Towards Sustainable HPC*

Maike Gilliot[1], François Bodin[2], Guillaume Colin de Verdiere[3], Marc Duranton[4], Mark Asch[5], Laurent Morin[6], and Gerd Büttner[7]

[1]ETP4HPC Office, Bruyères-le-Châtel, France
[2,6]University of Rennes, France
[3]CEA - DAM/DIF, Arpajon, France
[4]CEA - LIST, Gif-sur-Yvette, France
[5]University of Picardie, Amiens, France
[7]Airbus, Hamburg, Germany

March 30, 2021

Contents

1 Introduction

HPC has not been designed as an environmental-friendly technology. Its objective is rather to deliver high performance and high precision numerical tools for

*This work received funding under the EXDCI-2 project from the European Union's Horizon 2020 research and innovation programme under grant agreement No 800957.

science, highly optimised for delivering calculation results as fast as possible. Today, as any other societal domain, the HPC community has to accept its share of work in reducing e-waste and energy consumption.

For most ICT technologies, it is the production process where the most energy is spent [Sid11]. According to [LFR14], 81% of the energy consumption of ICT equipment occurs during the manufacturing process, while 19% is dedicated to the system's operation. In the case of supercomputers, these ratios are more balanced between the manufacturing process and the operation phase with respect to the Global Warming Potential ("GWP" for short, a standardised environmental impact measurement). We estimate the GWP for a supercomputer on the detailed analysis of the life cycle assessment of a high-end servers over the time: on average, it can be asserted that the manufacturing and the use part have roughly the same impact on the environment, assuming a lifetime of approximately 5 years ([Thi19], [Lau15], [Gup+20]).

On the one hand, the key factors during the manufacturing stage are the exploitation of rare minerals (e.g., gold), and the manufacturing of electronic components (ICs). According to these studies, the usage of non-volatile storage (SSD) has massively changed the global amount of IC die surface, and increased the weight of the manufacturing process in the overall GWP. On the other hand, the critical factor during the exploitation phase is the energy mix used producing the energy for running the system. Within Europe, the life-cycle GWP of the energy mix can vary by a factor up to 20 (Estonia vs. Sweden), and factor 10 between two major countries (Germany vs. France) [Ass18]. Also the geographic location of the system plays a critical role, as the ratio between the production and the exploitation phase in the global GWP can vary between 20% and 80%, depending on the configuration of the system and the exploitation location, but averaged at the European level, the ratio between manufacturing phase and production phase balances out. This preliminary overview also shows that a proper evaluation of the environmental impact of a supercomputer needs to takes into account all types of nodes: compute nodes, accelerators, and storage nodes as well. Note that the storage can represent up to 30% of the total number of nodes in a supercomputer [IDR20].

Much research has been done in the past years in reducing the energy consumption during the use phase: on the hardware side, (for example, by reducing the energy consumption of CPUs), as well as on the algorithmic side. Regarding the environmental impact of the manufacturing process, one can aim at:

1. improving and re-designing the manufacturing process itself ("frugal design"), and
2. increasing the lifetime of HPC systems, so as to amortise the environmental cost of the manufacturing process.

The first option offers probably (like many industrial processes) a high potential for improvement with respect to the overall environmental cost. As we are not experts in this domain, the remaining of the paper focuses on the second option: in this paper we explore the idea to extend the lifetime of HPC systems in order to improve the ecological balance. The driving question we tackle along this paper is: **what would be the consequences of increasing a system's lifetime to 20 years?**

Section 2 sketches the factors that may motivate such an approach, as well as potential inhibiting factors, looking at the users and the application developers. In Section 3, we also present some first ideas on how this could be achieved. This idea to increase the lifetime is not as far fetched as it could seem at first glance: in domains such as aeronautics, some systems already have a lifetimes of 20 years or more. In Section 4 we will discuss what can be learned from their experience, before closing this paper with some concluding remarks (Section 5).

2 Supporting factors and obstacles

Such an increase of the average lifetime of HPC systems would affect users, vendors, the HPC system developers, and application developers. Without being exhaustive, this section sketches some of the technical factors that could support such a change, as well as some inhibiting elements. By purpose, economic constraints are out of scope (as we are not experts in this domain). Such a shift would not emerge instantaneously, it would need to be advocated for and pursued actively by the different stakeholder within the HPC ecosystem.

2.1 Towards the Continuum

It has been a long time since HPC systems were only an insulated infrastructure. The convergence of computing and data has been a first step of the integration of computing centres in a set of infrastructures that gathers sensors, the edge, the fog, data centres denoted the continuum [Asc+18]. A continuum, which can be viewed as logically linked infrastructure elements, sits on top of the cyberinfrastructure.

The continuum concept emerges to account for the fact that many new applications integrate sources of data, intermediate infrastructures (storage facilities, fog [NIS18]), networks, and multiple numerical models running on supercomputers. The fundamental issue of the continuum is to provide the ability to deploy **complex end-to-end application workflows** [ORA20a] under data movement constraints (such as speed of light, latency, available bandwidth, storage volume, security, etc.) as well as resource allocation, as we cannot expect that all components of a continuum will be under the same administrative domain (multi-owner, multi-tenant infrastructures). Some examples of such continuum are deployed to implement digital twins [Wik20], sensor networks (for instance array of things [Urb20]), coupling scientific instruments and numerical computation (for instance the Phidias project [Cin20]).

It should be noted that maintaining an application relying on a complex workflow that is deployed on a large set of devices ranging from server/compute nodes and edge/sensors elements is very challenging. The hardware maintenance can be very expensive especially when the devices at the edge are installed in places (e.g. in the sea) difficult to access. On the software side, making updates to adapt functionalities and to fix bugs can also be very complex and risky. Container technologies [Lin20; Doc20] and microservices are particularly interesting to address these issues, but will not be sufficient to address the arising issues.

There is a convergence of objectives in deploying a complex application workflow in a continuum of resources and environmental considerations. Both objectives requires perennial approaches that reduces maintenance cost in software and hardware.

2.2 Danger of fossilisation

Keeping systems in operations for 15 to 20 years would strongly influence the software development and the innovation process.

Application and system developers would have to adapt and to tune their developments for this particular architecture, leading to highly optimised applications. The users would benefit from better performances of their applications, without new hardware, but "just" by better exploiting the underlying architecture. Today's application rarely fully exploit the compute power at their disposal. Optimising for a given architecture thus reserves a lot of potential gain, as can be observed for example in the aeronautic industry. Airbus, for example, relies for its entire 320-fleet for their on-board flight system on a single-core processor , which was brought onto the market in the 90th. Having to comply with heavy certification procedures, processors for the on-board systems are changed very rarely. The 320-family will probably be built for another 20 years, with a life expectancy of 20-30 years for airplanes. Thus, this processor will thus remain operational till 2060, approximately. In order to achieve better performances on application level, processor optimised code has been developed by Airbus.

Indeed, HPC in general does not have to comply to such strict certification rules. However, we can assume that to some extent long-lasting HPC systems may face similar traps and pitfalls as the airplane on board systems. In particular, sticking over a long period of time to one architecture influences also the development of new applications, as they will be tailored from the beginning with this particular architecture in mind - even if the architecture might be already outdated.

Moreover, incremental innovation would be hindered. The regular update of systems we see today allows for incremental changes and innovations, while preserving the compatibility with previous systems. Long-lasting systems on the other hand would perhaps become fossilised with time and less open to innovation. The cost of changing the architecture would be even higher as it is already today.

2.3 Physical Laws, Technologies and Human Resources

Over time, the reliability of these systems will be threatened by a decrease in individual transistor reliability due to manufacturing defects prevalent at deeply scaled technology nodes, device ageing related effects, etc. The chips built using these devices will be increasingly susceptible to errors due to the reduced noise margins arising from near-threshold voltage (NTV) operation (that will be necessary to meet the limits on power consumption). These effects are expected to increase the rate of transient and hard errors in the system. The scientific applications running on these systems will no longer be able to assume correct behavior of the underlying machine. The errors will propagate and generate various kinds of failures, which may result in outcomes in HPC applications ranging from data corruptions to catastrophic crashes [HE17]. Longer system lifetimes need to address this issue and offer solutions to handle at some higher level the faulty system components.

Today, from the hardware point of view, contrary to the past, there is little gain in upgrading CPUs for sequential codes [Eti18]. Increase in the number of cores has provided more performance for parallel codes but the largest gain potential being currently provided by accelerator technologies (especially with the increasing use of deep learning techniques [WWB19]). Because of the various parallel and distributed natures of the tasks composing application workflows, application developers are facing many challenges that require highly specialised skills (code optimisation, numerical models, data analytics, machine learning, etc.). For instance, one of the challenges is dealing with complex workflows to implement data related tasks such as assimilation, pre-processing, post-processing, code coupling, etc [ABN16].

A second challenge is to consider the hardware heterogeneity of the cyberinfrastructure the application is deployed on. To address the complexity of the task many programmers are considering approached where libraries and code modules are assembled using scripting-like languages (e.g. Python) [Göb+18; Mor20]. This approach allows to implement a separation of concerns between application related issues and code optimisation issues. It is also an opportunity to provide new algorithms, libraries, runtimes, workflows and specialised code generation tools [Lat+20; Git21].

In the future, for the development of HPC-applications, developers will have to rely even more on the support of methods and tools allowing for the separation and the isolation of the different parts of HPC applications, which require very diverse know-how and programming expertise.

2.4 Meeting the users' needs: end users and application developers

For the users, the frequent update (and renewal) of the systems provide increasingly powerful systems, capable of delivering the storage and computing capabilities that keep up with their needs [ETP20; PRA18].

Designing a new cyberinfrastructure must meet a set of new constraints and reach a trade-off that balances the necessary manpower on the one hand and a reduced consumption of resources and energy on the other hand. The fundamental nature of this trade-off is related to the ability to design efficient algorithms and implementations to support heterogeneous hardware architectures. This is intrinsically a task that requires not only scientific work (numerical methods as well as computer science tools) but also engineering support (the interested reader can refer to the ORAP Forum "Reconciling sustainability and performance, what challenges for Exascale?" [ORA20b]). This is a trend that is currently incompatible with the research organisation and funding in Europe.

The usual consideration in the acquisition of a cyberinfrastructure is the Total Cost of Ownership (TCO), with performance and energy consumption as part of the of the main criteria for the purchase decision. Today, performance is usually estimated using a set of benchmarks. These benchmarks, while relevant, correspond to a vision of the past pushed by users/scientists that wish to shorten the time to porting the code to the new cyberinfrastructure and to experimental results. Typically this is not favourable to any kind of innovative system, as this would require more effort in porting codes/algorithms.

3 Towards a New HPC Design Approach

If we consider extending the lifetime of cyberinfrastructure while avoiding fossilising we then face a very complex challenge that could be addressed via these three priorities :

1. Building modular hardware and software in order to improve parts reuse;
2. Modernising/improving applications in order to gain performance for the users (without updating the hardware);
3. Improving resilience to hardware and systems failures at application level to circumvent component failure;

In the remainder of this section we detail these priorities.

3.1 Building modular hardware

To achieve parsimony in resource spending, completely new hardware design must be considered in order to make future HPC systems more modular, thus reducing e-waste and allow for partial upgrades. Accelerator-based architectures [Eur19] can be a way to consider gaining performance while keeping the central parts of the systems. It should also be noted that this path of reasoning is in line with the end of Moore's Law, which obliges system designers to look for performance and energy efficiency by using specialised hardware. This approach has been known for a long time in embedded system design.

At the core of this approach are progresses in technology such as silicon photonics connections [Hip19] that thanks to their short latency, may help to build more modular machines (for instance by separating the memory from the compute part).

At the edge of the continuum the key factor is reconfigurability capabilities in order to adapt over time to new application needs. This is particularly important for devices (e.g., undersea sensors) that are difficult to reach and therefor expensive to maintain.

All these considerations require to rethink an economical sector whose rationale is based on "programmed obsolescence". In particular, this questions use of the latest technological node (7nm or less) that increases the faults caused by electro-migration (e.g. this is why a processor or a memory DIMM can only have a limited lifespan).

3.2 Modernising and improving applications

Keeping the hardware stable over many years means that the applications cannot gain in performace due to newer hardware; Thus, software optimisation becomes extremely important. This means to look for new, more efficient algorithms that can exploit accelerators. Many previous experiences have shown that [Sch+19]) well optimised codes have provided performance benefits much higher than the hardware's natural performance evolution.

It is important to note that many new applications are based on complex workflows deployed on a continuum. Efficiency must then be considered holistically, taking into account data movements between infrastructures - and not only the costs for the core numerical simulation. This includes data movements at all levels, from the Edge to the Fog and to the supercomputer, as well as from the execution unit to the (on-chip) cache, from cache to memory, and from memory to disk.

Ease of deployment must also be considered in order to reduce the cost of development. Virtualisation technologies, such as containers, are among the key technologies to allow a good fit between long lasting re-configurable hardware and application running on a continuum which is upgraded by parts[1].

When hardware evolutions do not provide a gain in performance, this gain will to be achieved by better algorithms and better implementation. These tasks rely on human intelligence and expertise, which will hopefully be simplified in the future by machine learning techniques.

3.3 Improving resilience to hardware and systems failures

Resilience to faulty hardware or other system failures is by no means a new domain. But an increase of the number of components (within the HPC system itself, and all other parts of the workflow), along with a longer lifetime might require to review and to improve resilience and fall-back mechanism. Anyway, exascale systems will require new approaches, due to their sheer size. In exascale systems, check-pointing may take (depending on the applications) up to several hours, which is not acceptable in the long run [2]. The progress in resilience will benefit exascale systems and allow for a longer lifetime.

4 Lessons to be learned from Aeronautics

First, the issue of sufficient supply (and support) of hardware components needs to be addressed. When the one-core processor of their on-board information system ran out of production, Airbus acquired several hundreds of thousands of these processors to secure their availability for the years to come. Whereas this may be feasible for a company such as Airbus, this solution would not work at large for other sectors and other clients.

Second, along with the hardware, the software environment must be maintained in its inital set-up, meaning that for the development and the testing of the application the software environment, including drivers and annex libraries, must be kept unchanged. This requires effort to maintain those "old" system configurations over many years - along with the know-how and the expert staff.

And third, this also requires Airbus to maintain expert skills for programming these old processors for many years. It can be hard to find younger experts with these competences. In HPC, we face a similar situation today with Fortran. With Fortran not being part of the scientific courses at university level, it gets difficult to hire Fortran experts today. Generalising this idea of long-lasting systems would also need to cope with the preservation of such competences.

[1] As long as the application is not hardware dependent.

[2] Example: RAMSES [RAM21] simulations on 10,000-20,000 cores currently use about 1-2 hour for the checkpoint/restart I/O, to be compared to the 24-hour job wall time on most supercomputers.This represents a 4-8% overhead.

5 Conclusion

In this paper we are advocating that new ways of designing systems are needed to account for environmental challenges. One can argue that HPC is a niche market whose impact is small enough not to require changes. However, its entanglement with the data production and use is promoting a generalisation of HPC technologies (for instance at the Edge) in a continuum of resources.

Developing more parsimonious long lasting systems requires to go back to computer science basics to produce more efficient algorithms, modular architectures, infrastructure and networks. E-waste needs to be explicitly integrated in the equation. Furthermore, the race for more data has also to end and we need to put a strong focus on data management (data production, data movements, data storage) at all scales in order to reduce energy consumption.

This also questions the way we organise research and industry HPC activities. Currently, most of the research teams lack basic support to develop and to optimise codes despite experiences showing great benefit [Sch+19; Tho20] in modernising application codes.

Europe could lead this transformation towards a more eco-friendly HPC approach: while in Europe we do consume a lot of the over all available HPC computing power, we are less active in the production of HPC systems. Europe could alleviate its dependency by extending the system lifetime and by investing more in services and software development than in hardware acquisition. Such a shift will not emerge instantaneously, it would need to be pushed and pursued actively by the different stakeholders within the HPC ecosystem. In particular, unprecedented R&D efforts should be initiated in the following directions:

1. Improving the chip manufacturing process in order to make it more environment friendly. As systems become more energy efficient (cf. [Ato21]) and energy becomes greener, the main emphasis is on the manufacturing process in order to archive progress.

2. Proposing new efficient modular system designs that allow easily for partial upgrades in order to avoid fossilisation of the infrastructures and also to promote support services.

3. Supporting improved use and integration of accelerators in HPC systems.

4. Improving the efficiency of HPC, HPDA and IA applications using more efficient algorithms and highly optimised codes.

One of the important side effects of such a shift toward sustainability is to ease the use of EU technologies and promote the excellence in applications and code development.

6 Acknowledgements

We would like to thank Thierry Bidot, Stephane Requena, Marcin Ostasz and Jean-Philippe Nominé for their valuable input and their support. With their questions and their comments they challenged our ideas in a highly productive manner.

References

[Sid11] Siddharth Prakash, Ran Liu, Karsten Schischke and Dr. Lutz Stobbe. *Timely replacement of a notebook under consideration of environmental aspects.* https://www.oeko.de/oekodoc/1584/2012-440-en.pdf. [Online; accessed 16-april-2020]. 2011.

[LFR14] Thomas Taro Lennerfors, Per Fors, and Jolanda van Rooijen. "Sustainable ICT: A Critique from the Perspective of World Systems Theory". In: *ICT and Society.* Ed. by Kai Kimppa et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 57–68. ISBN: 978-3-662-44208-1.

[Lau15] Laura Talens Peiró, Fulvio Ardente, EU Commission. *Environmental Footprint and Material Efficiency Support for product policy.* https://publications.jrc.ec.europa.eu/repository/bitstream/JRC95187/lb-na-27200-en-n.pdf. [Online; accessed 15/02/2021]. 2015.

[ABN16] Mark Asch, Marc Bocquet, and Maëlle Nodet. *Data assimilation: methods, algorithms, and applications.* Fundamentals of Algorithms. SIAM, 2016, pp. xviii + 306. URL: https://hal.inria.fr/hal-01402885.

[HE17] Hukerikar and Engelmann. "Resilience Design Patterns: A Structured Approach to Resilience at Extreme Scale". In: *Supercomput. Front. Innov.: Int. J.* 4.3 (Sept. 2017), pp. 4–42. ISSN: 2409-6008. DOI: 10.14529/jsfi170301. URL: https://doi.org/10.14529/jsfi170301.

[Asc+18] M Asch et al. "Big data and extreme-scale computing: Pathways to Convergence-Toward a shaping strategy for a future software and data ecosystem for scientific inquiry". In: *The International Journal of High Performance Computing Applications* 32.4 (2018), pp. 435–479. DOI: 10.1177/1094342018778123. eprint: https://doi.org/10.1177/1094342018778123. URL: https://doi.org/10.1177/1094342018778123.

[Ass18] Association of Issuing Bodies. *European Residual Mixes.* https://www.aib-net.org/sites/default/files/assets/facts/residual-mix/2018/AIB_2018_Residual_Mix_Results_v1_1.pdf. [Online; accessed 15/02/2021]. 2018.

[Eti18] Daniel Etiemble. "45-year CPU evolution: one law and two equations". In: *CoRR* abs/1803.00254 (2018). arXiv: 1803.00254. URL: http://arxiv.org/abs/1803.00254.

[Göb+18] Jens Henrik Göbbert et al. "Enabling Interactive Supercomputing at JSC Lessons Learned". In: *High Performance Computing.* Ed. by Rio Yokota et al. Cham: Springer International Publishing, 2018, pp. 669–677. ISBN: 978-3-030-02465-9.

[NIS18] NIST. *Fog Computing Conceptual Model.* https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-325.pdf. [Online; accessed 16-april-2020]. 2018.

[PRA18] PRACE. *Scientific Case.* `https://prace-ri.eu/third-scientific-case/`. [Online; accessed 16-april-2020]. 2018.

[Eur19] Eurolab4HPC. *Eurolab-4-HPC Long-Term Vision on High-Performance Computing (2nd Edition).* `https://www.eurolab4hpc.eu/media/public/vision/Eurolab-Vision_2.pdf`. [Online; accessed 30-april-2020]. 2019.

[Hip19] Hipeac. *HiPEAC Vision 2019.* `https://www.hipeac.net/vision/2019/`. [Online; accessed 30-april-2020]. 2019.

[Sch+19] T. C. Schulthess et al. "Reflecting on the Goal and Baseline for Exascale Computing: A Roadmap Based on Weather and Climate Simulations". In: *Computing in Science Engineering* 21.1 (Jan. 2019), pp. 30–41. ISSN: 1558-366X. DOI: `10.1109/MCSE.2018.2888788`.

[Thi19] Thinkstep AG. *Dell Technologies, Thinkstep AG. Life Cycle Assessment of Dell R740.* `https://corporate.delltechnologies.com/content/dam/digitalassets/active/en/unauth/data-sheets/products/servers/Full_LCA_Dell_R740.pdf`. [Online; accessed 15/02/2021]. 2019.

[WWB19] Yu Wang, Gu-Yeon Wei, and David Brooks. "Benchmarking TPU, GPU, and CPU Platforms for Deep Learning". In: *CoRR* abs/1907.10701 (2019). arXiv: `1907.10701`. URL: `http://arxiv.org/abs/1907.10701`.

[Cin20] Cines. *Phidias Project.* `ttps://www.phidias-hpc.eu/`. [Online; accessed 16-april-2020]. 2020.

[Doc20] Docker. *Docker Containers.* `https://www.docker.com/`. [Online; accessed 14-avril-2020]. 2020.

[ETP20] ETP4HPC. *Strategic research Agenda (SRA).* `https://www.etp4hpc.eu/sra.html`. [Online; accessed 16-april-2020]. 2020.

[Gup+20] Udit Gupta et al. "Chasing Carbon: The Elusive Environmental Footprint of Computing". In: *ArXiv* abs/2011.02839 (2020).

[IDR20] IDRIS. *Jean Zay : calculateur HPE SGI 8600,Description matérielle détaillée.* `http://www.idris.fr/jean-zay/cpu/jean-zay-cpu-hw.html`. [Online; accessed 15/02/2021]. 2020.

[Lat+20] Chris Lattner et al. *MLIR: A Compiler Infrastructure for the End of Moore's Law.* 2020. arXiv: `2002.11054 [cs.PL]`.

[Lin20] Linux. *Linux Containers.* `linuxcontainers.org`. [Online; accessed 14-avril-2020]. 2020.

[Mor20] Morris Riedel. *Selected European Perspectives and Trends as Top 5 Recommendations for Smart Cyberinfrastructures for AI in the Future.* `http://smartci.sci.utah.edu/`. [Online; accessed 17-april-2020]. 2020.

[ORA20a] ORAP. *Forum 43, Workflows for Scientific Computing.* `http://orap.irisa.fr/?page_id=1073`. [Online; accessed 16-april-2020]. 2020.

[ORA20b] ORAP. *Reconciling sustainability and performance, what challenges for Exascale?* `http://orap.irisa.fr/?page_id=1155`. [Online; accessed 20-april-2020]. 2020.

[Tho20] Thomas Schulthess. *Bridging the software and performance gap to exascale for weather and climate simulations.* `http://orap.irisa.fr/wp-content/uploads/2020/01/Thomas-Schulthess-Orap-44.pdf`. [Online; accessed 20-april-2020]. 2020.

[Urb20] Urban Center for Computation and Data. *Array of Things.* `http://arrayofthings.github.io/`. [Online; accessed 16-april-2020]. 2020.

[Wik20] Wikipedia. *Digital Twin.* `https://en.wikipedia.org/wiki/Digital_twin`. [Online; accessed 16-april-2020]. 2020.

[Ato21] Atos. *Press release on hydogene for data centers.* `https://atos.net/en/2021/communiques-de-presse_2021_02_25/atos-hdf-energy-datacenter-hydrogene-vert`. [Online; accessed 16/03/2021]. 2021.

[Git21] Github. *Kokkos repository.* `https://github.com/kokkos/kokkos`. [Online; accessed 25/01/2021]. 2021.

[RAM21] RAMSES Website. *Ramses repository.* `http://irfu.cea.fr/Phocea/Vie_des_labos/Ast/ast_sstechnique.php?id_ast=904`. [Online; accessed 25/01/2021]. 2021.