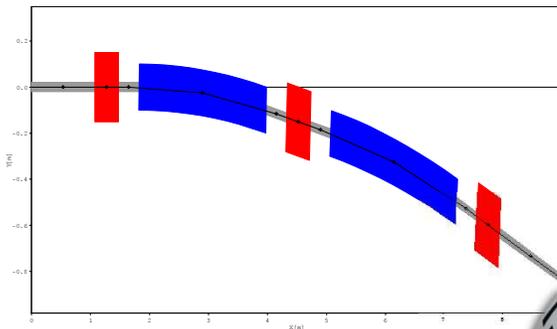




## Abstract & Objectives

Many of the tasks in accelerator physics need a complex modeling and heavy computational resource. One of this tasks is the charged particles beam behavior during the energy ramping in the booster synchrotron. As many as 10000 particles should be tracked for 4 million turns in the booster to get the physical properties of beam. For tracking each particle for one turn around the booster as minimum 2000 "elementary" actions should be done. Taking into account all above mentioned it is required to use powerful computer or cluster.



## Materials & Methods

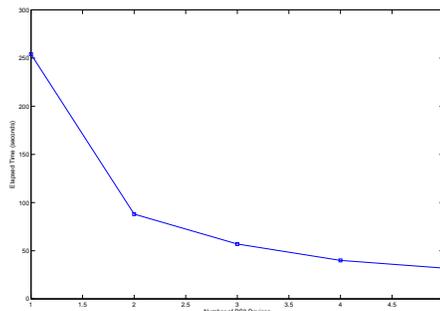
There are many particle accelerator simulation codes around the world. Some of them are for specific purpose others are general purpose. For our task accelerator simulation code named "ELEGANT" was chosen. "ELEGANT" stands for "Electron Generation And Tracking," a somewhat out-of-date description of a fully 6D accelerator program that now does much more than generate particle distributions and track them. "ELEGANT", written entirely in the C programming language[1]. Parallel version of program called Pelegant. Parallelization was done using MPICH which is greatly simplify implementation of code on supercomputers. Some of the abilities of "ELEGANT" listed below

- ✓ Tracking of rings, linacs, and transport lines
- ✓ Aperture determination
- ✓ Tracking and computations with errors
- ✓ Control lattice parameters
- ✓ Bunch generation, or reading of bunch data from files
- ✓ Multi-stage simulation
- ✓ Time-dependent ramping or modulation
- ✓ Determination of frequency maps for rings
- ✓ Change of particle type (default is electrons)



## Results

Computing group had established a small PS3 cluster (five PS3 devices) on SESAME site to analyze and deeply investigate Pelegant code over Yellow-Dog Linux platform, a need for software repository was raised so the computing group took the action by Mr. Mostafa Zoubi who has implemented Yellow-Dog Linux repository which is now effectively and harmoniously working on IMAN1 hardware. After implementing Pelegant in IMAN1 test task was used to investigate the performance of the code. Figure below shows the run time of the test task as the function of the numbers of computing nodes.



## Conclusion

- ❑ Parallel version of the "ELEGANT" was implemented on 512 nodes of IMAN1 supercomputer.
- ❑ Test task was used to test performance of "ELEGANT" on IMAN1. Results shows almost linear reduction of running time with the growth of the number of computing nodes.
- ❑ Utilizing the big number of CPUs available at IMAN1, PELEGANT code can run effectively and efficiently on IMAN1 resources.