CONDOR AS A RESOURCE FOR ACCELERATOR RESEARCH

J. Smith, C. Lingwood, Cockcroft Institute/Lancaster University, UK
J. McKenzie, J. Kewley STFC Daresbury Laboratory, Warrington, UK

Abstract

This work reports on the developments of a computational infrastructure framework that aids achievement of computational research objectives. Examples from a broad range of accelerator problems will be presented, along with ways in which the workflow can be modified.

INTRODUCTION

Condor is a platform for *High Throughput Computing*, that is where the total amount of computational work to be done is large, but none of it requires massive resources at any given time, contrasting with *High Performance Computing*, where exceptional performance is required, but not often for a sustained period of time. The authors note other groups in closely related fields have used Condor to aid research for somewhat different tasks[1], nevertheless the work here illustrates the flexibility of Condor for accelerator projects.

At the heart of Condor is a server process which collects both requests to do work, and offers of processing power. Requests to do work and offers of processing power are sent as 'classified adverts', and the central server matches up the most appropriate resources for each request, taking into consideration target platform, memory requirements, fair use considerations and any local policies in place on the other machines. A typical policy would be to prevent jobs from running while the computer is being used either for a non-condor intensive process, or while the computer is being used interactively. As condor can tell when both of these are the case, it is excellent for scavenging CPU cycles that would otherwise be wasted on desktop machines.

HARDWARE RESOURCES

The hardware behind the Cockcroft condor pool is comprised of a mixture of laptops, desktops and servers, running on a variety of linux and windows variants, owned by Lancaster, Liverpool, Manchester and STFC Daresbury Laboratory. In total around 100 cores are present however there are plans to expand it's scope considerably further.

ABCI SIMULATIONS

ABCI[2] calculates the interaction of an accelerated electron beam moving at near the speed of light with its surroundings, determining the impedance and wakefields experienced behind the bunch. Previous work used this condor pool to quickly evaluate the wakefield kicks for a class of collimator in a situation with a number of free variables[3]. No modification to the ABCI source code was required to achieve this parallelisation, and the swift mapping of the parameter space allowed an optimal solution to be determined.

ASTRA SIMULATIONS

Optimisation Task

Design and optimisation of an injector line for future FEL-based light sources is time consuming due to the large number of free parameters and their interdependence. Genetic/evolutionary algorithms are well suited to help automate this procedure. A multi-objective genetic algorithm, based on a non-dominating sorting technique, similar to [4] has been written in Mathematica. This generates input files for ASTRA [5] that are used to perform beam dynamic simulations of the injector. The genetic algorithm works by creating a population of input files and further 'generations' of solutions are created from the output of a single generation of simulations.

High throughput distributed computing, such as that provided by Condor, is an ideal technique to perform the large number of simulations required for this kind of optimisation procedure, particularly as each simulation in a 'population' is entirely independent of the others. Non-dominating sorting allows the genetic algorithm to optimise for multiple conflicting objectives concurrently, producing sets of solutions which are each equally optimal, showing the trade-off between the objectives. This technique has proved especially useful when comparing different high repetition rate injector options for the UK's New Light Source project [6]. Figure 1 shows a Pareto-like set for one of the NLS injector options, with the objective to minimise both transverse emittance and bunch length. Figure 2 shows one of the resulting solutions for a 200 pC electron bunch accelerated to approximately 120 MeV using a injector consisting of a VHF gun, 3 solenoids, a buncher cavity and ten 9-cell tesla-type superconducting cavities.

AJDISK SIMULATIONS

About AJDisk

AJDisk is a rapid quasi-2D model simulation tool developed from JapanDisk for predicting the properties of round and sheet beam klystron interaction structures. The dense beams are represented as dense disks or rings of electrons. In this case it is used to quickly calculate the main properties of a round beam klystron interaction structure without the need for computationally intensive 3D simulations.[8] Optimising a klystron interaction structure by hand is a long process with no guarantee of an optimum result as

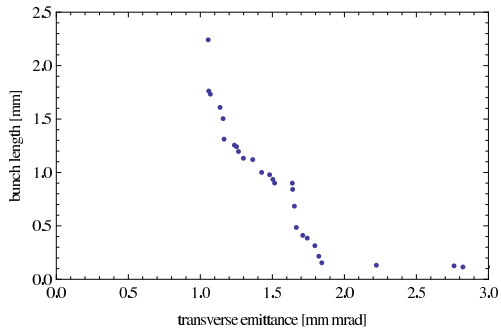

Figure 1: The Pareto-like front showing the trade-off between transverse emittance and bunch length for a VHF gun based photoinjector, operating at 200 pC

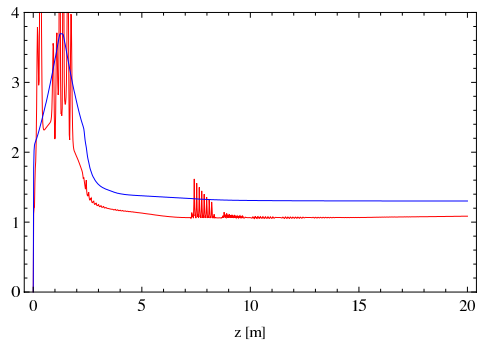

Figure 2: Normalised transverse emittance (red) and rms bunch length (blue) for a VHF gun based photoinjector, operating at 200 pC.

several parameters must be optimised simultaneously. An evolutionary algorithm is well suited to this multivariate and mutli-objective problem.[7]

Evolutionary Algorithm Selection and Application

Evolutionary algorithms *(EAs)* are an example of a parallel meta-heuristic search technique inspired by natural evolution. Unlike many optimisation methods, EAs work from a population of numerical solutions. Recombination operators are applied to share information, and mutation operators to explore new regions of the search space. This allows for complex, non-linear and multi-modal (multiple optima and/or sub-optima) optimisation problems to be efficiently explored.

EAs can be applied to any problem where a decision vector - a vector of parameters that can be chosen - can be mapped to a solution vector - a vector of characteristics to be optimised. Problems with multiple solutions are referred to as Pareto multi-objective optimisation problems. Rather than producing one optimum, a Pareto set of non dominated optima solution vectors are produced, where no one characteristic can be further optimised with detrimental effect on the other characteristics. This is shown in figure 3). [4]

Traditional multi-objective optimisation techniques are not well suited to a problem where solutions clump in a

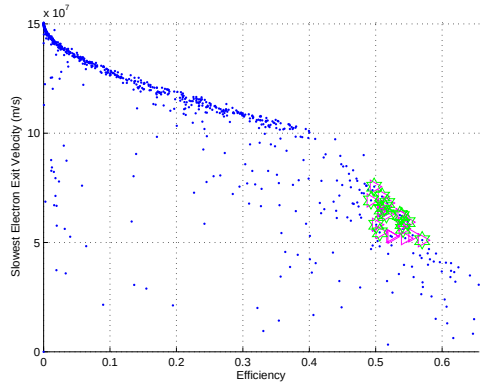

Figure 3: Example of Pareto front plot (purple and green marks represent the first pareto front)

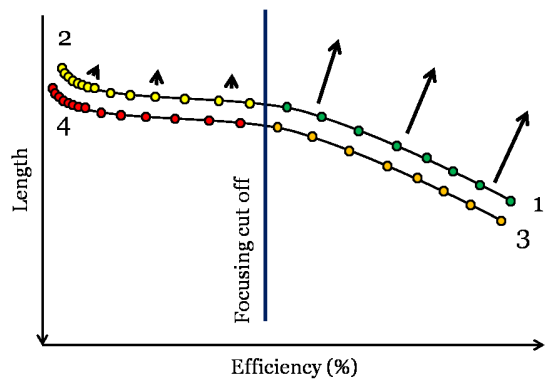

Figure 4: The effect of Pareto search pressure focusing on a klystron optimisation (labels represent Pareto front index, length of arrow represents search pressure)

poor region of solution space, for example short klystrons with poor efficiency are statistically likely, whereas short klystrons with high efficiency are desirable. With Pareto search pressure focusing we define a cut off (as a set fraction of the range) below which all individuals have their fitness (chance of selection for reproduction) reduced by demoting them to the next Pareto front. As fitness is dominated by the Pareto front index this lowers their probability of selection and so increases the search pressure in the desirable area. Figure4 shows the trade off between high efficiency and maximising the velocity of the slowest electrons to prevent reflections and illustrates pareto search pressure focusing.[7, 4]

Mapping Of Computational Task To The Condor Pool

A klystron interaction simulation of sufficiency accuracy can take from 2-20 minutes depending on the structure to be investigated. When it is considered that for a good optimisation at least 10,000 simulations must be performed, clearly some form of high throughput computing is required.

The Condor pool is a natural fit for large scale sets of AJDisk tasks as unlike most commodity clusters the nodes principally run Windows XP. Moreover it's ability to produce XML log files when jobs complete either successfully or otherwise allows it to interface with a Java toolkit[1] which handles the evolutionary algorithm and talks to directly to MATLAB. Consequentially the EA can be informed when a solution is ready, or if there is an error the job can be resubmitted.

AJDisk's variable run time, even before CPU differences are taken into account, renders the standard generation model of EAs inefficient as the slowest simulations determine the generation run time. An asynchronous model is more suited to this problem where a single calculation is now analogous to a generation. The EA maintains a buffer of decision vectors to be simulated, generating a new decision vector using the most up to date information available when required. This model maintains the load on the available cores at 100%, although in reality this is reduced due to unavoidable overheads due to Condor's negotiations and file transfers. In this case Condor not only increases the speed of computation but informs the design choices of the experiment.

SPUR SIMULATIONS

SPUR[9] is a tool to calculate the spectra of spontaneous radiation from undulators. Previously, MPI parallelisation was employed to split the points at which the radiation spectrum is calculated between different processors. Since these tasks do not require knowledge of the calculations being performed by their peers, Condor would provide a natural environment to perform these calculations. Work on dividing the problem like this is planned for the future.

EXPERIENCES WITH PROPRIETARY SOFTWARE

Condor policy can be set to run all tasks with minimal privileges, or to allow these to run with the privileges that the submitting user has on that machine. The latter is necessary if one wishes to use proprietary software, such as Vector Fields Opera, or CST Particle Studio. Both have solvers which accept distributed computing tasks, however these are unaware of other packages. Condor provides a mechanism to submit such jobs, and this has enabled the same powerful hardware used for simulations with CST to be shared with a number of other applications, such as those described elsewhere in this paper.

CONCLUSION

Condor is a simple way to bring together hetrogeneous resources from one or more groups to enable distributed computing with minimal investment. As most applications can be used unmodified, the time taken to set up the pool is offset by the increased throughput of simulation jobs a research group can achieve. A collection of examples have been presented demonstrating the versatility of Condor to help to speed up accelerator development.

[1]Condor Java API available from http://staff.aist.go.jp/hidenakada/condor_java_api/index.html

REFERENCES

[1] A. Baranovski et al., "CDF II Production Farm Project", NIM A 572 (2007) p. 399-401

[2] Y. H. Chin et al "ABCI Progresses and Plans: Parallel Computing and Transverse Shobuda-Napoly Integral", PAC'07, Albuquerque, NM, June 2007 p. 3306, http://www.JACoW.org

[3] J. Smith "GdfidL Simulations of International Linear Collider Candidate Collimator Assemblies", EPAC'08, Genoa, Italy, June 2008, WEPP165, p. 2874-2876 , http://www.JACoW.org

[4] K. Deb et al., "A fast and elitist multiobjective genetic algorithm: NSGA-II", IEEE Transactions on Evolutionary Computation 6 (2002) p. 182-197

[5] K. Fltmann, ASTRA homepage http://www.desy.de/mpyflo/Astra

[6] B. L. Militsyn et al "High Repetition Rate Electron Injectors for FEL-Based Next Generation Light Sources", LINAC08, Victoria, BC, September 2008, TUP042 p. 481-483 http://www.JACoW.org

[7] C. Lingwood et al., "Evolutionary Algorithms and High Efficiency Multiple Beam (MBK) design", IVEC 2009, Rome, Italy, April 2009.

[8] A.J. Jensen et al., "Sheet Beam Klystron Simulations Using AJDISK", IVEC 2006, Monterey, CA, April 2006, p. 489-490

[9] N. Rider, S. Reiche and D.J. Scott "SPUR: A New Code For The Calculation Of Synchrotron Radiation From Very Long Undulator Systems", EPAC'08, Genoa, Italy, June 2008, WEPC128, p. 2305 , http://www.JACoW.org.