Abstract

The Alpha Magnetic Spectrometer (AMS) is the high energy physics experiment installed and operating on board of the International Space Station (ISS) from May 2011 and expected to last through Year 2024 and beyond. The details of porting of the AMS software to the IBM Blue Gene/Q Architecture are discussed. The performance of the AMS reconstruction and simulation software in that architecture is evaluated and compared to the performance obtained on Intel based architecture.

Introduction

The Alpha Magnetic Spectrometer (AMS) is the high energy physics experiment operating on board of the International Space Station (ISS). The detector has large geometrical acceptance of 0.5 m$^2$ sr, and is equipped with 164 cones. The software was written using the C++ language and equipped with tens of thousands of lines of code. A complete set of software for the simulation and reconstruction of the AMS data is available on the open source high performance computing URL https://www.lpi psychosis.eu. The software is designed to run on the BlueGene/Q supercomputer, the architecture of which is described in detail in Ref. [2]. The BlueGene/Q architecture is described in details in Ref. [2]. It includes the login nodes, equipped with POWER 7-355 GHz processors, 128 GB of memory and running Linux 2.6 operation system, and compute nodes equipped with 164 cores. The POWERPC 4.6 GHz processors running lightweight proprietary kernel (CUN), with 16 GB of memory. The PowerPC A2 processor features four-way hyper-threading, so up to 64 threads per node can be run. All performance studies were done on compute nodes, while ported software equally works on login nodes.

IBM Blue Gene/Q: architecture and compilers

The IBM Blue Gene/Q architecture is described in details in Ref. [2]. It includes the login nodes, equipped with POWER7-355 GHz processors, 128 GB of memory and running Linux 2.6 operation system, and compute nodes equipped with 164 cores. The POWERPC A2 processor features four-way hyper-threading, so up to 64 threads per node can be run. All performance studies were done on compute nodes, while ported software equally works on login nodes. There are few distinct features of this architecture found to be essential for the software porting:

- 64 bit address space;
- Big Endian addressing scheme;
- Limited support for Linux system calls and in particular no support for fork() and system() calls on compute nodes;
- Massively parallelization beyond the SMP one. OpenMP[3] is used usually to synchronize threads running in different nodes.

The actual porting of the software were done on JUQUEEN computer of Juelich Supercomputing Center[4], where the minimal job configuration in the batch system includes 32 nodes or 2048 threads, while typical one consists of 128 or 512 nodes.

The IBM compilers xC 12.1 and sff 14.1 were used to compile and link all the software. These compilers support OpenMP[5] directives, with the major exception of not supporting the omp threading pragmas for any STL container (vector, map, etc.). xC 12.1 supports a subset of C++11 directives, and in particular thread local storage (TLS) via __thread directive with the same exception for STL containers.

Software porting – ROOT

The ROOT 5.34.62 was not available to this platform (codenamed here as Linux/pleixyppc6) due to incompatibility between the ROOT CINT interpreter and 64 bit addressing space with BigEndian features of PowerPC processors[7]. This was fixed by changing a line in the cint/init.c/val/vle file like:

```cpp
< if (bufstype != 'i') return (TBuffer *) buffer; lin;
```

Another minor issue was xC compiler internal error during compilation of RooFit dictionary. This was fixed by dividing the dictionary file by several parts.

After the successful build of the root executable and all shared and static libraries became possible, see Fig. 1.

Software porting – GEANT4

The IBM Blue Gene/Q Architecture was not supported by GEANT4 10.1 package[8]. To do that, the following architecture file Linux ppc-nt.gmk was added:

```cpp
GLOBALLIBRARY(Include/linux.h)
GLOBALLIBRARY(Include/synthcplex.i)
particle/management/src/GetDetectorsSpace.cc
generate/parcels/src/GEANT4relationship.cc
```

After the changes, the GEANT4 10.1 libraries were built and test examples successfully ran in multi-threaded mode.

Statistics of simulation software – GEANT4

The port of CERNLIB[9] software was needed, as AMS software depends on it, to ensure the FORTRAN local variables being initialized in stack to allow thread safe processing. Also the MINUIT package needed to be adapted to thread safe mode using OPENMP technique.

Software porting – AMS software

• Simulation of Linux system() calls

Due to absence of system() support on CNK kernel, the following system calls were rewritten using the C++ libraries (including _read, _write, _close, _open, _exit, __thread, _getenv, __sleep, etc.);

• Memory management

Due to lack of support Linux system routines like getrlimit() ... the proprietary routines were used to estimate the amount of free memory available for jobs execution.

Conclusions

The AMS and other (ROOT, GEANT, CERNLIB) software was successfully ported to IBM Blue Gene/Q architecture. Massively parallel jobs, up to 2048 nodes and 131K threads successfully ran on JUQUEEN computer for wall of 24 hours, which it the maximum amount of time allowed by batch job scheduler. The AMS massive simulation data production is expected to start in the year 2017 to deliver up to 15% of AMS simulated data.

Acknowledgements

The authors thank the Juelich Supercomputer Center for hospitality and for providing the access to JUQUEEN computer.

References

[1] Ting S C 2007 AMS-02 TIM Meeting at CERN