Supercomputing for the “Small” Masses

BSC/UPC CUDA Center of Excellence
Barcelona Supercomputing Center (BSC-CNS) and Universitat Politecnica de Catalunya (UPC)

I. The “Small” Market

Smartphones and tablets are becoming pervasive devices; 712.6 million Smartphones were sold worldwide in 2012, a 44.1% more devices than the 494.6 million sold in 2011, and 133.9% more than the 304.7 million sold in 2010. Similarly, 106.1 million tablets were sold 2012, a 54.4% more units than in 2011, when 68.7 million tablet devices were sold. If we compare these figures to the 3.5 million personal computers (both desktop and server) sold in 2012, we see that during the last year, the mobile processor market outnumbered the traditional desktop and sever market by a factor of 234X.

Such a huge difference in sales between mobile devices and traditional computers enables the economy of scale of processor manufacturing to offer lower prices on mobile processors than in desktop and sever chips. For instance, a NVIDIA Tegra 3 Multi Processor System on Chip (MPSoc) costs less than $80, while an Intel server processor is around $1000, and its desktop counterpart around $300. This difference makes devices built using mobile processors more attractive to customers.

In BSC/UPC we believe that mobile processors will become dominant, in the very same way microprocessors did in the eighties. This believe has lead us to pursue building a system around a mobile MPSoC that can be used as building block in High Performance Computing (HPC), server, desktop, and mobile environments. For this to happen, regular CPU cores cannot be solely used, but accelerators to offload compute intensive workloads are required. We have worked with NVIDIA and SECO to build the first system that integrates mobile processors (a Tegra 3 MPSoC) and CUDA-capable GPUs (Quadro 1000M): the CARMA Kit. Our final goal is not only bringing CUDA to mobile devices, but also bringing mobile processors to desktop, server and HPC systems.

II. The First Small Step

The ARM architecture is currently the main player in the mobile processor world, being the architecture implemented by chip manufacturers, such as NVIDIA, Samsung, Apple, Freescale, or ST. Most Smartphones and tablets are built around chips which include one or several ARM cores, where the operating system and the applications execute. This is why we have focused on the ARM architecture.

The path to bring together an ARM MPSoC and a CUDA-capable GPU started with the SECOCQ7-MXM carrier board. This board allows plugging in QSeven daughter boards, such as the SECO QuadMo747-X/T20 that includes a NVIDIA Tegra 2 MPSoC integrating two ARM Cortex A-9 processors running at 1GHz. It has been already showed that both Smartphones and tablets can be built using ARM mobile MPSoCs, so there is little interest in benchmarking our initial ARM board for those environments. We have focused on running HPC workloads on the Tegra 2 MPSoC to identify those potential bottlenecks that could prevent mobile processors from being used in HPC and desktop environments.

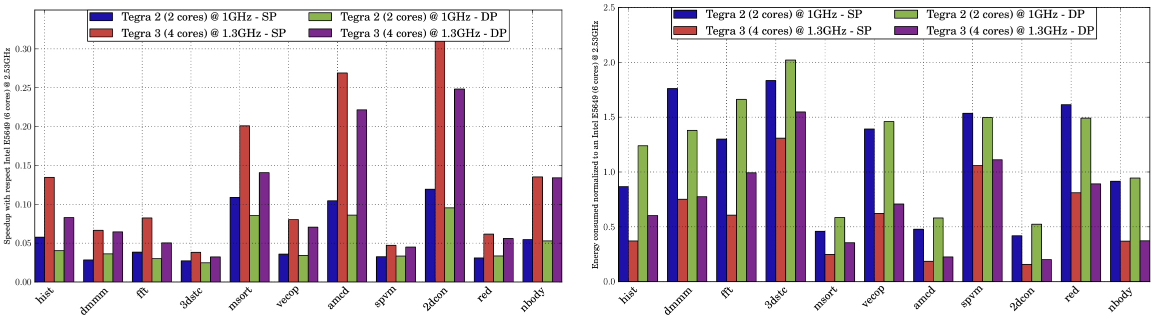
A. Single Board Desktop Systems

Desktop systems are typically built around a single multi-core chip. To test if the Tegra 2 and Tegra 3 alone are sufficient to build a desktop system we developed eleven micro-benchmarks that stress the computation capabilities of the MPSoC: *Vector Operation* (vecop), *Dense Matrix-Matrix Multiplication* (gemm), *3D Stencil* (3dstc), *2D Convolution* (2dcon), *1D Fast Fourier Transform* (fft), *Reduction* (red), *Histogram* (hist), *Merge Sort* (msort), *N-Body* (nbody), *Atomic Monte-Carlo Dynamics* (amcd), and *Sparse Vector-Matrix Multiplication* (spvm).

Figure 1(a) shows the performance for each benchmark when running in the Tegra 2 MPSoC, compared to an execution on a single Intel Xeon E5649 @ 2.53GHz. This Figure clearly shows how the performance of mobile MPSoCs is far behind modern high-performance processors. The Tegra MPSoCs are 20X slower than the Xeon chip in compute bound benchmarks (e.g., (*dgemm*, *fft*) and 5X slower for memory bound codes (e.g., *amcd* and *2dcon*. However, these results are much different if the power consumption of the chips being compared is taken into account. Figure 1(b) shows the estimated [1] energy consumed by each processor to compute the solution of each benchmark. These results show that, in most benchmarks, the energy consumption of the Tegra 3 MPSoC is lower than the Xeon chip. The energy consumption is higher for the Tegra MPSoCs in those codes that are extremely compute bound, such as *3dstc*.

Results in Figure 1(b) seem quite promising, but these numbers only reflect the energy consumed by the MPSoC. If we measure the total system power consumption, this is between 2 ~ 2.5 times higher than the MPSoC Thermal Design Power (TDP). This illustrates a peculiarity of mobile processors: the power consumed by the boilerplate electronics is higher than the power consumed by the processor. Our measurements also show that the total power consumption of the Tegra 3 system is similar to the Tegra 2 system, although the MPSoC has twice the amount of CPU cores, and the clock speed is 300MHz higher. This similar consumption is mostly

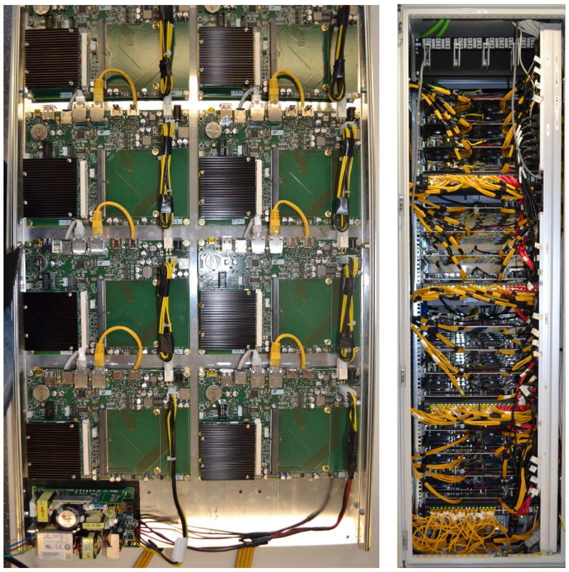
[1] This estimation is based on the TDP



(a) Performance Speedup.

(b) Energy to Solution (estimated).

Fig. 1. Tegra MPSoC compared to a 6-core Intel Processor



(a) Blade.

(b) Rack.

Fig. 2. Tibidabo, the first ARM cluster built ever

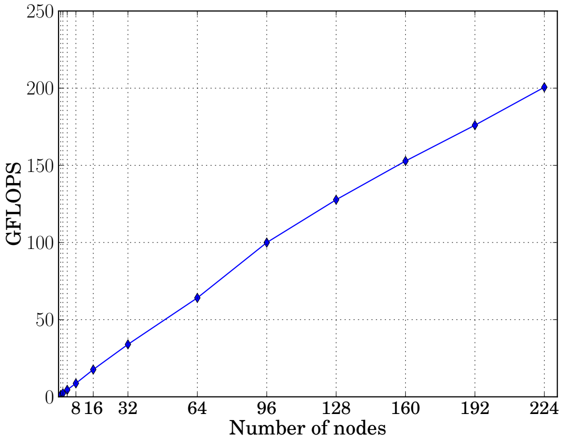


Fig. 3. Performance scalability of HPL on Tibidabo

due to the improvements in the Tegra 3 design, but also due to our benchmarking of Tegra 2 board, that allowed us to point out those peripherals that were consuming energy but were mostly useless. As a result, the SECO board delivered in the CARMA kit does not include a secondary 100Mbps Ethernet controller that was present in the first SECO boards.

During this benchmarking phase we also identified the following issues that were corrected in the CARMA kit:

- Lack of support for SATA hard drives.
- Inability of plugging a discrete GPU to the PCIe bus.

B. Tibidado: The First ARM Cluster Ever

At BSC we continually look for innovative solutions to build HPC systems. As part of this vision, we built a first HPC prototype based on Tegra 2 SECO boards, called Tibidabo. Figure 2(a) shows a blade of Tibidabo, consisting of eight Tegra 2 boards, connected through a 1GbE switch. Sixteen of these blades are connected into a rack, as shown in Figure 2(b), which contains a total of 128 nodes. Tibidabo consists of two racks (256 nodes), and a NFS server that provides a shared filesystem to all nodes.

Each Tibidabo node runs an Ubuntu Linux with kernel 2.6.36, and includes the GNU Compiler Collection 4.4, and 4.6. Job management is done using SLURM, and communication between nodes can be done using MPICH2 1.4. Several applications, including Yales2, Euterpe, SpecFEM3D, MP2C, BigDFT, Quantum Expresso, PEPC, SMMP, ProFASI, and COSMO have been successfully compiled and executed on Tibidabo.

For benchmarking purposes, we have centered our effort in running HPL; Figure 3 shows the performance of HPL when running on Tibidabo for different node counts. HPL has an almost linear scalability on Tibidabo, but a quite low performance of 1 GFLOP per node. As a result, the energy efficiency of Tibidabo is quite low due to the power consumed by peripherals and boilerplate electronics in the board. This shows the need to add components that boosts the performance of systems based on mobile chips using GPUs.

III. CUDA for the "Small"

Our experience using the Tegra 2 SECO board and building Tibidabo served to guide the design of the

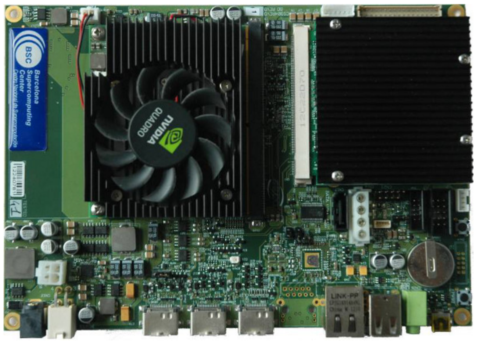


Fig. 4. A CARMA Kit board including a Tegra 3 MPSoC and a Quadro 1000M

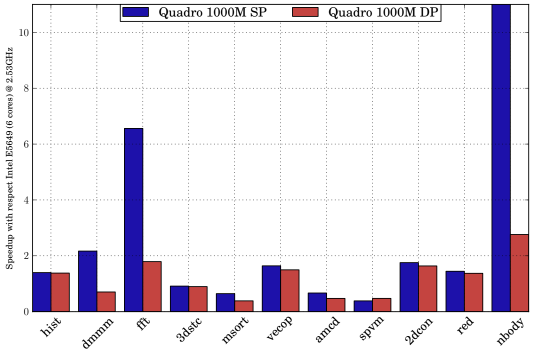


Fig. 5. Performance speedup of CARMA Kit compared to an Intel E5649 (6 core) @ 2.53GHz

CARMA Kit, also built by SECO. Figure 4 shows the CARMA Kit; this board is an evolution of the SECOQ7-MXM board, where the Tegra 2 MPSoC has been replaced by a Tegra 3 chip integrating four Cortex-A9 CPU cores running at 1.3GHz, and a Quadro 1000M GPU has been attached through the PCIe bus.

The CARMA Kit is supported by CUDA 5.0 through a cross-compilation environment, which we have used to execute the BSC benchmarks. Figure 5 shows the performance of these benchmarks when running on the CARMA Kit using the GPU. These results show that the CARMA kit is able to achieve higher single-precision performance than a single CPU chip in those benchmarks where the compute to data transfer ratio is high (*fft*, *2dconv*, and *nbody*). The CARMA kit does not perform as good on double-precision workloads because the Quadro 1000M GPU does not include full double-precision support. This is not an issue for most desktop and mobile systems, but in HPC environments a GPU with full double-precision should be used.

The performance of the CARMA Kit on benchmarks where the computation to data ratio is small is quite poor. The root of this behaviour is the PCIe 4x 1.0 included in the board, which delivers a maximum of 500 MBps. Moreover, the Quadro 1000M GPU does not allow concurrent execution of CUDA kernels and DMA transfers, and only includes one DMA engine. Hence, data transfers and GPU computation are serialized, and the benchmarks cannot exploit the full-duplex capabilities of the PCIe bus. This issues can be solved in mobile and low-cost dekstop systems by integrating the GPU logic inside the MPSoC, and in high-end systems by supporting a full PCIe and using high-end GPUs.

IV. The "Small" Software Stack

The solution that we pursue with this low-power approach does not come without challenges. Systems built from low-power platforms will face a number of challenges that will be exposed to the software.

Applications need to scale to a higher number of compute nodes to achieve competitive performance. Massive parallelism (millions of threads) becomes compulsory for Petascale performance. Low-power components will rely on smaller on-chip memory structures, so applications must improve their data reuse, exposing temporal locality and reducing reuse distance. Applications must also exploit all the on-chip heterogeneous compute devices. Low-power devices will have lower bandwidth I/O interfaces, requiring applications to overlap communication with computation.

We believe that runtime-managed task-based parallel programming models are the best solution to manage that complexity. OmpSs (OpenMP SuperScalar) is the BSC proposal for programming massively parallel heterogeneous systems, and we are currently exploring and targeting to HPC systems built from mobile processors such as the CARMA Kit.

In OmpSs, the programmer partitions the application into potentially asynchronous coarse-grain tasks, and exposes the dataflow across tasks. The tasks can recursively decompose into further finer-grain tasks along the way. From there, it is the responsibility of the runtime management system to detect when a task is ready to execute, allocate it to a compute device, and schedule execution and data transfers. The architecture complexity is hidden, handled by the runtime that detects, manages, and exploits parallelism, synchronization, locality, heterogeneous devices, and distributed memories.

OmpSs also hybridizes well with MPI, since synchronous MPI operations can be encapsulated in an asynchronous OmpSs task, which automatically achieves overlap of communication with computation, and decouples performance from the I/O system characteristics.

V. Conclusions

We believe that the CARMA Kit is a big step towards enabling CUDA to reach a broader range of devices. The work we have done has helped to produce the first hardware that allows executing CUDA applications on ARM platforms, and will help bringing CUDA to mobile devices in the near future. Furthermore, we have also shown that HPC systems can be built using mobile parts and GPUs, and supercomputing workloads can be executed on those systems using novel programming models such as OmpSs.