Low latency network and distributed storage for next generation HPC systems: the ExaNeSt project

IOP Conf. Series: Journal of Physics: Conf. Series **898** (2017) 082045 doi:10.1088/1742-6596/898/8/082045

Low latency network and distributed storage for next generation HPC systems: the ExaNeSt project

R Ammendola[1], A Biagioni[2], P Cretaro[2], O Frezza[2], F Lo Cicero[2], A Lonardo[2], M Martinelli[2], P S Paolucci[2], E Pastorelli[2], F Pisani[456] F Simula[2], P Vicini[2], J Navaridas[3], F Chaix[7], N Chrysos[7], M Katevenis[7] and V Papaeustathiou[7]

[1] INFN, Sezione di Roma Tor Vergata, Italy
[2] INFN, Sezione di Roma, Italy
[3] University of Manchester, England
[4] University of Bologna, Italy
[5] INFN, Sezione di Bologna, Italy
[6] CERN, Switzerland
[7] FORTH - Foundation For Research & Technology, Hellas

E-mail: andrea.biagioni@roma1.infn.it

Abstract. With processor architecture evolution, the HPC market has undergone a paradigm shift. The adoption of low-cost, Linux-based clusters extended the reach of HPC from its roots in modelling and simulation of complex physical systems to a broader range of industries, from biotechnology, cloud computing, computer analytics and big data challenges to manufacturing sectors. In this perspective, the near future HPC systems can be envisioned as composed of millions of low-power computing cores, densely packed — meaning cooling by appropriate technology — with a tightly interconnected, low latency and high performance network and equipped with a distributed storage architecture. Each of these features — dense packing, distributed storage and high performance interconnect — represents a challenge, made all the harder by the need to solve them at the same time. These challenges lie as stumbling blocks along the road towards Exascale-class systems; the ExaNeSt project acknowledges them and tasks itself with investigating ways around them.

1. Introduction

The ExaNeSt project [1], started on December 2015 and funded in EU H2020 research framework (call H2020-FETHPC-2014, n. 671553), is a European initiative aiming at developing the system-level interconnect, a fully-distributed NVM (Non-Volatile Memory) storage and the cooling infrastructure for an ARM-based Exascale-class supercomputer. The ExaNeSt Consortium combines industrial and academic research expertise in the areas of system cooling and packaging, storage, interconnects, and the HPC applications that drive all of the above.

ExaNeSt will develop an in-node storage architecture, leveraging on low-power NVM devices. The distributed storage system will be accessed by a unified low-latency interconnect, enabling scalability of either storage and I/O bandwidth together with the compute capacity. The unified RDMA-enhanced network will be designed and validated using a testbed based on FPGAs and passive copper and/or active optical channels, allowing the exploration of interconnection topologies, congestion-minimizing routing functions and support to system resiliency.

ExaNeSt also addresses packaging and liquid cooling, which are of strategic importance for the design of realistic systems, and aims at an optimal integration which will be dense, scalable and power efficient. In an early stage of the project, an ExaNeSt system prototype, characterized by 500+ ARM cores, will be available acting as platform demonstrator and hardware emulator.

ARM is the industry leaders in power-efficient processor design. ARM-based servers are currently under evaluation as an alternative to the x86 and POWER-based servers which are prevalent in data-centers and supercomputers for both research and business [2, 3, 4]. This technological approach for a scalable and low-energy solution to computing is shared with other projects with the common goal to deliver a European HPC platform: (i) ExaNoDe [5] focuses on delivering low-power compute elements for HPC and (ii) ECOSCALE [6] focuses on integrating FPGAs and providing them as accelerators in HPC systems.

A set of relevant ambitious applications, including HPC codes for astrophysics [7, 8, 9, 10], spiking neural networks simulation [11], engineering [12, 13], climate science [14], materials science [15] and big data [16], will support the co-design of the ExaNeSt system to provide specifications during design phase and application benchmarks for the prototype platform.

In this paper we describe the interconnect technologies and multi-tiered topologies that will be studied during the project. A specific section describes initial work done on the ExaNeSt large scale network simulator required to evaluate different architectural solutions and understand related criticalities. We also provide a status report of the project initial developments.

2. Multi-tiered, scalable interconnects for unified data and storage traffic

The development of an interconnect technology suitable for exascale-class supercomputers is one of the main goals of the project; we envision it as a hierarchical infrastructure of separate network layers interacting through a suitable set of communication protocols. Topologies in the lowest tiers are hardwired due to choices made in the prototype design phase. However, design at the network level is configurable and will be the subject of study throughout the next year.

The *Unit* of the system is the Xilinx Zynq UltraScale+ FPGA, integrating four 64-bit ARMv8 Cortex-A53 hard-cores running at 1.5 GHz. This device provides many features, the following being the most interesting: (i) a very low latency AXI interface between ARM subsystem and programmable logic, (ii) cache-coherent accesses from the programmable logic and from the remote unit and (iii) a memory management unit (MMU) with two-stages translation and 40-bit physical addresses, allowing external devices to use virtual addresses thus enabling user-level initiation of UNIMEM communication. The FPGA block diagram is shown in figure 1.

The *Node* (figure 2) is the Quad-FPGA Daughter-Board (QFDB) containing four Zynq Ultrascale+ FPGAs, 64 GB of DRAM and SSD storage connected through the ExaNeSt Tier 0 network. The inter-FPGA communication bandwidth and latency affect the overall performance of the system. As a consequence, at QFDB level, ExaNeSt provides two different networks, one for low-latency exchanges based on LVDS channels and AXI protocol, the other for high-throughput transmissions through High Speed Serial links (HSS).

For inter-node communication, the QFDB provides a connector with ten bidirectional HSS links for a peak aggregated bandwidth of 20 GB/s. Four out of ten links connect neighbouring QFDBs hosted on the *Mezzanine/Blade* (Tier 1). The first Mezzanine prototype (Track-1), shown in figure 3, enables the mechanical housing of 4 QFDBs hardwired in a 2D cube topology with two HSS links (2×16 Gb/s) per edge and per direction. The remaining six HSS links, routed through SFP+ connectors, are mainly used to interconnect mezzanines within the same Chassis (Tier 2). Furthermore, they can also be exploited to modify the Intra-Mezzanine topology.

The Mezzanine sports additional slots to host thermal mockups to evaluate the liquid-cooled mechanics and optional off-the-shelf ARM-based computing modules, the Kaleao KMAX [17]. Nine such as mezzanines will fit within an 11U (approximate height) "half depth" chassis.

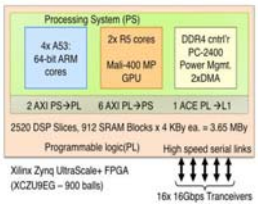

Figure 1. Zynq FPGA.

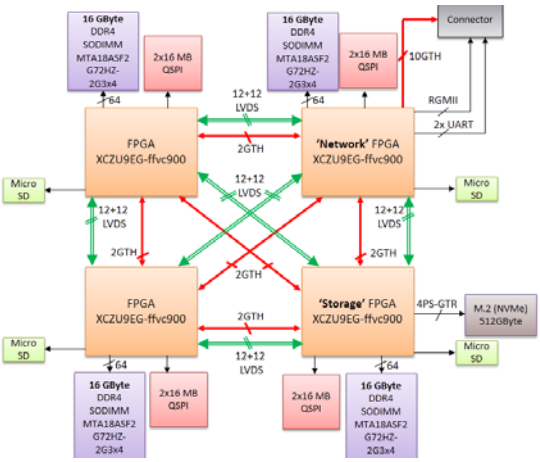

Figure 2. QFDB node.

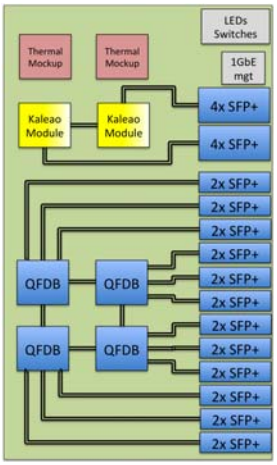

Figure 3. Mezzanine.

3. Topologies

ExaNeSt explores both *direct* blade-to-blade and *indirect* blade-switch-blade networks. The former type, with direct links (Inter-Mezzanine) between blades, is frequently called "switchless" and has been employed in many HPC installations. These interconnects distribute the switching and routing functions to units that are integrated close to computing elements. The latter will be tested connecting the blades to commercially available components, based on ASICs or FPGAs.

Each mezzanine provides 24 SFP+ connectors to communicate with other mezzanines within the same Chassis. So many independent channels allow for a high level of flexibility to experiment with several direct network topologies.

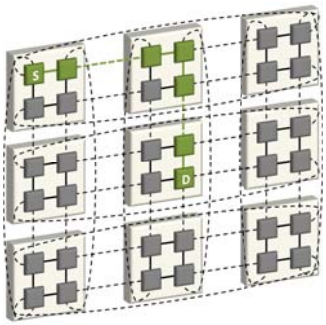

Figure 4. QFDBs within the chassis shape a 2D Torus topology (Tier 1/2).

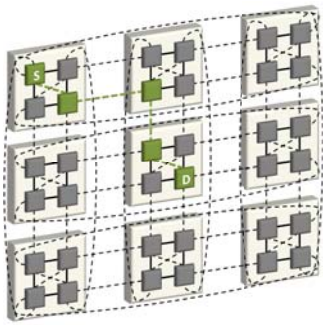

Figure 5. Performance boost due to the intra-Mezzanine (Tier 1) all-to-all topology.

A first scenario is shown in figure 4 where 2D torus topology is chosen to interconnect the QFDBs of the 9 blades of a chassis. The solid and dotted lines are the intra-Mezzanine and inter-Mezzanine I/O interfaces respectively. Since local (within the mezzanine) and remote (neighbouring mezzanine) QFDBs are in the same network hierarchy, 2 HSS per direction for remote channels are used to balance the network capability. A 6×6 Torus topology is the resulting configuration where the longest path consists of 6 hops implementing a Dimension-Order Routing (DOR) algorithm (see section 5).

An additional design option would use the "diagonal" links to interconnect the QFDBs in a mezzanine resulting in a all-to-all topology. With this simple modification — which also requires

the implementation of a more complex routing algorithm — two hops are saved on average, as sketched in figure 5; our estimation for single hop latency is about 200 ns.

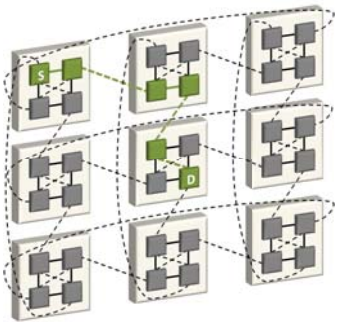

Figure 6. Dragonfly topology interconnecting Mezzanine Supernodes (Tier 2).

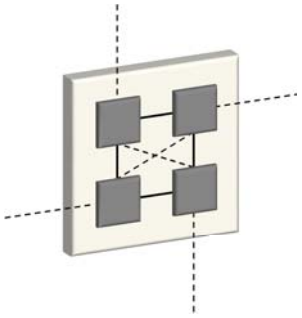

Figure 7. Each QFDB exploits only one SFP+ cable for inter-Mezzanine network.

A second scenario foresees a Dragonfly [18] network implementation as in figure 6. Each blade corresponds to a supernode (figure 7) connected to the neighbouring nodes with just one inter-Mezzanine channel.

A further latency reduction (3 hops for the longest path as depicted in figure 8) is gained by connecting each QFDB of a Mezzanine with their counterparts on neighbouring Mezzanines, shaping four 3×3 2D torus networks (figure 9). Moreover, counterparts QFDBs residing on Mezzanine in neighboring chassis (Tier 3) can be arranged in a 3D torus; in this way we exploit two additional external inter-Mezzanine channels eliminating the diagonal links on the QFDB. Each set of QFDBs is a 3D torus interconnect $3 \times 3 \times C$ where C is the number of chassis.

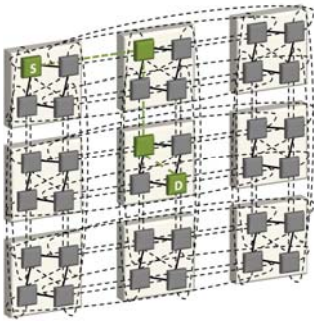

Figure 8. An alternative topology to the simple torus network.

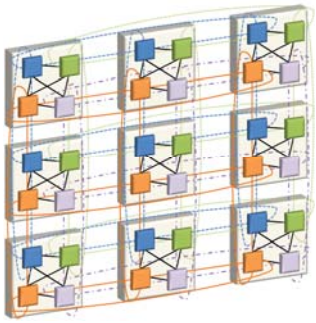

Figure 9. Four 2D torus networks interconnecting the mezzanines.

4. Communication Protocol

The UNIMEM [19] memory model offers a global address space with direct coherent accesses to remote memories. UNIMEM was originally developed in the context of EuroServer [20] project and in ExaNeSt, it will be tested in large-scale ARM-based systems. The UNIMEM architecture is suitable for systems consisting of several *coherence islands*. In ExaNeSt, the four processors inside an FPGA form one such coherence island including processors, their coherent caches (thanks to the adoption of ARM cache coherent interconnect protocol (CCI)), memory and peripheral devices, and one or more external ports for incoming and outgoing accesses.

A global address space (GAS) is a memory space shared among several processors. A single global address in this space is consistently mapped to a physical memory location (or peripheral device) irrespective of which processors among the participants in the GAS issues the transaction. Such a global address space may be physical or virtual. In the latter case, a mapping function, consistently shared across processors, is needed. On the other hand, a global physical address points to a fixed physical location. In this case, the address bits typically identify the location in a hierarchical fashion: the most significant bits identify a node of the system, next comes the physical page within the memory of that node, and last the byte within the page.

The global virtual address space gives more flexibility, because it enables pages in the global virtual address space to migrate from one node to another, and still be accessed using the same address (`live page migration`). For instance if one node breaks, remote nodes may still access the data stored in the corresponding address space, if data successfully migrated to another node and the page table(s) have been updated to translate the address to the new node. A case of interest to ExaNeSt is when page migration is only allowed within the boundaries of a node. In this case, the global virtual address needs to identify a node and a virtual address within that (host) node. A global page table is not needed here. The memory management unit within the host node will map all incoming virtual addresses to local physical addresses.

With UNIMEM, in order to enable a global address space, each island has to allocate disjoint regions of its physical addresses to local memory, local peripherals, and to the external world — *i.e.* memories and peripherals of other islands. The memory address "window" to the external world enables direct memory and I/O accesses to remote islands, through simple load/store instructions. The interface to the remote world can be any communication protocol (AXI, Ethernet, etc.). A translation mechanism in hardware will provide a dynamic or static mapping between the island's physical address space and the global address space. If the window is large enough to support the entire global address space, then the coherence island can directly access any memory in the system without the need of any complicated translation mechanism in hardware. In the ExaNeSt prototype, the Zynq Ultrascale+ MPSoC will support a 448 GB external window allowing to have more than 16 coherence islands (with 16 GB of DRAM memory each) without the need of a dynamic mapping.

4.1. High-Throughput intra- and inter-Mezzanine communication

APElink [21] is the communication protocol for the management of data flow over the HSS links. It is based on a word-stuffing protocol, meaning that data transmission needs submission of a *magic* word every time a control frame is dispatched to distinguish it from data frames.

The word-stuffing APElink protocol includes two words — Magic and Start — into the data flow over the HSS links to establish the logical link between nodes. The transmission of the packet header is announced with this sequence. Since misrouted packets are disruptive for the network, the highly critical header integrity could be protected by an Error Detection Code (EDC) or Error Correction Code (ECC), depending on the Bit Error Rate (BER) that we experience in the network. To prevent the reception buffer from overflowing, the IP manages the flow between two neighbouring nodes by keeping track of the APElink words sent. Buffer availability is measured by *credit*; occupancy of the receiving buffer is contained in the *credit*. Outbound words consume it, causing transmission suspension as soon as a programmable credit threshold is reached – *i.e.* *credit* is exhausted — and resuming as soon as info about newly available space bounces back to the transmitter — *i.e.* *credit* is eventually restored. Furthermore, this information is mandatory for the Virtual Cut-Through (VCT) switching mechanism described in section 5.

Information regarding the health of the node can be embedded in the *credits*, allowing a fault communication mechanism to avoid a single point of failure and guaranteeing a fast broadcast of critical status [22]. The diagnostic messages embedding in the communication protocol limits the amount of additional overhead and avoids that this flow affects overall performance.

5. Routing

The APEnet IP [23] is the data flow handler over the HSS network (Tier 0/1/2), implementing low-latency and high-speed communications between the Mezzanines. The *Router* is the component in charge of determining the path the messages will follow to reach their destinations.

Current APEnet implementation adopts a deterministic Dimension-Order Routing (DOR or e-cube) policy: it consists in reducing to zero the offset between current and destination node coordinate along one dimension before considering the offset in the next dimension. The APEnet DOR router is able to handle more than one packet transaction at a time and specialized priority registers — writable at run-time — allow for limited but effective routing function customization.

The employed switching technique — *i.e.* when and how messages are transferred along the paths established by the routing algorithm — is Virtual Cut-Through (VCT): the router starts forwarding the packet as soon as the algorithm has picked a direction and the buffer used to store the packet has enough space.

A more sophisticated routing logic will of course be able to consume the coordinates in a more exotic way or recognize critical directions and then change appropriately the packet header to follow an alternative path to their the destination. Partly and fully adaptive (star-channel [24]) routing algorithms are under evaluation and several preliminary results are shown in section 6.

Finally, we will implement a set of effective collective communication functions, typically acting as a bottleneck for the HPC systems. To enhance system application performance at very large scale, a new design with hardware offloading of these functions is under development.

6. Network simulator results

Benchmarking and characterization of an interconnection network depend on many parameters — *e.g.* traffic pattern, buffers and network sizes — therefore at very large scale simulations are mandatory to evaluate solutions and understand criticalities. The simulator is implemented in a modular way to ease switching between different network designs (topologies, routing algorithms and traffic generators) and to allow for an effective way of measuring latency and accepted traffic.

Table 1. Topology and routing algorithm analyzed.

Network topology	**Routing Algorithm**
2D torus 10×10	e-cube, star-channel, smart dim-order
2D torus 32×32	e-cube, star-channel, smart dim-order
3D torus $10 \times 10 \times 10$	e-cube, star-channel, smart dim-order
Fully connected dragonfly "72 nodes"	min-routing
Fully connected dragonfly "1056 nodes"	min-routing

For APEnet in ExaNeSt evaluation, we used the OMNeT++[25] framework to implement the base functionality of the APEnet network in a proprietary simulation library.

As a first evaluation, we use synthetic benchmarks on different network configurations. All the nodes were producing traffic using a Bernoulli process and with a uniformly random destination. The tests were performed for 2D/3D torus and dragonfly topologies and several routing algorithms, as listed in Table 1. In the test, the accepted traffic is normalized dividing it by the number of nodes in the network.

Preliminary results for the network accepted traffic (figure 10) show a linear region shared by all the different network configurations tested. When the network is in the linear region, it is below its critical congestion threshold and properly handles the incoming traffic; enhancing the applied load results in higher accepted traffic. If the applied load is above the saturation point, the accepted traffic starts to exit from the linear region of the plot and reaches a plateau. The plateau value could not correspond to the maximum value that the network is able to deliver due to congestion effects. The 2D tori handle $\sim$45% normalized applied load in a 10×10 network

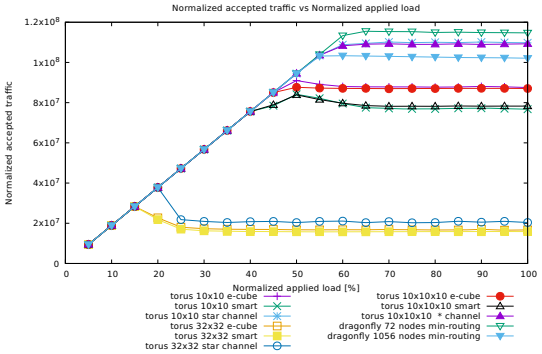

Figure 10. Normalized accepted throughput vs applied load.

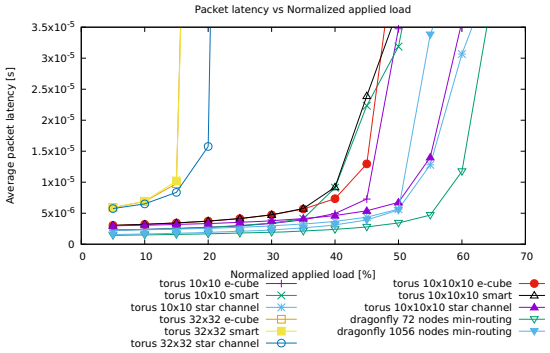

Figure 11. Latency vs applied load for the different configuration tested.

configuration, but only ∼20% in a 32 × 32 configuration. Tori are not optimized for uniform network traffic and the performance degrades quickly increasing the radius of the configuration. To reduce the radius, we can move from a 2D to a 3D torus. The fully adaptive star-channel routing algorithm provides a better use of the available network resources, resulting in higher sustained load and lower latency (figure 11) than those achievable by using the simpler e-cube (DOR) routing. The 72 nodes dragonfly performs better than 10 × 10 torus adopting the fully adaptive algorithm but on a smaller network, while the 1056 nodes setup shows better performances than the non-adaptive tori but with lower throughput than the fully adaptive ones.

7. ExaNeSt project status

The ExaNeSt project started at 1Q16 and foresees two consecutive, 18-months long periods. The first phase, which will end in July 2017, focuses on the design of general system architecture and the realisation of the main hardware building blocks. During the second phase, the project consortium will integrate, test and evaluate through application benchmarks the delivered system prototype. The current status of ExaNeSt hardware blocks is:

- The QFDB has been designed and it is under production; first release is expected for 2Q17.
- The Mezzanine board, hosting up to 4 QFDBs and/or thermal mock-up for power dissipation analysis will be released few weeks after the QFDB.
- The design of QFDB's FPGA firmware has been started on a Xilinx FPGA development kit. We procured a number of Trenz(R) Zynq UltraScale+ systems (TE0808-03ES2-S) to implement an emulator of final ExaNeSt hardware.
- Small-size clusters based on Trenz FPGA, interconnected via 10 gbps custom links, have been installed: (i) to test and evaluate performance of the ARM V8 multi-core programmable systems and HSS transceivers, (ii) to deliver a first release of the ExaNeSt network, based on INFN APEnet router and Forth Unimem-based network interface.

8. Conclusions

In this paper we reported the general vision and few preliminary details of the ExaNeSt system network. ExaNeSt project aims at designing a densely packed, immersion-liquid cooled HPC system integrating a huge number of low power 64-bit ARM processors embedded in the last generation 16 nm Xilinx high-end FPGA (Zynq UltraScale+). The advanced design of the hardware system will leverage on reconfigurable components to integrate and test multiple network topologies (from high radix Dragonfly to n-Dim Torus). A benchmarking set of HPC

and big data applications able to scale to ExaFLOPs will be used to select the most performance and power effective interconnection architecture that will be integrated in the ExaNeSt system.

On the basis of the project budget and expected cost of the components at 4Q18, the final prototype will be made of an assembly of a dual liquid cooled chassis hosting $\sim$ 50 QFDBs (*i.e.* 200 FPGAs) distributed on 9 Mezzanines per chassis and interconnected via the brand new custom ExaNeSt network architecture based on multiple 10 gbps channels.

9. Acknowledgment

This work was carried out within the ExaNeSt project, funded by the European Union Horizon 2020 research and innovation programme under grant agreement No 671553.

References

[1] Katevenis M *et al.* 2016 The ExaNeSt Project: Interconnects, Storage, and Packaging for Exascale Systems *2016 Euromicro Conference on Digital System Design (DSD)* pp 60–67
[2] Aroca R V and Gonalves L M G 2012 *Journal of Parallel and Distributed Computing* **72** 1770 – 1780 ISSN 0743-7315 URL http://www.sciencedirect.com/science/article/pii/S0743731512002122
[3] Luijten R P and Doering A 2013 The dome embedded 64 bit microserver demonstrator *Proceedings of 2013 International Conference on IC Design Technology (ICICDT)* pp 203–206 ISSN 2381-3555
[4] Rajovic N *et al.* 2013 Supercomputing with commodity cpus: Are mobile socs ready for hpc? *2013 SC - International Conference for High Performance Computing, Networking, Storage and Analysis (SC)* pp 1–12 ISSN 2167-4329
[5] ExaNoDe http://exanode.eu/ accessed: 2017-02-02
[6] Mavroidis I *et al.* 2016 Ecoscale: Reconfigurable computing and runtime system for future exascale systems *2016 Design, Automation Test in Europe Conference Exhibition (DATE)* pp 696–701
[7] Capuzzo-Dolcetta R, Spera M and Punzo D 2013 *Journal of Computational Physics* **236** 580 – 593 ISSN 0021-9991 URL //www.sciencedirect.com/science/article/pii/S0021999112006900
[8] Springel V 2005 *Monthly Notices of the Royal Astronomical Society* **364** 1105 URL http://dx.doi.org/10.1111/j.1365-2966.2005.09655.x
[9] Monaco P, Theuns T and Taffoni G 2002 *Monthly Notices of the Royal Astronomical Society* **331** 587 URL http://dx.doi.org/10.1046/j.1365-8711.2002.05162.x
[10] Theuns T *et al.* 2015 Swift: Task-based hydrodynamics and gravity for cosmological simulations *Proceedings of the 3rd International Conference on Exascale Applications and Software* EASC '15 pp 98–102
[11] Paolucci P S *et al.* 2015 *Journal of Systems Architecture* ISSN 1383-7621 URL http://www.sciencedirect.com/science/article/pii/S1383762115001423
[12] Januszewski M and Kostur M 2014 *Computer Physics Communications* **185** 2350 – 2368 ISSN 0010-4655 URL http://www.sciencedirect.com/science/article/pii/S0010465514001520
[13] OpenFOAN http://openfoam.org/ accessed: 2017-02-02
[14] RegCM http://www.ictp.it/research/esp/models/regcm4.aspx accessed: 2017-02-02
[15] Plimpton S 1995 *Journal of Computational Physics* **117** 1 – 19 ISSN 0021-9991 URL http://www.sciencedirect.com/science/article/pii/S002199918571039X
[16] MonetDB https://www.monetdb.org/ accessed: 2017-02-02
[17] KALEAO URL https://www.kaleao.com/Products/kmax
[18] Kim J, Dally W J, Scott S and Abts D 2008 Technology-driven, highly-scalable dragonfly topology *2008 International Symposium on Computer Architecture* pp 77–88 ISSN 1063-6897
[19] Marazakis M *et al.* 2016 Euroserver: Share-anything scale-out micro-server design *2016 Design, Automation Test in Europe Conference Exhibition (DATE)* pp 678–683
[20] Durand Y *et al.* 2014 Euroserver: Energy efficient node for european micro-servers *2014 17th Euromicro Conference on Digital System Design* pp 206–213
[21] Ammendola R *et al.* 2013 *Journal of Instrumentation* **8** C12022 URL http://stacks.iop.org/1748-0221/8/i=12/a=C12022
[22] Ammendola R *et al.* 2015 *Future Generation Computer Systems* **53** 90 – 99 ISSN 0167-739X URL http://www.sciencedirect.com/science/article/pii/S0167739X14002751
[23] Ammendola R, Biagioni A, Frezza O, Lo Cicero F, Lonardo A, Paolucci P S, Rossetti D, Salamon A, Salina G, Simula F, Tosoratto L and Vicini P 2011 *Journal of Physics: Conference Series* **331** 052029
[24] Gravano L, Pifarre G D, Berman P E and Sanz J L C 1994 *IEEE Transactions on Parallel and Distributed Systems* **5** 1233–1251 ISSN 1045-9219
[25] OMNeT++ project URL https://omnetpp.org/