The teraflop supercomputer APEmille: architecture review and project status report ⋆

F.Aglietti, A.Bartoloni, C.Battista, S.Cabasino, M.Cosimi, A.Michelotti, A.Monello, E.Panizzi, P.S.Paolucci, W.Rinaldi D.Rossetti, H.Simma [1] , M.Torelli, P.Vicini

INFN - Sezione di Roma I, piazzale A. Moro 2,I-00185 Rome, Italy

N.Cabibbo

ENEA - Ente per le Nuove tecnologie, l'Energia e l'Ambiente, Roma, Italy
INFN - Sezione di Roma I, piazzale A. Moro 2, I-00185 Rome, Italy

W.Errico, S.Giovannetti, F.Laico, G.Magazzù, R.Tripiccione

INFN - Sezione di Pisa, via Livornese 582/a, I-56010 S.Piero a Grado (PISA), Italy

APEmille is a SPMD parallel processor under development at INFN. It is the third generation of the APE family of parallel processors. APEmille is suited for *grand challenges* computational problems such as QCD simulations, climate modelling, neural networks, computational chemistry, numerical wind tunnels, seismic and combustion simulations. Its 1 Teraflop/s peak performance and its architecture, together with its language features, allow such applications to execute effectively. APEmille is based on an array of custom arithmetic processors arranged on a tridimensional torus. The processor is optimized for complex computations and has a peak performance of 528Mflop at 66MHz. Each replica has 8 Mbytes of locally addressable RAM. Two other custom processors are used for program flow control, global addressing and inter node communications. APEmille is accessed over a PCI interface. A network of PCs act as the host computer. APEmille is mainly programmed in The TAO language, a powerful easy-to-ues extensible language. A C++ compiler for APEmille will also be developed.

⋆ Talk presented at CHEP97 (Berlin, April 1997)
[1] on leave from: DESY-IfH,15735 Zeuthen, Germany

Preprint submitted to Elsevier Preprint *29 July 1997*

1 INTRODUCTION

APEmille is a parallel processor oriented to floating point intensive computation and suitable for massively parallel homogeneous problems. It is a SPMD (Single Program Multiple Data) array processor with local addressing capability. It has a basic tridimensional toroidal interconnection network implementing first neighbour communications with low latency and high bandwidth. Long distance communications and more general routing capabilities are also present, although the performance is lower in these cases.
APEmille is the last generation of APE supercomputers family following Ape [1] and Ape100 [2,3]. This INFN project [4,5] aims to develop a teraflops processor for Quantum Chromo Dynamics (QCD) complex systems and fluid dynamics simulation and modelling. APEmille performance is therefore optimized in these areas. The APEmille programming language is also tuned towards these applications. This machine is however well balanced also for many other homogeneous problems, eg. seismic migration, atmosphere and climate modelling, computational chemistry, combustion simulation, image processing. Thus, many of the computatinal *grand challenges* can be tackled using this architecture.

2 APEmille ARCHITECTURE

APEmille is a tridimensional torus of processing elements with distributed data memory and periodic boundary conditions. Each node (PE) is directly connected to its six neighbours. A communication network establishes communications between distant nodes. The grid of APEmille processing nodes is organized as a set of adjacent *cubes* of size $2 \times 2 \times 2$ in which the vertices corresponds to the PEs. APEmille configurations range from a single cube to 256 cubes with topology $32 \times 8 \times 8$.
APEmille is based on custom processors. These are a sort of VLIW-like (Very Long Instruction Word) processors, in which each field of the instruction word drives a different operational block inside the processor, thereby exploiting internal parallelism.

Thee APEmille architecture uses a Harvard architecture: program memory and data memories are physically different; there is one program memory, while the data memory is distributed, each node has its own one.
There are three distinct processors in the APEmille architecture: *Tarzan* drives the program flow and issues the global addresses to the PEs; *Jane* is the node processor, performing floating point and integer computations; *Comm_1000* is the communication processor. Tarzan and Comm_1000 are replicated every eight Janes.

Tarzan drives the program flow for its eight nodes taking into account *global conditions* (results of its own comuptations) as well as *local conditions* (those computed by the Janes). Tarzan computes global addresses and delivers them to the nodes. At each clock cycle Tarzan can send one instruction word and one global address to the Jane processors.
Jane is the processing element. It is a VLIW (in the sense explained above) pipelined processor. It performs abasic operation called `normal`$(R = a \times b + c)$ on real (single and double precision) and complex numbers (32 bit single precision). Jane is able to start a new normal operation at each clock cycle. It has an eight-stage (six-stage) pipeline for double precision (single precision). Integer computations, as well as logical bitwise operations, are performed and use a different pipe whose length is 2 clock cycles.

Jane is able to add a local offset to the global address received from Tarzan to obtain a local address for its data memory. Moreover Jane can evaluate local conditions, using the result to inhibit (locally) the execution of portions of code, or to participate to Tarzan's local aggregate evaluations.
Comm_1000 is the communication processor. It performs several kinds of inter node communications: fixed distance homogeneus communications (both first neighbour and longer distance communications), broadcast communications (one node or a few nodes in different planes of the grid are data sources, while all the others are receivers), and (in a future APEmille enhancement) non-homogeneus communications, in which each node can send (or ask for) data to (from) a node at an arbitrary distance.
APEmille is connected to a network of conventional PCI based computers that acts as the host computer. There is one such computer for each subcrate (set of 32 PEs).
Three kinds of software tools will be provided with APEmille: the language related tools (cross compilers, optimisers, libraries), the Operating System related tools (the monitor program and the graphic symbolic debugger) and the Development tools (the simulator and the graphic profiler).

3 HARDWARE IMPLEMENTATION

An APEmille cube, the building block composed of eigth processing elements, is actually a single printed circuit board where Tarzan, Comm_1000 and eigth Janes are assembled. The memory banks for program and data memories are also plugged on this board.
The APEmille custom processors are designed by the APE Group and implemented as standard cell ASIC design metodology.
Tarzan is capable of integer computations and has two arithmetic units: the ALU and the AGU (Address Generation Unit), a dedicated device for address computation. Tarzan has a register file (multiport internal RAM) with

64 register (32 bits each). It has its own static data memory (256KBytes static RAM) and drives the progam memory, which distribute program words to Tarzan itself and to the Jane processors. SDRAM (synchronous dynamic RAM) memory technology is used for program memory. Due to the VLIW architecture of the processors the program memory is $512KWords \times 160bits$ each. Most of these bits drive Jane internal devices while the remaining ones will drive Tarzan devices.
As already remarked, Jane is used for integer and floating point computations. It has a 512-deep register file with five ports. The register file based architecture allows one `normal` and one I/O operation at each clock cycle. JANE's data memory, based on SDRAM technology, ranges from 2 to 8 MWord/node (from 8 to 32 MBytes/node).

Jane complies with the IEEE standard for floating point numbers and delays the rounding of $a \times b$ to after the sum, i.e. it rounds only the result of (the infinitely precise) $a \times b + c$.
Comm_1000 is a bitsliced device: four identical chips are required on each board to drive the communication of the 32bit words exchanged among the Janes. Each chip manages one fourth of the word.
APEmille clock frequency is 66MHz. When dealing with complex arithmetics where a normal operation consists in 8 floating point operations, Jane's peak power is 528Mflop/s. Performances for the different configurations scale linearly: 4 Gflops for a single board and 1.081 Tflops for the full APEmille configuration. The PCI interface (named APEnic) will provide an I/O bandwidth of 133MB/s.

4 LANGUAGES & PROGRAMMABILITY

The main constraint when dealing with code developers who are scientists rather than professional programmers is that they prefer to concentrate on the physical problem and the algorithm used to solve it rather than on the details of its program implementation.

We are trying to build an user-friendly environment for this class of users in two different ways:

- implementing standard languages – although with some non-standard features linked to the non-standard architecture; these are necessary to let people reuse code developed on different platforms;
- designing new languages that try to solve some problems associated with the new general purpose languages.

To satisfy these constraints we planned to implement two languages on this

machine: the C++ language and the TAO language.
The C++ compiler will have some extensions to exploit APEmille features; it will partially implement the standard.
The TAO language is the language that we designed already for Ape100, and that we are now refining for APEmille.
We are developing PICO, a machine independent optimizer which will be the core of the APEmille high level language compilers.
TAO is an extensible language which has fortran flavour and, as its main characteristic, allows the user to define new operators and new statements or overload old ones. Using the TAO language in its native form still remains possible and it becomes very natural to mix the new statements and operators to the basic ones to produce compact but expressive code. An example of extension, on which we spent much effort, is the QCD header file which extends the TAO language with data structures and operators used in QCD simulations.

5 APEmille STATUS REPORT

The hardware and software architecture of the machine is fully defined and custom processors description is almost completed.
The APEmille simulator, named JUNGLE, is now able to simulate the whole machine in very deep details allowing APEmille users to develop code and evaluate performance of the real machine.
The physical components of APEmille (VLSI devices, board architecture and PCI interface) were described using the VHDL language which is easily interfaced with the JUNGLE simulator to test the electrical behaviour of the APEmille devices.
The prototypes of the APEnic PCI interface and of the backplane as well as a "test board", a reduced APEmille processing board useful to test technology solutions, will be completed and tested soon.
The whole APEmille supercomputer will be assembled in two step: from the beginning of 1998, we will produce and test a 256 nodes system (first APEmille prototype) followed, in a few months, by the assembly of the whole APEmille system.

References

[1] P.Bacilieri et al. The APE project: a Gigaflop Paraller Processor for Lattice Calculations, Computing in High Energy Physics 85 (Elsevier Science Publ. B.V., North-Holland, 1986)

[2] A.Bartoloni et al., A Hardware Implementation of the Ape100 Architecture, Int. Jou. Mod. Phys. C 4 (1993) 995

[3] A.Bartoloni et al., The Software of the Ape100 Processor, Int. Jou. Mod. Phys. C 4 (1993) 969

[4] A.Bartoloni et al., APEmille Proposal, INFN internal document, 1994

[5] A.Bartoloni et al, Addendum to the APEmille Proposal, INFN int. doc., 1995