CERN MODULAR PHYSICS SCREENSAVER
OR
USING SPARE CPU CYCLES OF CERN'S DESKTOP PCS.

Eric McIntosh, Andreas Wagner, CERN, Geneva, Switzerland

Abstract

CERN has about 5500 Desktop PCs. These computers offer a large pool of resources that can be used for physics calculations outside office hours.

The paper describes a project to make use of the spare CPU cycles of these PCs for LHC tracking studies. The client server application CPSS (compact physics screensaver) is implemented as a lightweight, modular screensaver and a Web Application containing the physics job repository. The information exchange between client and server is done using the HTTP protocol. The design and implementation is presented together with results of performance and scalability studies. A typical LHC tracking study involves some 1500 jobs, each over 100,000 turns requiring about 1 hour of CPU on a modern PC. A reliable and easy-to-use Linux interface to the CPSS Web application described in this paper has been provided. It has been used for a production run of 15,000 jobs, using some 50 desktop Windows PCs, which uncovered a numerical incompatibility between Windows 2000 and XP. It is expected to make available up to two orders of magnitude more computing power for these studies at zero cost.

INTRODUCTION & OBJECTIVES

Available Unused Computing Resources

The desktop computer park at CERN consists of about 5500 desktop PCs running the Windows operating system. The estimated usage of their CPU resources during the normal lifecycle is shown in Table 1. The CPU speed breakdown of CERN's desktop computers is shown in Figure 1. It can be easily seen that there is a significant potential of unused computing resources available.

Table 1: CPU Resource Usage

Lifecycle of standard CERN Desktop PC		
Lifetime	About 3 years	25400 hours
Office Usage	40 hour/ week * 50 weeks/year * 3 years	6000 hours
Idle Time	non office hours	~20000 hours *)
*) NB 1: Assuming PC powered on 24/7 NB 2: Idle time during normal office usage not counted!		

The idea of using idle CPU cycles was previously very successfully applied in the context of astronomical research with Seti@Home [1] and BOINC [2]. CERN is also evaluating LHC@home [3] based on BOINC.

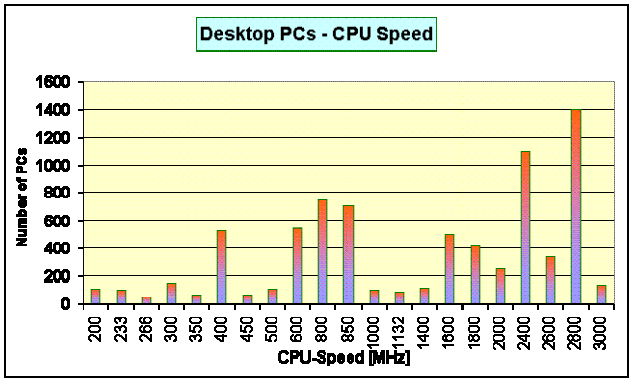

Figure 1: CPU speed distribution of CERN's desktop PCs.

Workload for LHC design studies

In the LHC design and commissioning phase many tracking studies are needed to verify the long-term stability of the beam. This application was typically run on large CERN Linux batch clusters and it has now been ported to Windows. The scale and extent of such studies has always been limited by the available computer capacity. The modest input/output requirements, as shown in Table 2, demonstrate the suitability of the application for a distributed computing system.

Table 2: SixTrack – computing resource usage

SixTrack—Required PC Resources	
Total Memory Requirement	65 MB
Working Set	32 MB
Input Files	250-500 KB
Output Files	20 MB +15 MB per particle pair (about 3MB compressed)
A complete study produces ~500 MB for 100 000 turn jobs	
CPU time - PIII 1600 MHz	1 hour

ACCELERATOR DESIGN STUDIES

Accelerator design studies, typically using the MAD8 [4] or SixTrack [5] Fortran 77 programs have been an important part of the CERN computing workload for many years. These computations are floating-point intensive but have rather modest input/output requirements compared to the event simulation/processing applications of the experiments.

The studies have been performed on the classical computer services; first mainframes (vectorised / pipelined), then workstations, and now on LINUX PC clusters. They are excellent candidates for the workload of a distributed system.

The SixTrack program, optimized for LHC studies, tracks many pairs of nearby particles, with different initial angles and amplitudes, for up to a million turns round multiple models of the accelerator.

Provided the tracking is performed in a symplectic manner, the determination of the onset of chaotic motion defines the Dynamic Aperture. A Linux run_environment, to facilitate the generation, submission, execution, and analysis of the thousands of jobs used, is also available.

The principal objectives were to verify the SixTrack results on Windows PCs, extend the run_environment to facilitate processing with CPSS, on the GRID, and on BOINC.

SIXTRACK ON WINDOWS

The SixTrack program is portable; it is part of the SPEC FP 2000 Benchmark Suite [6], but while individual runs may be comparable across Compiler, Operating System and Hardware platforms, a complete study must be performed on identical hardware and software. Small initial numerical differences grow exponentially over time leading to different values of the dynamic aperture which makes comparative magnet evaluation studies much more difficult. First tests on Windows using the CERN installed Compaq Visual FORTRAN Version 6, now obsolete, showed a thousand fold reduction in performance when dealing with infinite and undefined values. This issue was resolved by more frequent checking for chaotic particles. Further testing then uncovered a difference between results from Windows 2000 and Windows XP systems. This was traced to the input values, of approximately 1000 out of 60 million magnet errors, being one least significant bit bigger on the W2000 system. This was never satisfactorily explained but "the problem does not exist with up-to-date compilers". The Lahey [7] lf95 compiler was adopted instead. Not only did this compiler solve the input problem, but, provided compatible versions for Linux and Windows were used, gave identical results across these platforms.

Finally, a checkpoint/restart capability was introduced to SixTrack, a classical time space trade-off producing a state file of 10KB as against a full dump of over 30MB. Occasional, even frequent interruptions of the desktop PC screensaver can now be tolerated and allow extended runs of one million turns or more.

CPSS ARCHITECTURE

To make use of the unused CPU cycles of desktop PCs and to reuse the existing Web infrastructure of CERN a simple lightweight modular screensaver was developed. The client server application communicates via HTTP with a Web Application containing the job repository. The same protocol is used for job submission and download of results (Fig 2).

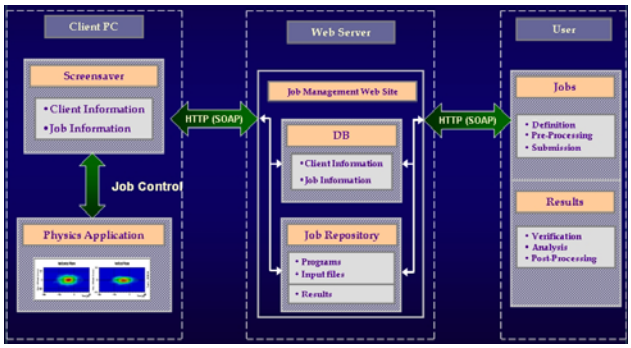

Figure 2: Compact Physics Screensaver Architecture

In a typical work cycle of the CPSS client screensaver it first contacts the server to check for updates and then requests task units that it downloads and executes. Once the execution of the task is finished the result files will be uploaded back to the server. The description of each task contains client requirements such as OS version, CPU speed minimum, available disk space and priority. Furthermore it contains a task description in terms of input/output files, the pre-, main- and post-commands, checkpoint/restart capability and result files.

Figure 3: Status display of the CPSS screensaver

One of the design considerations was to implement the CPSS client in a non-intrusive way. If the user returns to his workplace while a task is being processed, the screensaver will shut down instantly and hand back control to the user. The next time the CPSS client starts up

it will reconfirm the task assignment and if not instructed by the server to drop the task, it will resume its execution (if checkpoint/restart is enabled) or simply restart the task from the beginning.

THE SIXTRACK RUN_ENVIRONMENT

The original SixTrack run_environment provided shell scripts to prepare the mask input files from the LHC database, prepare the final input files and job scripts, submit the jobs, and process the results. It was necessary to modify each and every script for different runs or case studies.

The variables of interest were gathered into an environmental definition file, job submission was extended to CPSS, and new scripts were provided to download results, check the run status, and re-submit lost or erroneous tasks. The jobs running on CPSS (or on other distributed systems such as the GRID or BOINC) cannot in general directly return results to the remote file systems. Instead the results are held in a buffer, verified, and then distributed. Book keeping is done with a small text file database and a set of log files.

CPSS DEPLOYMENT & PERFORMANCE

CPSS is presently used for CERN's Windows based desktop PCs on a voluntary basis. After several months of reliable operation on some 50 PCs, an invitation was sent to users of the CERN IT department encouraging them to install the CPSS client, followed by an article published in the CERN internal Weekly Bulletin in the middle of August. The increase in the number of clients as a result of these announcements can be clearly seen in the number of participating machines (see Fig. 4). There are presently about 500 desktop PCs registered.

The throughput of CPSS increased proportionally. The number of results returned and the CPU time per day, as well as the total number of processed tasks and total CPU contribution, is shown in Figure 5. About 200,000 tasks for various LHC design studies have been processed using some 10,000 days of CPU time.

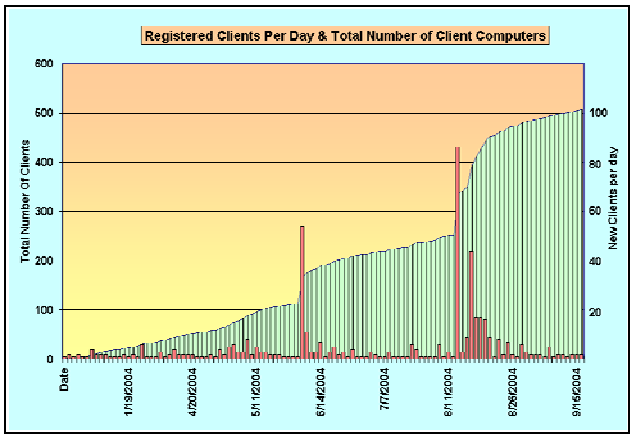

Figure 4: Evolution of the number of registered clients

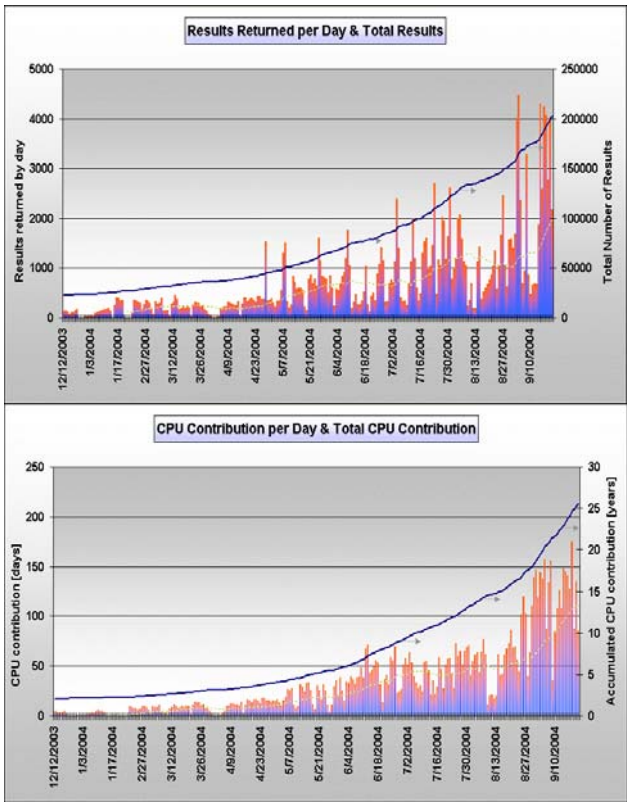

Figure 5: Number of results processed and CPU contribution per day and accumulated values.

RESULTS

First studies involving 1,500 and 15,000 1 hour jobs, 100,000 turns for 30 particle pairs, 5 angles, and 6 initial amplitudes were completed successfully. The 1,500 results were verified as being identical with those from Linux. In the 15,000 job case many tasks were duplicated and the results again verified as being identical. This was sufficiently encouraging to attempt a much more detailed investigation of the possible Dynamic Aperture limitations using 1000 angles implying 200 times more computing. This study is almost 50% complete using the existing 500 clients, and should finish before the end of the year. The results for a particular case of the study are shown in Figures 6 and 7.

The robustness of the procedures and the recovery from the inevitable errors, node crashes, data corruption, and user mistakes, is satisfactory.

Testing of the effect of beam-beam interactions has uncovered further numerical differences, again at the level of the least significant bit. Some results of the EXP and LOG functions have been found to be different between different brands of PC in spite of using the same statically-linked executable, i.e. the same libraries and generated code. The same or similar effect has been produced using other compilers and libraries. Unless this problem is resolved it will not be possible to easily use SixTrack on a heterogeneous computing system.

The concept of using spare desktop CPU cycles for extensive accelerator design studies has been validated. The scalability of the run_environment and associated data structures needs to be improved to cope with the very large number of runs and results which are now possible

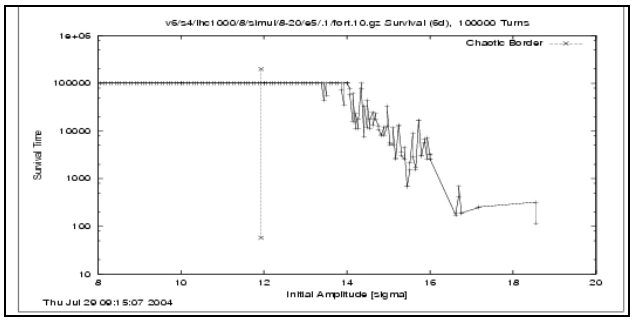

Figure 6: Survival time as a function of the initial amplitude.

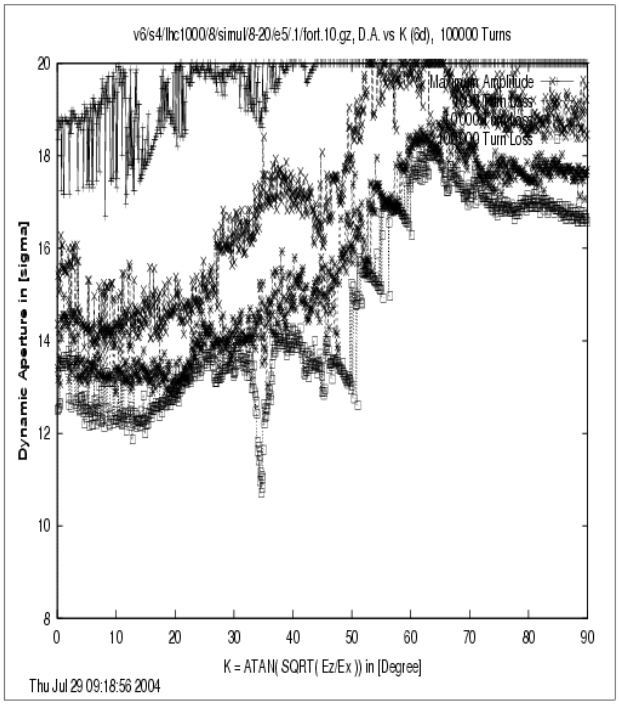

Figure 7: Dynamic Aperture as a function of the initial angle

REFERENCES

[1] SETI@home: http://setiathome.ssl.berkeley.edu/
[2] BOINC: http://boinc.berkeley.edu/
[3] LHC@home: http://cern.ch/athome
[4] MAD8: http://cern.ch/mad/
[5] SixTrack: http://cern.ch/frs
[6] SPEC FP2000 Benchmark Suite: http://www.spec.org/
[7] Lahey Fortran: http://www.lahey.com