.

2. **pi_calc.c**

This program is written in C and will calculate the value of Pi to the no of places given as a command line argument. This is a typical example of long running job. This job is included in the test suite to test if the information is lost on the disturbance of the executing machine. The no of places supplied must be the power of 2. This job prints the Pi value thus calculated on to the stdout, which is redirected to a file pi_calc.out.

The rest of the programs are of Monte Carlo simulations, these form the typical problems executed on the grid for High Energy Physics experiments.

The Monte Carlo technique is a very flexible method for simulating light propagation in tissue. The simulation is based on the random walks that photons make as they travel through tissue, which are chosen by statistically sampling the probability distributions for step size and angular deflection per scattering event. After propagating many photons, the net distribution of all the photon paths yields an accurate approximation to reality.

3. **tiny_mc.c**

Simulates light propagation from a point source in an infinite medium with isotropic scattering. The entire source for this program fits on a single page and is a good way to get an overview of the entire Monte Carlo process.

4. **small_mc.c**

Simulates light propagation from normal irradiation of a semi-infinite medium with anisotropic scattering. It calculates the volumetric heating as a function of depth. This program takes two whole pages, but is very handy for a bunch of problems.

5. **time_mc.c**

Simulates the time resolved backscattering of a semi-infinite medium with anisotropic scattering. This program is adapted from `small_mc.c` above, and shows how simply time resolved simulations can be done.

6. **cvh_pass.c**

This program is a an implementation of the brute force password cracking method on the grid. It uses the brute force method to guess a password. It also displays the time taken for it to complete. For benchmarking purposes, this ouput of the program can be used. This is also a typical example of long running jobs submitted to the Grid for

solutions. It can also be used constructively to test the strength and quality of the password chosen.

7. **cvh_cpu_intensive.c**

   The last program of the suite is involving many bit wise operations to make it CPU intensive and not Memory or IO intensive.

Details of tests conducted and results

The system was subject to tests wherein the queue completion time was taken in consideration. Here we are not concentrating on the completion time of a single job. We have formed queues of the jobs described above. These queues are submitted from the submitting machine and their completion is monitored.

The time taken for the completion of each queue is reported. The whole bunch of queues of 7 jobs in all is submitted atleast 3 times . This is done to get a fair approximation of the completion time of the queue on the test machine.

The following is the description of the queues used in the evaluation :

host_job: 50 Jobs in a queue

pi_calc.c: 50 Jobs in a queue

small_mc.c: 50 Jobs in a queue

tiny_mc.c: 50 Jobs in a queue

time_mc.c: 50 Jobs in a queue

cvh_pass.c: 50 Jobs in a queue

cvh_cpu_intensive.c: 50 Jobs in a queue