Mont-Blanc work
Past, present & future

Etienne WALTER, Project manager (Bull/Atos)
Coordinator of Mont-Blanc phase 3
PAST
Mont-Blanc objectives

Vision: leverage the fast growing market of mobile technology for scientific computation

- Mont-Blanc
  - Proof of concept: HPC computing based on mobile embedded technology

- Mont-Blanc 2
  - Extend the concept and explore new possibilities

- Mont-Blanc 3
  - Prepare an industrial solution
  - Test market acceptance
Mont-Blanc

Motivation

- Mobile devices
  - Low power
  - Lower frequencies
  - Clock gating mechanisms
  - Low cost
  - Mass market
- Increasing performance
  - Good ratio
  - Performance / power
  - Performance / $
- Many missing features
  - Low core count
  - Poor network
  - Lack of software

Demonstrated

- Parallel system software
  - Cluster installation & management
  - Runtime
  - Tools
- Application scalability
  - Even with very low end component technologies

Mont-Blanc - Orap presentation
March 2017
The Mont-Blanc prototype ecosystem

Prototypes are critical to accelerate software development

System software stack + applications

- PRACE prototypes
  - Tibidabo
  - Carma
  - Pedraforca
- Mini-clusters
  - Arndale
  - Odroid XU
  - Odroid XU-3
  - NVIDIA Jetson
- Mont-Blanc prototype
  - Operational since May 2015 @ BSC
  - 1080 compute cards
  - Dual Cortex-A15
  - GPU Mali-T604
  - 4 GB LPDDR3
  - Up to 64 GB local storage
  - USB-to-Eth network
  - Fine grained power monitoring system
- ARM 64-bit mini-clusters
  - APM X-GENE2
  - Cavium ThunderX
  - NVIDIA TX1

2012 2013 2014 2015 2016 2017
The Mont-Blanc prototype

**Exynos 5 compute card**
- 2 x Cortex-A15 @ 1.7GHz
- 1 x Mali T604 GPU
- 6.8 + 25.5 GFLOPS
- 15 Watts
- 2.1 GFLOPS/W

**Carrier blade**
- 15 x Compute cards
- 485 GFLOPS
- 1 GbE to 10 GbE
- 300 Watts
- 1.6 GFLOPS/W

**Blade chassis 7U**
- 9 x Carrier blade
- 135 x Compute cards
- 4.3 TFLOPS
- 2.7 kWatts
- 1.6 GFLOPS/W

**Rack**
- 8 BullX chassis
- 72 Compute blades
- 1080 Compute cards
- 2160 CPUs
- 1080 GPUs
- 4.3 TB of DRAM
- 17.2 TB of Flash

- 35 TFLOPS
- 24 kWatt
Application based fault tolerance

Issue: Provide resiliency in an unprotected system

➔ Fine grained task-based checkpointing within the runtime (FTI)
  ▪ Error recovery

➔ Application Based Fault Tolerance
  ▪ Error detection
  ▪ Exploit “unused” bits of data (pointers) for storing check-sum of data

➔ Full chain tested on a real application

✓ Successful experiments on x86, ARMv7 and ARMv8
✓ Compensating hardware limitations at software level
Issue: Improve the Mont-Blanc power monitoring infrastructure

- Fine grained power monitoring infrastructure
  - Development of energy aware job scheduling policies
  - Improvement of tools

- Integrated with standard tools
  - Paraver to correlate performance and power consumption
  - SLURM plugin for jobs energy accounting

✓ Development of energy aware job scheduling policies
✓ Improvement of tools
Memory fault statistics and analysis

Fact: Memory of Mont-Blanc prototype is not ECC protected
Can we survive with this? What do we learn from this?

- Developed a simple in-house memory scanner
- Scanning user-space (~3GB/node) when nodes are idle
- Collected statistics for 13 months / 12 Tbytes/hours

After data analysis, a model of “quarantine technique” has been developed.
MTBF from 2.1 hours to 156.9 hours.
Affecting 0.5% of the cluster resources.

99% of the errors are concentrated in 3 nodes

Cf Unprotected computing: a large scale study of DRAM raw error rate on a supercomputer
L. Bautista, F. Zyulkyarov, O. Unsal, S. McIntosh-Smith
### Mont-Blanc 1/2 observations

<table>
<thead>
<tr>
<th>Overall need of tighter co-design</th>
<th>Between applications, system integrator and technology providers (IP, SoC, runtime...)</th>
</tr>
</thead>
</table>
| Hardware architecture             | • Node architecture is a key component  
• A more balanced & more aggressive approach is needed for full exploitation of benefits |
| Software needs to evolve from latency-oriented applications to throughput-oriented | • Fundamental system and designer mentality  
• Need to move towards task-based models, asynchrony, overlap communication/computation  
• With minimum code refactoring cost |
PRESENT
New challenges with the diminishing technology scaling observed (on core performance, frequency …)
  ▪ need to leverage new paradigms

How to optimize the global compute efficiency?
  ▪ find optimal balance for
    • I/O bandwidth & latency, memory, compute nodes
  ▪ bypass unnecessary latencies
  ▪ investigate benefit of heterogeneous architectures
Going forward: Technical challenges (2/2)

- Assess planned architecture with simulation work
- Get prepared for massive and efficient parallelism
- Enhance the software environment
  - run times (OmpSs/OpenMP), tasks operations (MPI), scheduler, compiler & math libraries
- Always keep in mind applications
  - assess solution with real applications
  - enhance performance & energy monitoring tools (MAQAO, system level)
  - hide/minimize the impact of HW architecture
Mont-Blanc 3 project key objectives

- Design a compute node based on ARM architecture for a pre-exascale system
  - Well balanced: Memory, Interconnect, IO
  - Simulation will be used to evaluate the options on applications

- Evaluate new high-end ARM core and accelerator, and assess different options for compute efficiency
  - Heterogeneous cores, new option for VPU, high performance core
  - Some assessment with existing solutions will be done using applications
  - One key idea: prepare to transform applications from being latency limited to throughput limited

- Develop the software ecosystem needed for market acceptance of ARM solutions
Work domains

- Computing Efficiency
- Architecture
- Balanced Architecture
- Applications
- Sw Environment
- Hw. Platform (test cluster) & mini clusters
- Simulation
### WP5 Simulation Tools: evaluation of existing tools

<table>
<thead>
<tr>
<th>Name</th>
<th>on-line/offline</th>
<th>scope</th>
<th>granularity</th>
<th>speed(up)</th>
</tr>
</thead>
<tbody>
<tr>
<td>gem5 (classic memory)</td>
<td>on-line</td>
<td>core/SoC</td>
<td>u-arch/insts</td>
<td>~100-200 KIPS</td>
</tr>
<tr>
<td>Elastic trace (gem5)</td>
<td>off-line</td>
<td>interconnect/memor y system</td>
<td>u-arch/insts</td>
<td>~ 7x (gem5)</td>
</tr>
<tr>
<td>SimMATE (gem5)</td>
<td>off-line</td>
<td>memory system</td>
<td>u-arch/insts</td>
<td>~6x – 800x (gem5)</td>
</tr>
<tr>
<td>Garnet (gem5)</td>
<td>on-line</td>
<td>interconnect</td>
<td></td>
<td>~0.2x (gem5)</td>
</tr>
<tr>
<td>TaskSim</td>
<td>off-line</td>
<td>compute node/task scheduler</td>
<td>arch/tasks</td>
<td>~ 10x native (burst)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>~20x gem5 (memory)</td>
</tr>
<tr>
<td>Dimemas</td>
<td>off-line</td>
<td>cluster/off-chip network</td>
<td>bursts/messages</td>
<td>“very fast”</td>
</tr>
<tr>
<td>SST/macro</td>
<td>on-line</td>
<td>cluster/off-chip network</td>
<td>bursts/messages</td>
<td>~ 0.3-3x native</td>
</tr>
</tbody>
</table>
Multi-scale simulation workflow plan

- **SimMATE**
- **Elastic trace**
- **gem5 (on-line)**
- **TaskSim**
- **Dimemas**
- **MUSA**
- **SST/macro**

Scope:
- core/SoC
- multi-core
- compute-node
- cluster

Granularity:
- u-arch/insts
- arch/insts
- bursts/tasks
- inter-proc messages
Objective:
- Provide input for co-design:
  - key input for simulation
- Search how to adapt programming practice to the throughput age
  - develop and promote best practices/recommendations on programming methodologies
- Demonstrate architecture implementation

Sources:
- Industrial applications (AVL)
- Applications of interest at Supercomputing centers
- Relevant Mini-apps from all over the world

Main achievements
- Comprehensive study of a large number of applications
- Porting & evaluation on ARM v8 platforms
## HPC ARM Software Stack

<table>
<thead>
<tr>
<th>Applications (source files – C/C++/Fortran/Python etc.)</th>
<th>Compilers</th>
<th>Scientific libraries</th>
<th>Developer tools</th>
<th>Cluster management</th>
<th>Runtime libraries</th>
<th>Hardware support</th>
<th>Operating System (Linux kernel-based)</th>
<th>(Heterogeneous) Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernels</td>
<td>GCC</td>
<td>ATLAS</td>
<td>ATLAS</td>
<td>SLURM</td>
<td>OpenMP</td>
<td>Power management</td>
<td>Ubuntu</td>
<td>CPU</td>
</tr>
<tr>
<td>Mini-applications</td>
<td>LLVM</td>
<td>ARM Mathematics Libraries</td>
<td>Perf</td>
<td>OpenLDAP</td>
<td>Nanos++</td>
<td>Lustre</td>
<td>OpenSUSE</td>
<td>CPU</td>
</tr>
<tr>
<td>Full applications</td>
<td>Pathscale</td>
<td>BLIS</td>
<td>MAQAO</td>
<td>NTP</td>
<td>CUDA</td>
<td>NFS</td>
<td>CentOS</td>
<td>CPU</td>
</tr>
<tr>
<td></td>
<td>Mercurium</td>
<td></td>
<td>Allinea</td>
<td>Ganglia</td>
<td>MPI</td>
<td></td>
<td>Fedora</td>
<td>GPU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Drivers</td>
<td>Accelerator</td>
</tr>
</tbody>
</table>

### Compilers
- GCC
- LLVM
- Pathscale
- Mercurium

### Scientific libraries
- ATLAS
- ARM Mathematics Libraries
- BLIS

### Developer tools
- Extrae
- Perf
- MAQAO
- Allinea

### Cluster management
- SLURM
- OpenLDAP
- NTP
- Ganglia

### Runtime libraries
- OpenMP
- Nanos++
- HSA
- CUDA
- MPI

### Hardware support
- Power management
- Lustre
- NFS
- DVFS

### Operating System (Linux kernel-based)
- Ubuntu
- OpenSUSE
- CentOS
- Fedora
- Drivers

### (Heterogeneous) Hardware
- CPU
- GPU
- Accelerator
- Network

### Extrae
- Perf
- MAQAO
- Allinea

### Scientific libraries
- ATLAS
- BLIS

### Developer tools
- Extrae
- Perf
- MAQAO
- Allinea

### Cluster management
- SLURM
- OpenLDAP
- NTP
- Ganglia

### Runtime libraries
- OpenMP
- Nanos++
- HSA
- CUDA
- MPI

### Hardware support
- Power management
- Lustre
- NFS
- DVFS

### Operating System (Linux kernel-based)
- Ubuntu
- OpenSUSE
- CentOS
- Fedora
- Drivers

### (Heterogeneous) Hardware
- CPU
- GPU
- Accelerator
- Network

**Porting PGI-Flang**
- Optimised ARM mathematics libraries

**Porting MAQAO**
- OpenMP+OmpSs heterogeneous support improvements
- OmpSs accelerator support improvements
- LLVM+GNU OpenMP optimisations
- Porting TinyMPI
- OmpSs and MPI communication/computation overlap
- Topology-aware MPI

**Power management improvements**
- Linux kernel heterogeneous support improvements
- memcpy-optimisations techniques via RDMA

**+ Contribution to OpenHPC (from 1.2)**
Mont-Blanc - Orap presentation

March 2017

Computing Efficiency

Architecture

Balanced Architecture

Applications

Sw Environment

Hw. Platform (test cluster) & mini clusters

Simulation
Compute Efficiency Effort

Objective: propose more aggressive architectural designs
The focus is on:
- increasing computation throughput
- optimizing energy efficiency

Three lines of effort are followed:
- Development and evaluation of a symbiotic combination of different types of compute cores (big.LITTLE, in-order / OoO)
- Analysis of ARM compute acceleration technologies (such as SVE) and optimisation of the memory system
- Evaluation of existing SoCs
Exploiting run-time priorities from the NoC

- **Hypothesis:** Improve the performance of OmpSs applications by using QoS in the NoC favouring tasks in the critical path.

- **Goal:** Reducing packet latency and, consequently, miss penalty of memory instructions of critical tasks.

- **Method:** Generated packets (triggered by L1 cache misses) will include priorities (already used by Nanos’ scheduler) for QoS arbitration in NoC routers.

- **Background:**
  - Adjust core frequency based on task criticality.
  - Additional support to change cores frequency using priorities.
  - We employ the same infrastructure to assign the priority of each L1 cache miss.

*Credit: CATA: Criticality Aware Task Acceleration for Multicore Processors (Castillo, E. et al. @ IPDPS’16).*
The new test-platform

PRACE prototypes
- Tibidabo
- Carma
- Pedraforca

Mini-clusters
- Arndale
- Odroid XU
- Odroid XU-3
- NVIDIA Jetson

Mont-Blanc prototype
- Operational since May 2015 @ BSC
- 1080 compute cards
- Dual Cortex-A15
- GPU Mali-T604
- 4 GB LPDDR3
- Up to 64 GB local storage
- USB-to-Eth network
- Fine grained power monitoring system

ARM 64-bit mini-clusters
- APM X-GENE2
- Cavium ThunderX
- NVIDIA TX1

Mont-Blanc 3 Test platform
- Cavium ThunderX2
- Bull Sequana architecture
Integration in industrial design (Bull sequana island)
Bull sequana compute blade with Cavium ThunderX2 cpu

Interconnect Mezzanine

Cavium ThunderX2 CN99xx ARM V8 Processor
Test-platform objective

- Integration test-bed for sw infrastructure & applications
- Validation of simulation models
- Test customer acceptance
- Generate further requirements

to be deployed in H2 2017
FUTURE
Towards a market driven approach

- Prepare a solution than can be brought to the market
  - assess what would still be missing/needed to fulfill customer expectations
  - find right cost/features balances
  - adapt an industrial integration approach, preparing scalability and volume production
  - set-up industrial partnerships & ecosystem

- Keep the right timing
  - Time-to-Market
  - but development cycles are heavy & risky

- Get prepared for next steps
  - acceptance of ARM based solutions
  - approach the market with project results
Further steps (within or beyond current scope of work)

- Continue effort on software stack
- Continue SoC design work
  - Memory issues: work on NVM/HBM/3D-memories
  - Interconnect integration
- Work on data (storage/movement)
- Work on accelerators
- Secure the SoC provisioning
  - reinforce the partnership with our provider (Cavium)
  - evaluate alternative sources
  - consider a European sourcing (funding issue)
Our vision:
- Need to permit « good independence » between apps & HW levels
  - (SVE’s vector-length agnosticism is a good example)
  - need to provide programmers with the right programming paradigms
  - kind of “loose coupling” model between apps & hw developments
- One single solution cannot optimally fit all needs, all Apps
  - need for an architecture flexible enough to permit and facilitate adaptation

We are (co)operating in a wide environment
- European projects & stakeholders (POP, Sage projects for instance)
- Non-European projects: US (Argonne labs), Japan …

Exascale is a collective journey
- we’re probably just in the middle
- & Europe must take a significant part
Thank you for your attention

etienne.walter@atos.net

Credits: all Mont-Blanc partners (third phase, but also first & second phases)

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement n° 671697