# apeNEXT: Experience from Initial Operation*

N. Christian, L. Morin, D. Pleiter†, H. Simma (DESY, 15738 Zeuthen, Germany)
F. Belletti, F. Schifano, R. Tripiccione (INFN, I-43100 Ferrara, Italy)
S. Delucca, F. Rapuano, D. Rossetti, P. Vicini (INFN, I-00185 Roma, Italy)
F. Bodin, J. Fouriaux (IRISA/INRIA, Rennes, France)
Ph. Boucaud, O. Pene (LPT, Orsay, France)

*Abstract*

apeNEXT[1, 2, 3] is the latest generation of massively parallel machines optimised for simulating QCD formulated on a lattice (LQCD). In autumn 2005 the commissioning of several large-scale installations of apeNEXT started, which will provide in total a compute power of 15 TFlops. This fully custom designed computer has been developed by an European collaboration composed of groups from INFN (Italy), DESY (Germany) and CNRS (France). In this contribution we give an overview on the system architecture and software, present performance numbers and finally report on experience gained during the first months of machine operation.

## INTRODUCTION

Progress in the field of Lattice QCD (LQCD) heavily depends on the availability of computing resources. State-of-the-art projects require several TFlops of sustained performance per year. Applications from LQCD spend most of their time in a few kernel routines. Typically, a routine performing a matrix times vector operation is most relevant. This so-called fermion matrix is huge but sparse, and a remarkably small set of performance signatures are relevant for implementing multiplication efficiently on any given computer architecture:

- Arithmetic operations are dominated by floating-point operations with complex operands. At least a significant part of the computations have to be carried-out in double precision.
- The memory interface should be able to sustain a bandwidth of about 0.5 Flop per Byte, i.e. data re-use is relatively high compared to other HPC applications.
- A homogeneous domain decomposition is used for parallelisation. Communication with nearest neighbour nodes arranged in a 3- or 4-dimensional torus is usually sufficient. Both high bandwidth as well as small latencies are required since message sizes are small (of the order of 200 Bytes).

## HARDWARE DESIGN

apeNEXT is a custom designed machine optimised for applications from LQCD both in terms of price-performance ratio and power consumption.

The floating-point unit of the custom designed processor executes at each clock cycle a multiply-add operation $a \times b + c$ (a so-called "normal operation"), where $a$, $b$, and $c$

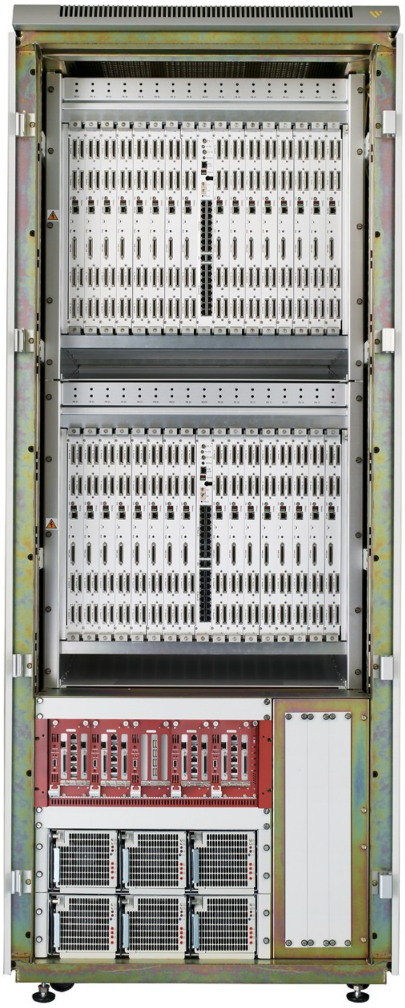

Figure 1: Front view of an apeNEXT tower (photo: Exadron).

[*] Talk given by D. Pleiter at CHEP06, Mumbai (India).

[†] dirk.pleiter@desy.de

| | |
|---|---|
| Nodes/tower | 512 |
| Peak performance | 0.6 TFlops |
| Power consumption | 10 kWatt |
| Flops/Watt | 60 MFlops/Watt |
| Flops/Euro | 0.5 MFlops/Euro |

Table 1: apeNEXT system parameters.

are double precision complex numbers. Running at a clock speed of 150 MHz this results in a peak performance of 1.2 GFlops per processor. Via the memory interface each clock cycle one complex number can be loaded into the processor or stored to memory, i.e. the peak bandwidth is 2.2 GBytes per second. The memory hierarchy is relatively simple. The very large register file consists of 512 64-bit registers, which is sufficient to keep inside the processor all data items that will be eventually re-used. A data prefetch queue allows to hide latencies for both local and remote memory access.

Each processor has $2 \times 7$ on-chip LVDS link modules and is connected by bi-directional links with its nearest neighbours on a 3-dimensional torus. The 7th link is used for connecting some processors to a front-end host system. Via each of the links 8 bits can be transmitted per clock cycle. The apeNEXT network does not only provide a very large bandwidth, but has also an extremely small latency of about 0.1 $\mu$sec. A 16 bit CRC is transmitted after sending one 128 bit word to protect this data path against bit errors.

A schematic overview on the architecture is shown in Fig. 2. Each node has an I2C interface which is used for booting and debugging. One processor per board can be connected to a host interface board via a bi-directional LVDS link. This link is used for fast I/O operations. All nodes are connected to a simple tree network for global logical and interrupt signals. This network is used, e.g., to efficiently evaluate global conditions or to stop the machine in case an exception occurred on any of the nodes.

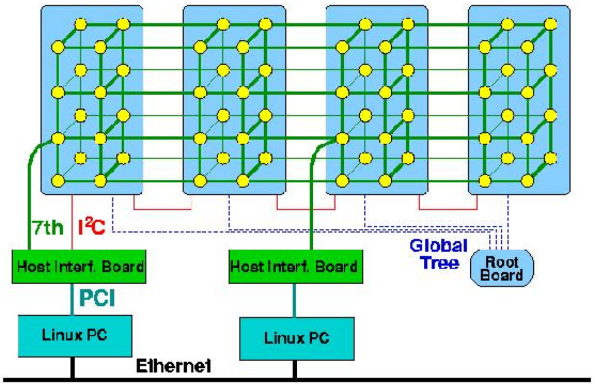

Figure 2: Schematic overview on the apeNEXT system with 4 boards. 16 processors are assembled on each board.

An apeNEXT tower hosts two crates, where each crate consists of a backplane with 16 processing boards and 1 root board. Each of these processing boards has 16 processors. A fully populated apeNEXT tower comprises therefore of 512 processors providing 0.6 TFlops peak performance and 128 GBytes of memory.

A host PC system consisting of 4 blade servers has been integrated in each tower. Two host interface boards are inserted into each blade server providing a total of 48 I2C and 8 LVDS channels. For the apeNEXT installation in DESY Zeuthen each blade server, which acts as a slave PC, has been connected via two bonded Gigabit Ethernet interfaces to a master PC. A schematic overview on the chosen host system interconnect is shown in Fig. 3. Different network layouts can obviously be set-up to suit different user requirements. User jobs are started from the master PC.

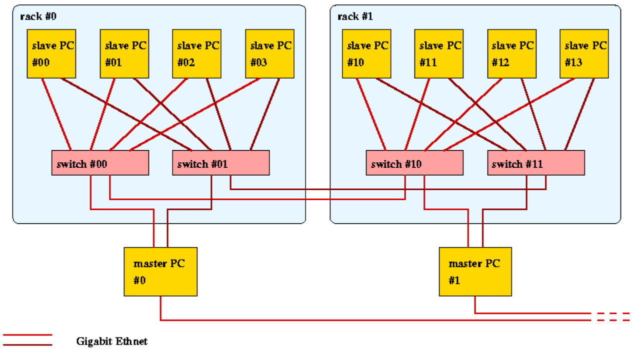

Figure 3: Schematic overview on the interconnect of the host system.

## SYSTEM SOFTWARE

The operating system is distributed on the Linux PCs of the host system and the apeNEXT custom hardware. The functionality of the operating system includes user program loading, I/O operations, system monitoring and debugging. At the transport layer, which physically uses both the I2C and LVDS links, only a few operations had to be implemented: global write, slice write, broadcast read, multi-data read. A schematic overview on the operating system transport layer is shown in Fig. 4. On top of this transport layer a slim protocol layer provides the most essential services like data read and write or file open and close operations. Protocol overhead is minimised by using an encoding which fits into a single 128-bit word.

Programs for apeNEXT can either be written using C or TAO, a FORTRAN-like programming language based on a dynamically grammar allowing the user to define objects and to overload functions. While TAO has been the only programming language available for previous generations of APE computers, the availability of a C compiler is a new feature of apeNEXT. Both compilers generate a high level assembly code which still hides many of the machine de-

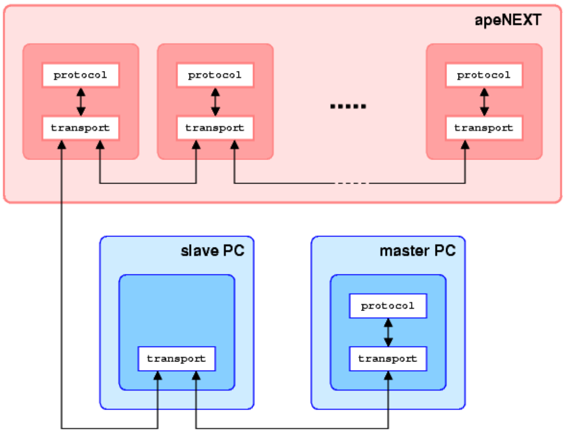

Figure 4: Schematic overview on the operating system transport layer, which is distributed on the apeNEXT custom hardware (upper part) and the host system (lower part).

tails. An assembly preprocessor was implemented to translate the high level to low level machine assembly. At this stage the compiler chain includes an optimising software, `sofan`. The optimisations performed by sofan aim on improving instruction parallelism which are difficult to implement in the compilers due to a lack of information on the resource usage at this level. Currently implemented optimisations include merging multiply-adds into normal operations, removing dead code and register copies, register renaming, merging of address computations.

Hardware and software have been benchmarked for various application kernels. In particular, we investigated the performance of the most important kernel, the multiplication of a vector by the fermion matrix. For the widely used Wilson-Dirac formulation of this matrix we were able to sustain 54% of the peak performance, even if we distributed our problem on the maximum possible number of nodes.[1] In this case, only during 4% of the clock cycles the processor is stalled waiting for data to arrive across the network. This clearly demonstrates the excellent scaling of this application on the apeNEXT architecture.

## DEPLOYMENT AND INITIAL OPERATION

Deployment of apeNEXT production systems started in October 2005 and is expected to be completed by mid 2006. A total of 25 apeNEXT towers procured by INFN (Italy), DESY Zeuthen (Germany), University of Bielefeld (Germany) and CNRS (France) will be operated at three different sites. These machines will provide an aggregated compute power of 15 TFlops, which is mostly dedicated to simulations of Lattice QCD.

Since putting the first machine into operation at DESY Zeuthen[4], we have been able to reach a stable production mode. To qualify the hardware and eliminate any weak nodes, the machines were extensively tested for weeks by running dedicated hardware tests and full physics production codes in replication mode. The latter allowed us to verify that different machines re-produce strictly bit-identical results.

From the experiences during the initial months of operation we are optimistic to reach a similar level of stability as for previous generations of APE machines. Typical jobs run for 12-24 hours on 256 processors. From the rather short period of operation we expect the mean time between interventions to be significantly larger than one month per machine.

## ACKNOWLEDGEMENTS

We would like to acknowledge all people that have collaborated during the last years to the apeNEXT project. In particular R. De Pietri, F. Di Renzo, W. Errico, H. Kaldass, G. Magazzù, N. Paschedag and L. Sartori. We also acknowledge the very fruitful collaboration with Exadron, an Eurotech Company, that has manufactured many of the hardware components and has done final system integration.

## REFERENCES

[1] F. Belletti *et al.* [APE collaboration], "Computing for LQCD: apeNEXT," Computing in Science and Engineering, Vol. 8, No. 1 (2006).

[2] F. Bodin *et al.* [APE collaboration], " Status of the apeNEXT project," Nucl. Phys. Proc. Suppl. **119** (2003) 1038;

[3] F. Bodin *et al.* [APE collaboration], "APE computers – past, present and future," Comp. Phys. Comm. **147** (2002) 402.

[4] `http://www-zeuthen.desy.de/apewww/apenext.html`

[1] The maximum possible number of nodes depend on the problem size. Typically applications will be executed on 256-1024 processors.