D4.1

Reviewed co-design methodology, and detailed list of actions for the co-design cycle

Carlo Cavazzoni, Luigi Genovese, Dirk Pleiter, Filippo Spiga

Due date of deliverable: 31/05/2019 (**month 6**)
Actual submission date: 31/05/2019
Final version: 31/05/2019

Lead beneficiary: CINECA (participant number 8)
Dissemination level: PU - Public

Document information

Project acronym:	MAX
Project full title:	Materials Design at the Exascale
Research Action Project type:	European Centre of Excellence in materials modelling, simulations and design
EC Grant agreement no.:	824143
Project starting / end date:	01/12/2018 (month 1) / 30/11/2021 (month 36)
Website:	www.max-centre.eu
Deliverable No.:	D4.1

Authors: Dr. Carlo Cavazzoni (CINECA)

Dr. Luigi Genovese (CEA)

Dr. Dirk Pleiter (Forschungszentrum Jülich)

Mr. Filippo Spiga (Arm Ltd)

To be cited as: Cavazzoni et al., (2019): Reviewed co-design methodology, and detailed list of actions for the co-design cycle. Deliverable D4.1 of the H2020 project MAX (final version as of 31/05/2019). EC grant agreement no: 824143, CINECA, Casalecchio di Reno (BO), Italy.

D4.1 Reviewed co-design methodology, and detailed list of actions for the co-design cycle

Table of Content

Executive Summary

The main topic of this document is to internally review and refresh the co-design methodology described in the project proposal and define a first set of actions that will lead to internal activities in the consortium targeting all MAX codes.

The *MAX co-design cycle* consists of a multi-level co-design cycle where interactions and review checkpoints between various actors (e.g. domain scientists, HPC performance engineers, HW/SW architects) take place at different levels of granularity, from full application down to self-contained micro-kernels. It is expected that certain tasks within the MAX co-design cycle are hardware specific, others hardware agnostic and other focus on programming models and application of software engineering practices.

For the duration of MAX, we anticipate a co-design cycle will be instantiated at different stages of the general cycle due to different objectives and key outcomes. Every action will be tailored and customized based on the needs of a specific code or community. The high level objective is to maximize exploitation of heterogeneous hardware with the minimum possible amount of hardware-specific modifications on the MAX community codes.

This deliverable is structured in 4 major sections:

- Section "*The worldwide Race toward Exascale*" highlights major announcements and updates from other Exascale initiatives outside Europe;
- Section "*Review of co-design methodology*" broadly revisit our co-design methodology providing few additional details about the interplay between hardware and software components;
- Section "*Work Package Tasks Updates*" reports different activities that have been carried out in the different tasks of this Work Package in the first 6 months of the project;
- Section "*Conclusions and next steps*" closes this deliverable and set the stage for the beginning of coding and evaluation activities for the next core two years of the project.

The worldwide Race toward Exascale

In this section we report about the status of other Exascale project in the world targeting, among others, material science as one of the key application areas.

In Europe, EuroHPC Joint Undertaking [1] is the entity responsible for acquiring and providing a world-class petascale and pre-Exascale supercomputing and data infrastructure for Europe paving the way toward Exascale solutions based on European technologies. In-depth updates and progresses on the European Exascale map have been recently presented at the EuroHPC Summit Week in Poznań, Poland [2].

On the path from pre-exascale and peta-scale EuroHPC systems towards exascale systems, the fraction of technology developed in Europe is supposed to be increased. In particular, the EC has the expectation that european processor technology is being used. These come with commodity Arm-based cores supporting Scalable Vector Extension (SVE, [3]) instructions as well as custom accelerator cores based on RISC-V [4].

EuroHPC plans to have at least two pre-Exascale systems to be installed before the end of 2020, and to fund a Research and Innovation program, that should be the base for co-designing the processor and accelerator through the European Processor Initiative (EPI, [5]) and the architecture through the implementation of two to four Exascale pilot systems.

Subject to funding availability, two Exascale systems will be procured by EuroHPC in Europe by 2023. At least one will be based mostly on European technology, and in particular on the European processor and accelerator.

Latest updates from United States

The US Department of Energy (DoE) is the primary responsible to develop an Exascale roadmap in USA. Alongside strategic procurements coordinated by several National laboratories, the DoE is funding a large umbrella collaborative project called Exascale Computing Project (ECP) [6] to develop a capable exascale ecosystem. ECP touches hardware integration, software technologies and application developments. Co-design of US Exascale systems is driven by the ECP Proxy App Suite [7]. Within this group of mini-apps, surprisingly there is only *miniQMC* (extracted from the popular QMCPACK [8]) which is related to material science.

US are planning to deploy two pre-exascale systems by 2020 and 3 exascale systems by 2021-2022. The two pre-exascale systems will be:

- `Perlmutter` (also known as "NERSC-9", [9]) at the National Energy Research Scientific Computing Center (NERSC) high performance computing user facility operated by Lawrence Berkeley National Laboratory. It will feature a AMD Epyc CPUs and next-generation Nvidia Tesla GPUs. It targets ~100 PetaFlop/s.
- *`Crossroads`*, which has not yet been announced, is going to be managed by Los Alamos National Laboratory and Sandia National laboratory. It likely going to target ~200 PetaFlop/s.

As part of Perlmutter procurement, NVIDIA and NERSC will partner to co-develop and enhance PGI OpenMP capabilities to support OpenMP 5 standard and NVIDIA GPU in offloading mode. This demonstrates the willingness of pushing open standards for the long-term benefits rather than proprietary solutions (e.g. OpenACC).

In respect of Exascale, the three systems all capable to exceed 1 ExaFlop/s in double precision are `Frontier` at Oak Ridge national laboratory, `Aurora` at Argonne National Laboratory and `El Capitan` at Lawrence Livermore National Laboratory (this being the only system not yet officially announced).

`Aurora` [10, 11] is scheduled by 2021 and it will be the very first Exascale system deployed for open science able to greater diversity of workloads, including machine learning and data intensive tasks, in addition to traditional simulations. It will be based on future Intel Xeon Scalable processor, Intel Optane DC Persistent memory and Intel's Xe compute architecture (most probably an accelerator-type of device). Intel is developing an open-source parallel programming framework called `oneAPI` aiming to simplify and unify the developer experience and target diverse computing engines such as CPU, GPU, FPGA and also custom accelerator ASICs. Further details of `oneAPI` will be released publicly during 2019.

`Frontier` (also known as "OLCF-5", [12]) is scheduled for 2021 and it will be accessible for open science like Aurora via the DoE INCITE programme targeting computationally intense simulation and data challenges with the potential to significantly advance key areas in science and engineering. Like the Aurora system, it will be a CRAY system based on the new Shasta architecture and Slingshot interconnect. The choice of computing component is very different, Frontier will use high-performance AMD EPYC CPU and AMD Radeon Instinct GPU technology with a 4:1 GPU-to-CPU ratio. The peak performance target is set to exceed 1.5 ExaFlop/s. Historically AMD has lack of a great software ecosystem in efficiently programming their own high-end GPU products. With this new system planned, AMD is committed to improve bottom-up the entire software stack focus again on open standards like OpenMP and portable compiler technology.

While today's Summit system (IBM Power 9 and NVIDIA "Volta" GPU) is still operational, the Oak Ridge Leadership Computing Facility (OLCF) is beginning to accept project proposals for its Center for Accelerated Application Readiness (CAAR) in order to prepare the current generation codes for the future system. Awarded projects will obtain early access to hardware prototypes.

Latest updates from Japan

For many years the K Computer [13], Japan's long-lasting top flagship national supercomputer, has been recognised worldwide for its innovation and achievements. In 2014 the Japan Ministry of Education, Culture, Sports, Science and Technology (MEXT) initiated the Flagship 2020 "Post-K" project aiming to develop the K Computer's successor, an exascale-ready system capable of at least 10x performance improvement on real applications compared to the current K Computer. Japan is fostering a healthy ecosystem of Tier-2 centres each targeting deployment of 10~100 PetaFlop/s between 2019 and 2027. Only one Exascale system will be developed and it will be hosted by the RIKEN Center for Computational Science (R-CCS) in Kobe.

Japan's future "Post-K" supercomputer [14], recently named *Fugaku* in honor of Mt Fuji, aims to replicate the success of the K Computer by designing a balanced system design for efficient data movement. Thanks to its unique characteristics, K Computer have been awarded for many consecutive years at the top of the HPCG Top500 [15] and Graph500 [16] ranking. The Fugaku supercomputer promises to replicate the same success.

The co-design of Japan's only Exascale system was driven by the FIBER mini-app suite [17]. The suite was developed between 2014 and 2015 by the RIKEN Advanced Institute for Computational Science (RIKEN AICS) as co-design vehicle for the Post-K programme. Within this group of 8 mini-apps there are 2 with a material science focus: `CCS QCD` and `NTChem-MINI` (ab-initio quantum chemistry mini-app for the molecular electronic structure calculations).

In the past year, multiple presentations and papers have highlighted key characteristics of the system such as the brand-new Arm-based A64FX CPU [18, 19] and 6D Tofu-D interconnect [20]. Multiple recent public presentations [21] also highlights the timeline and speculated sizes of the full system deployment.

In short, the core component (A64FX CPU) of this supercomputer is an ARMv8-A based chip with 512 bit SVE vector instructions. It includes in a single water-cooled SoC 4 cluster of 12 cores each (48 in total for compute plus 4 cores exclusively dedicated to handle the operating system and I/O, hence not accessible via user space), 32 GByte of High Bandwidth Memory (HBM) and integrated Tofu-D interconnect. Each chip support various AI data-types (FP16, INT8, etc.) and promises to deliver ~3 TFlop/s within a competitive power envelope.

The exact size and final system configuration of the Fugaku supercomputer is not made public yet. The decommission of the K Computer and refurbishment of RIKEN R-CCS datacentre in Kobe will begin in August 2019. The Fugaku supercomputer will consume the exact physical space of the current K Computer. Full system bring-up and access to early users is scheduled by 2020.

Latest updates from China

Within all the Exascale worldwide, the ones pursued in China are unfortunately the less transparent and less detailed to the public. A dominant motivation behind China race toward Exascale has been the country's commitment to rely more on homegrown hardware and software technologies rather than procure those from outside of its own borders, particularly from the United States.

There are three major strategy toward exascale in China that can be summarised as follows:

- The "*Sunway Pla*n" based on SW26010 CPU currently claiming 3.06 TFlop/s peak (3.13 PetaFlop/s each compute module). It will be located in National Laboratory for Marine Science and Technology in Tsingtao;
- The "*Tianhe Plan*" based on Matrix 2000+ currently claiming 3.14 PetaFlop/s peak each compute module. It will be located in TianJin;
- The "*Sugon Plan*" based on Hygon x86 processors and custom build DCU accelerators currently claiming 3.19 PetaFlop/s peak each compute module. It will be located in Zhengzhou.

Further details have been summarised in a recent presentation at the EuroHPC Summit week [22]. These lead institutions are already deploying operational pre-exascale prototypes in the

order of tens of PetaFlop/s. Various speculative sources claim at least one Exascale chinese system appear by 2020 but no confirmed news is public yet. It is confirmed a deployment of a ~100 PetaFlop/s system based on Hygon CPU and DCU accelerator in Zhengzhou by 2020 and its expansion to reach exascale is scheduled to be completed by 2022.

Unfortunately due to barrier language and almost secretive approach to their internal development, we have no reliable information about which application domains and which mini-apps have been used for co-design.

Review of co-design methodology

In Figure 1 we report the breakdown of the MAX co-design cycle, as it was described in the project proposal of the second phase of MAX.

The flowchart summarizes our methodology which is based on two main double nested cycles: 1) an *outer loop* that targets large codebase re-engineering driven by a refactoring strategy based on performance evidence gathered from Proof-of-Concepts; 2) an *inner loop* that evaluates micro-benchmarks and build performance models using simulation and emulation tools which give the opportunity to explore micro-architecture and code transformations at a manageable scale.

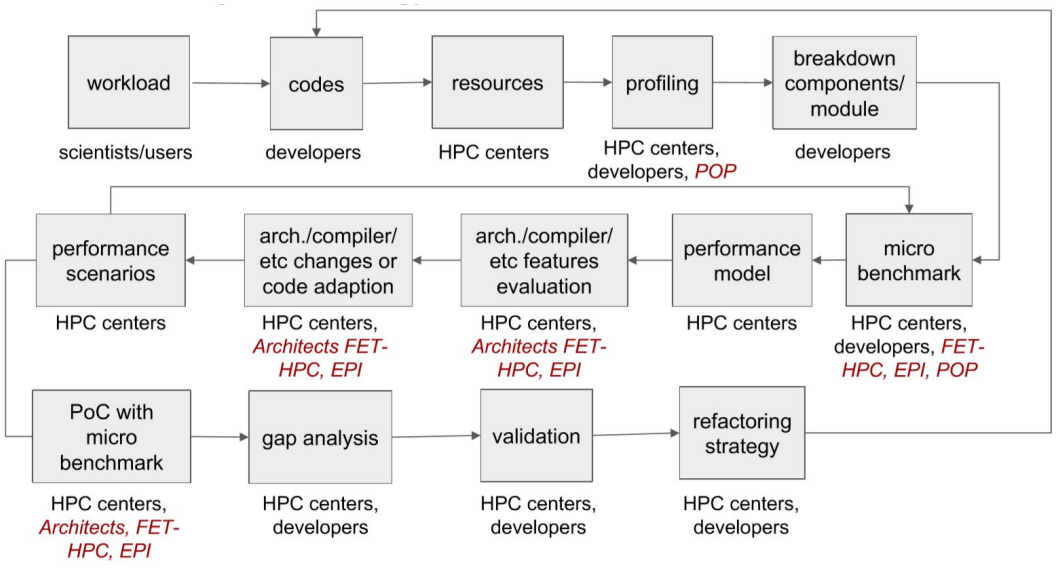

Figure 1 -- This diagram highlights the MAX co-design cycle as it was described in the MAX proposal. Under each block are indicated the owners of a specific action (in red the non-MAX ones) accountable to contribute to the overall vision.

We had the opportunity in this first 6 months to wrap-up the brainstorming around the methodology and we begun discussing practical options with different stakeholders inside MAX project representing chip designer (Arm), system integrators (E4) and supercomputing centers (CINECA, Juelich, CEA, BSC, CSCS).

Finally, we try as much as possible to compare and improve our methodology by tap into the ones adopted in other flagship EU projects (e.g. EPI) and other successful FETHPC EU-funded projects.

MAX aims to develop a sustainable transition toward the Exascale era for all its material science flagship codes. We are also considering use-cases arising from the user communities that are relevant for the specific objectives of Work Package 4. So far, we did not explicitly reach out to the broader community asking for new inputs, we leave this action to the second half of the project.

Depending on the maturity of the new technologies and paradigms under investigation, the granularity of our activities can vary considerably. In many practical scenarios, we foresee the need to restrict the investigation to one or few small kernels, even micro-kernels (less than 200 lines of code), in order to be effective and achieve meaningful results in a reasonable time.

Mini-apps have been used in many other Exascale initiatives as vehicle to drive co-design activities, alongside classic benchmarks and micro-kernels (e.g. BLAS, LAPACK, FFT). In MAX first phase two very successful mini-apps derived from Quantum ESPRESSO codebase [23] have been developed: `LAXLib` and `FFTXLib`.

These mini-apps are self-contained, they operated both in serial and parallel, can be linked to optimized numerical libraries provided by vendors (cuBLAS/cuSOLVE, Intel MKL, IBM ESSL, Arm Performance Libraries) or open source alternatives (e.g. netlib BLAS/LAPACK, OpenBLAS, ATLAS). They also provide a sort of unit-testing to control correctness and reproducibility of results.

Now we are planning to introduce at least two new kernels/mini-apps form BigDFT and CP2K codes. The mini-app derived from BigDFT will be focus on convolution operations using wavelets base functions. Since convolutions are also the core computational kernels of many neural network frameworks, this mini-app maybe be used to test ideas and programming models resulting in a broader interest beyond the material science community. The mini-app derived from CP2K will focus on `libDBCSR` (or `DBCSR` for short) [24], a sparse matrix library designed to efficiently perform sparse matrix matrix multiplication across CPU (MPI and OpenMP) and GPU (still experimental support).

Overall, we aim to continue to maintain these MAX mini-apps and make them available fully independently by original code. These will be published online on a public repository and basic documentation provided. Contributions from external bodies will be controlled by MAX code owners and subject to strict code review to avoid introducing binding dependencies. We will work to make these mini-app as much as possible compliant with the proxy app standards adopted by the ECP project in USA.

No matter how small a mini-app or a micro-kernel can be, exploring new technologies and new paradigms is both exciting and risky. We will prioritize our efforts on technologies and

parallel computing paradigms (languages and frameworks) not only based on their future potential but also based on their solid roadmap and their broader impact in many other HPC communities. Every evaluation is going to consider tradeoff between performance and another factor like programmability, flexibility in production deployment, manutenability, cost. The project timeline dictates us to be able to deliver tangible results and concrete actions that have an impact on the main MAX codes today (short term practical objective) and also create a deep understanding on which are the right hardware and software key characteristics to build the best future HPC system fit for the European material science community (long term aspirational goal).

The core principles of the MAX co-design methodology have been presented at the EuroHPC co-design workshop in Poznań during the EuroHPC Summit Week (Figure 2).

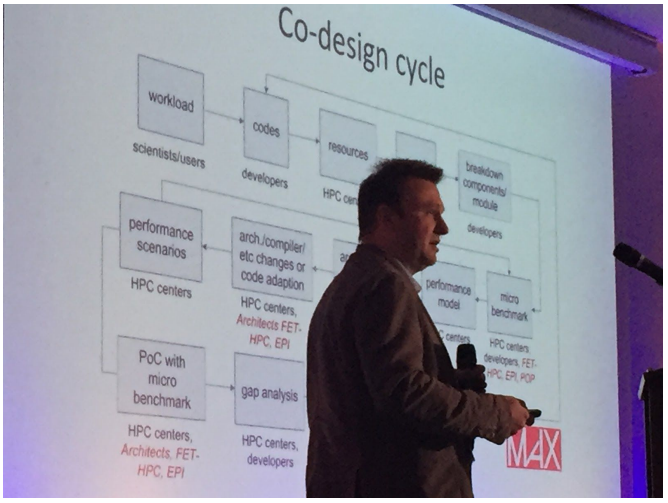

Figure 2 -- Carlo Cavazzoni (work-package leader) showcasing MaX co-design cycle at the EuroHPC co-design workshop in Poznań (2019).

Going into more practical terms, we were able to classify opportunities in 3 major categories:

- *Hardware-centric*, both agnostic or specific to a specific hardware architecture;
- *Software-centric*, both software-to-hardware and software-to-software;
- *System-centric*, from a node to a full system level.

Hardware-centric co-design opportunities

Hardware-centric co-design opportunities relevant for MAX codes and communities can be classified as hardware-agnostic and hardware-specific. Ideally, hardware-agnostic opportunities allow a broader applicability and avoid lock-in to a specific preferred technology provider. In practice, every hardware evaluation do require a specific hardware which is inevitably provided by a specific provider. We cannot ignore that, in many cases, vertical hardware integration turns out to be extremely effective.

Looking at the inner co-design cycle, hardware-agnostic and hardware-specific considerations can blend if the right method is in place and the right questions are considered. An example

can be the evaluation of different floating point vector units. Identify the ideal vector length in respect of memory subsystem constraints will be a valid outcome applicable to any hardware that includes a similar feature. At the same time, each hardware (ARM, Intel, RISC-V or AMD) will have specific compiler back-end that generates instructions for the different vector units. The resulting degree of vectorization depends by ISA expressiveness and compiler maturity, the energy efficiency and peak performance depend by manufacturing engineering process and the micro-architecture.

Hardware co-design opportunities are prioritised by taking into account their impact on MAX scientific use-cases toward the exploitation of future pre-Exascale and Exascale infrastructure capabilities in Europe. MAX, as part of a large and tightly connected ecosystem of HPC projects, we will consider a priority those actions that are related to, or may have a positive impact for the European Processor Initiative project's outcomes.

Software-centric co-design opportunities

Software-centric co-design opportunities can be grouped into 3 categories [25]:

- *Hardware-to-Software*: e.g. validating and influencing the definition and evolution of APIs to support the new low-level hardware features;
- *Software-to-Software*: e.g. exploration of new parallel programming languages (e.g. OpenMP vs OpenACC) to achieve performance portability;
- *Software-to-Applications*: e.g. integrating the new domain specific libraries (e.g. SIRIUS) or specialised numerical libraries into existing MAX codes;

Under the software-to-software co-design opportunities, we will explore over time the evolution of those core programming languages (e.g. new Fortran and C++ standards) and parallel programming paradigms (e.g. OpenMP, OpenACC, MPI+X) used in MAX. We are not expecting our communities to rewrite entirely MAX applications following a complete different paradigm or language, the amount of software development and engineering effort required goes beyond what is practically achievable in the lifespan of an European project. We witness true large-scale code transformations in very few occasions, mostly mission-critical simulation codes used by large national laboratories with a budget of tens of millions of dollars.

In this new phase of MAX we will rely on less intrusive directive-based approaches to achieve realistic performance targets on those applications or those applications features currently unable to exploit many-core or accelerated systems. OpenMP 5 represents a good candidate to tackle portability across future heterogeneous accelerated Exascale systems thanks to full support for accelerator devices, descriptive loop constructs, understanding of multilevel memory systems and enhanced portability. While OpenMP 5 standards has been finalised, approved and published [26], tools like compilers and run-times may take years to fully implement such features. Actually we observed that sometimes features are never implemented because "too hard" or not useful in practice by real code. MAX will engage with

compiler developers, both commercial and open-source, to influence wherever possible the implementation features relevant to MAX codes.

System-centric co-design opportunities

System-centric co-design opportunities are maybe less obvious and less attractive for domain scientists but do play a critical role in enabling new challenging simulations and high-throughput workflows on modern large-scale HPC systems. Several MAX codes are currently used as benchmark tools for procurements, both for bidding and acceptance: for example the *PRACE Unified European Application Benchmark Suite* (UEABS) [27] contains two codes (Quantum ESPRESSO and CP2K) belonging to MAX portfolio.

Looking at how an HPC system is built, there are many design choices that heavily affect the performance of a code (and there is not a mix that is perfect for all the possible codes deployed on a system): latency and bandwidth of high-speed interconnect for both point-to-point and collective operations, the topology of the interconnect and its effects on traffic congestion and Quality of Service, ratio of sockets versus accelerators, ratio of cores versus memory; type of memory technology, type of memory persistency (volatile or non volatile) and many other factors.

Due to lack of time and considering the main focus being on improving first the MAX codes, we will focus on exploring combination of compute, memory and communication *via* target benchmarking campaigns on as many different systems as accessible to our communities via PRACE [28] and national HPC centers. We will leverage results from H2020-funded FETHPC projects or collaborate with ongoing FETHPC projects. For instance, SAGE [29, 30] and SAGE-2 [31] provide the opportunity to explore future high-performance object stores for MAX workflows. The MAESTRO [32] project has one of the MAX software components, namely SIRIUS [33] library, in its co-design application portfolio.

MAX focus as Centre of Excellence is not only about large-scale simulations. Materials modeling has evolved in the past 5 years dramatically and many exciting contemporary discoveries have been possible by orchestrating thousands of high-throughput calculations and then mining the obtained results. Running workflows rather than single monolithic jobs creates a new series of challenges on current HPC systems which will multiply by 10 or 100 times when pre-Exascale and Exascale systems will become operational.

AiiDA [34, 35] is the framework selected by MAX to manage, preserve, and disseminate the simulations, data, and workflows related to material science. Within this Work Package we aim to advise AiiDA developers of the challenges of the future systems and, as feedback, we will formalise what requirements future HPC middleware stack (e.g. OS customisations for jitter reduction, effective containers technologies, robust policies for job schedulers) need to satisfy to enable complex workflows to be executed smoothly and with high performance. In this context we plan to work also with Fenix and the ICEI project [36] on deploying such platform services at scale and integrating AiiDA in a federated e-infrastructure.

Work Package Tasks updates

In this section we briefly report on key decisions, general directions and activities carried on within the different tasks after the initial consultations among Work Package members in this first 6 months of activities. Each Task progresses in parallel and almost independently compared to others for the entire duration of project. Moreover each Task contributes in various degree and form into all 6 planned Work Package deliverables.

Advanced programming models [Task 4.1]

This task focuses on the analysis, evaluation and adoption of forefront and Exascale-oriented programming models, with particular focus on heterogeneous architectures and many-core exploitation.

In several occasions we discussed internally about the programming paradigms most promising for future Exascale architectures. We have not quite grasped yet which is the "best horse to bet on" but we have clearly identified the importance of following standards and the need to introduce novelty in our current legacy coding practices by following small iterations rather than via radical and massive changes.

MPI and OpenMP, *de facto* standards in HPC, are set to stay relevant and broadly adopted for at least another decade. We foresee improvements in legacy codes via two major trends: 1) the gradual substitution of synchronous parallel code with asynchronous one; 2) the gradual adoption of task-based parallelism replacing the old-fashioned legacy fork-join model which introduces more and more synchronization penalties in high core-count manycore systems.

In particular we will continue to investigate task level programming using OpenMP applied to various MAX mini-apps. One activity started already in MAX first phase, and set to be completed in MAX, is related to FFTXlib. The objective is to move from a Proof-of-Concept done in CUDA Fortran [37] (vendor specific and completely custom for NVIDIA GPU) to a production-ready implementation using OpenMP portable directives targeting many-core CPUs or (potentially) any bus-attached accelerator. We already faced the challenge that the OpenMP implementations available today do not feature the syntax constructs that will enable us to achieve our objective in a clean way. We successfully managed to exploit a workaround thanks to the expertise provided by the POP CoE [38].

Where true high-performance without compromises can only be achieved by computational kernels written for specific architectures, the encapsulation of all the specialisation into a self-contained library helps enormously. However, to maintain and generate custom kernels for many architectures is a tedious task.

One option MAX is planning to explore is the use of meta-programming paradigm to generate specific kernels tuned for specific architectures, at compile-time or at run-time. The BigDFT code makes use of a self-contained library who has been developed following this meta-programming principle. `libconv` [39] is a fast and automatized convolution library

born from the needs of the BigDFT code which employs wavelets as an internal computational basis. Some operations involving wavelets may be written in terms of convolutions with short, separable filters. The `libconv` library uses the BOAST [40] meta-programming engine which is able to perform source-to-source optimization and abstracts the convolution generations. Such an approach is of great interest in the context of the auto-tuning process, as the metrics which drive the convolution kernel generations might be put in correlation with various performance figures, like the ratio FLOPS/BYTES or number of registers used.

We will work to identify other opportunities like `libconv` in other MAX codes. The next case eligible for benefitting from this approach could be mesh points computation of real-space grid magnitudes in SIESTA.

Exploitation of emerging (multi-tier) memory hierarchies [Task 4.2]

The scope of this task is to review how MAX codes are using the current memory hierarchy of HPC systems, and study how they can exploit the new type of memories featuring pre-Exascale and Exascale HPC systems. This is, in particular, relevant in the context of the EPI processor, which plans to integrate two different types of memory, namely High Bandwidth Memory (HBM) for a fast memory tier and DDR memory for a capacity optimised tier.

It is well known that one of the main issues in future Exascale architecture will be the widening of the gap between floating point performance and memory bandwidth to bring the data fast enough to wider execution units. To mitigate the issues new memory tier will be introduced in Exascale architecture increasing the memory hierarchy.

Then, following our methodology, concerning the co-design actions about the exploitation of multi-tier memory subsystem, we started from the analysis of the scientific use-cases, chosen among the demonstrator of Work Package 6. With the help of MAX scientists, we are going to set priorities regarding these use-cases and codes, to start the investigation about the role of multi-tier memories and derive best practices to make the best use of them

In particular we will investigate and prioritize at least two use-cases that can benefit from the exploitation of non-volatile memory (NVM) and HBM. The NVM tier can be used to reduce the latency and bandwidth of access to intermediate (within a workflow) data objects that today are saved in the filesystem. HBM tier instead can be used to increase the bandwidth for smaller data object used in the innermost loops of the numerical kernel. In both cases co-design actions can be done to evaluate the effectiveness of different set of configuration parameters, like the size of the tiers and the access mode (cache, direct, mixed, etc…), for the selected scientific use case. This is carried out within three activities:

- Profiling the MAX codes with respect to the usage of the memory hierarchy in available Tier-0 HPC systems (e.g. Marconi at CINECA and/or Piz Daint at CSCS)

using tools that allow for identification of frequently and less frequently accessed memory pages, e.g. the Working Set Size (WSS) Tools for Linux [41];
- By means of two case-studies involving MAX codes, investigating how different bandwidth and capacity of HBM chips could affect code performances at Exascale and how the codes should eventually be refactored to better exploit HBM;
- By means of two case-studies involving MAX codes, investigating how different Non Volatile Memory (NVM) chips usage models (e.g. as pageable memory, block device, etc) and different bandwidth and capacity could affect code performance and code functionalities (e.g. check-point restart, scratch file, in core vs out of core, etc) at Exascale and how the codes should eventually be refactored to better exploit NVM.

Co-design [Task 4.3]

This task is broadly responsible for the implementation of the co-design cycle including the development of performance models and code enablement for new forthcoming architectures (e.g. Arm SVE).

In the first six months we organized thanks to Arm member of MAX two webinars covering the topics of architecture simulation of complex System-on-Chip (SoC) using `gem5` [42] and Arm SVE instruction set emulation using `armie` [43]. These tools are open-source of freely accessible, heavily customizable for an expert audience. Both of them are also used in other European projects (EPI and Mont-Blanc 2020 [44]). The webinars have been recorded and will be made available publicly together with the material. We aim to re-run these webinars again opening attendance to members of other CoEs as well.

The next immediate step is to coordinate the selection of few mini-app or micro-kernels to be used for architecture design exploration. These exercises will provide to us an exceptional amount of insights about low-level hardware constraints. We will coordinate our work with similar on-going activities in EPI, focusing on domain-specific aspects of MAX codes. There are two main design points worth exploring in this Task that are closely aligned with EuroHPC goals:

- Using armie, studying the amount of SVE vector instruction generated by various compilers and executed;
- Using gem5, measuring the arithmetic intensity (FLOPS/BYTE ratio) and how it varies based on different memory technology and cache sizing.

We will not focus on the effects of various Network-on-Chip (NoC) topologies on the ability of any workload to efficiently scale to tens of threads in the same SoC.

Our ultimate intent is to perform the analysis and provide feedback and scenarios for different architecture parameters within reason. We rely on Arm staff expertise to do so. However we are not going to drown any definitive conclusion but instead keep our findings at the level of academic research open to public dissemination.

Profiling and monitoring performance [Task 4.4]

This task is in charge of taking care of the benchmarking and profiling activity for the MAX flagship codes. This is, in principle, a quite simple activity, that actually requires a strong coordination among all the participants in order to deliver a high quality product, of which both the developers and the end users could take benefit. In the recent past, benchmarking and profiling were naively conducted by each code owner in an independent manner. As a consequence, the results of these benchmarks were inhomogeneous and scattered in several places. One of the targets in this task is to perform benchmarks and profiling in a more homogeneous way and to collect them with a common standard format in a single place open to the users.

To reach this objective, we started a "Benchmarking Working Group" open to the contributions of the participants of other Work Packages and code owners. In the first of these meetings, we worked to have a common approach and to distinguish what are the differences between what we call "benchmarking" and "profiling". We agreed that the former is an activity that should be performed on the official release of the codes with the aim to assess the performance reached by the flagship codes on different architectures. These data figures should be made comparable and available to the end users. To this aim, we will use the GitLab repository of MAX [45]. The first actions in this direction are the decisions of a "benchmarking calendar" in order to schedule the activities, and the collection of the input datasets for the execution of the tests.

On the other side, we agreed that the "profiling" activity is something that should mostly target the developers, highlighting the weaknesses in the performances of the codes and pointing out where to work for their optimization. Differently from the benchmarking, the profiling activity will be performed during some meetings scheduled during the year. During these meetings, code developers and HPC technologists will work side by side to analyze the modules/components critical for the application performances.

During this first stage, it emerged one important criticality related to the availability of computational resources. It is very important to clarify how these could be provided in the context of the european infrastructure. It should be also noted that the current amount of resources that PRACE reserves for the CoEs in the regular calls, i.e. 0.5% of the total number of funded node-hours, is insufficient even only for the purposes discussed in this task.

Conclusion and next steps

We are at the beginning of this co-design journey with both new challenges ahead and a strong foundation from previous MAX experience.

To strategically align MAX activities to other key European Research & Innovation projects like the European Processor Initiative, we need to go beyond the typical code profiling

exercise based only on time-to-solution. It is time to selectively investigate hardware architecture at node level by using simulation and emulation techniques.

Profiling and benchmarking are still very relevant activities in the MAX co-design cycle and serve the two objectives: (1) helping identify fine-grain computational kernels which are relevant enough to kick-start in-depth architectural design exploration (e.g. see "*Hardware specific co-design opportunities*"); (2) assessing the coarse-grain behaviour of many modules (some better optimized than other) working together running at medium and large scale.

This last point allow us to focus on the role of programming models (e.g. see "*Software specific co-design opportunities*") and to quantify the effects on performance of complex system-level design choices (e.g. see "*System specific co-design opportunities*").

Due to the iterative and feedback-driven nature of MAX co-design cycle, we avoid on purpose to define a monolithic set of action for each code or for each mini-app selected. The cases we provided while describing core principles are examples. The exact activities will be refined in parallel to the work performed by other Work Packages focused on software engineering and code modularity (WP1, WP2, WP3).

For the duration of MAX, we will rely on European HPC National Centres to access a variety of different architectures. While exploring diverse hardware platform is positive, we will look at opportunities to access large portions of next pre-Exascale supercomputing systems in collaboration with EuroHPC governing boards.

MAX partners are in contact with all major HPC technology providers (Intel, NVIDIA, AMD, Marvell) and technology integrators (ATOS, HPE, CRAY, Lenovo). These entities usually provide technology briefing under NDA with direct customers (e.g. national HPC centres). We are planning to leverage these information by drawing general future trends and directions in both Hardware and Software without breaking confidentiality.

It is also worth establish meaningful interactions and close collaborations with other European-funded projects focus on hardware technologies (EPI, DEEP-EST [46]), programming models (EPiGRAM-HS [47], EPEEC [48]) and other Centres of Excellence (POP-COE, BioExcel [49]). Each engagement will be handled *ad hoc* based on shared converging interests. We see the ownership of these relationships being aligned with the objectives and mission of MAX Work Package 4. Other Centres of Excellence are interested in adopting similar co-design cycle as the one adopted by MAX, we will exploit the European network of Centre of Excellence coordinated by Focus-CoE [50] to disseminate our findings and expand training about how effectively drive co-design explorations using simulation and emulation tools.

Last but not least, alongside monitoring progress made by similar Exascale projects in other parts of the world, Work Package 4 will engage in proactive Technology Watch by scouting important trends and changes in the HPC market that may have an effect on the evolution of the MAX codes leveraging synergies with efforts in PRACE-6IP [51] WP5. Technology

roadmaps in High Performance Computing are subject to significant changes with short notice, often not because the technology itself but driven by market and business decisions. Since effective co-design for Exascale is tightly coupled to the evolution of current commodity technology, it is important to monitor closely what is happening in the ecosystem to make informed decisions sooner rather than later. Work Package 4 members will build awareness around those changes and share in an appropriate manner to the whole MAX scientific communities.

In Summary, key activities of Work Package 4 in this initial 6 months have been:

- First series of constructive discussions between the group related of what technologies and programming models are still in scope (and why) in respect to Exascale;
- First series of internal webinars delivered by Arm showcasing simulation and emulation tools useful to unveil behaviour of workloads on future Arm-based hardware (e.g. future European General Purpose Processor);
- First landscaping exercise to identify which programming models are worth pursuing first and their current level of maturity to be adopted gradually into big codebase.

We plan to review and report co-design progress during all future deliverables, in particular

- in D4.2 "*First report on code profiling and bottleneck identification, structured plan of forward activities*" will help identify pertinent kernels and potential mini-apps (due Month 9);
- in D4.4 "*First report on co-design actions. Initial evaluation of the effectiveness of innovative programming models, different memory hierarchies and performance estimators for the co-design vehicle applications.*" (due Month 18) will also identify additional activities specifically focused on EPI general purpose processor and EPI accelerator.

We will continue for the entire duration of MAX project to monitor the evolution of hardware and software technologies relevant to MAX community.

References

[1] EuroHPC Joint Undertaking, https://eurohpc-ju.europa.eu/

[2] EuroHPC Summit Week 2019 (EHPCSW2019), Poznań, Poland, https://exdci.eu/events/eurohpc-summit-week-2019

[3] Nigel Stephens et al, "*The ARM Scalable Vector Extension.*" IEEE Micro 37, 2 (2017) -- Preprint: https://alastairreid.github.io/papers/sve-ieee-micro-2017.pdf

[4] RISC-V, https://riscv.org/

[5] European Processor Initiative, https://www.european-processor-initiative.eu/

[6] US DoE Exascale Computing Project (ECP), https://www.exascaleproject.org/

[7] Exascale Computing Project Proxy App Quality Standards and Best Practices, https://proxyapps.exascaleproject.org/standards/

[8] QMCPACK, https://qmcpack.org/

[9] NERSC's next Petascale supercomputer "*Perlmutter*", https://www.nersc.gov/systems/perlmutter/

[10] Argonne National laboratory's next Exascale supercomputer "*Aurora*", https://www.alcf.anl.gov/articles/introducing-aurora

[11] Aurora public technical specifications (by WikiChip), https://en.wikichip.org/wiki/supercomputers/aurora

[12] Oak Ridge National laboratory's next Exascale supercomputer "*Frontier*", https://www.olcf.ornl.gov/frontier/

[13] The K computer, https://www.r-ccs.riken.jp/en/k-computer/about/

[14] The Flagship2020 "*Post-K*" project, https://www.r-ccs.riken.jp/en/postk/project

[15] Top500 HPCG benchmark, https://www.top500.org/hpcg/

[16] Graph500 benchmark, https://graph500.org/

[17] FIBER mini-app suite, http://fiber-miniapp.github.io/

[18] HotChip'30 "*Fujitsu High Performance CPU for the Post-K Computer*" presentation, https://www.hotchips.org/hc30/2conf/2.13_Fujitsu_HC30.Fujitsu.Yoshida.rev1.2.pdf

[19] Official public A64FX specifications by Fujitsu, https://www.fujitsu.com/global/about/resources/news/press-releases/2018/0822-02.html

[20] Y. Ajima et al., "*The Tofu Interconnect D*", 2018 IEEE International Conference on Cluster Computing (CLUSTER), Belfast (2018) -- Preprint: https://www.fujitsu.com/hk/Images/08514929.pdf

[21] "*Japanese HPC Infrastructure Update*" presented at BDEC plenary at EHPCSW2019, https://events.prace-ri.eu/event/850/contributions/702/attachments/959/1544/4_15.05_Japan_Update_Kondo.pdf

[22] China Exascale plans update at EHPCSW2019, https://events.prace-ri.eu/event/850/contributions/702/attachments/959/1543/3_15.05_Chinese_Update_Jingheng_Xu.pdf

[23] `Quantum ESPRESSO`, https://gitlab.com/QEF/q-e

[24] `libDBCSR` library, https://www.cp2k.org/dbcsr

[25] Co-design principles adopted by DEEP projects, https://www.deep-projects.eu/co-design.html

[26] OpenMP 5 Technical Specification, https://www.openmp.org/wp-content/uploads/OpenMP-API-Specification-5.0.pdf

[27] The Unified European Application Benchmark Suite (UEABS), http://www.prace-ri.eu/ueabs/

[28] Partnership for Advanced Computing in Europe (PRACE), http://www.prace-ri.eu/

[29] H2020-EU.1.2.2 "*Percipient Storage for Exascale Data Centric Computing*" (SAGE) project, grant agreement No. 671500, http://www.sagestorage.eu/

[30] Narasimhamurthy Sai et al., "*The SAGE project: a storage centric approach for exascale computing: invited paper.*", Proceedings of the 15th ACM International Conference on Computing Frontiers (2018) -- Preprint: https://arxiv.org/pdf/1807.03632.pdf

[31] H2020-EU.1.2.2 "*Percipient Storage for Exascale Data Centric Computing 2*" (SAGE2) project, grant agreement No. 800999, http://www.sagestorage.eu/

[32] H2020-EU.1.2.2 "*Middleware for memory and data-awareness in workflows*" (MAESTRO) project, grant agreement No. 801101, https://www.maestro-data.eu/

[33] `SIRIUS`, https://github.com/electronic-structure/SIRIUS

[34] G. Pizzi et al., "*AiiDA: automated interactive infrastructure and database for computational science*", Comp. Mat. Sci. 111, 218-230 (2016)

[35] `AiiDA`, http://www.aiida.net/

[36] Interactive Computing E-Infrastructure for the Human Brain Project (ICEI) project, grant agreement No. 800858 funded by the EC under the Framework Partnership Agreement of the Human Brain Project (HBP), https://fenix-ri.eu/

[37] NVIDIA CUDA Fortran, https://developer.nvidia.com/cuda-fortran

[38] H2020-EU.1.4.1.3 "*Performance Optimisation and Productivity 2*" (POP-COE) project, grant agreement No. 801101, https://pop-coe.eu/

[39] `libconv`, a fast and automated Convolution library, https://github.com/luigigenovese/libconv

[40] `BOAST`, https://github.com/Nanosim-LIG/boast

[41] Working Set Size (WSS) Tools for Linux, https://github.com/brendangregg/wss

[42] `gem5` simulator, http://gem5.org/Main_Page

[43] Arm Instruction Emulator (`armie`), https://developer.arm.com/tools-and-software/server-and-hpc/arm-architecture-tools/arm-instruction-emulator

[44] H2020-EU.2.1.1 "*Mont-Blanc 2020, European scalable, modular and power efficient HPC processor*" (Mont-Blanc 2020) project, grant agreement No. 779877, https://www.montblanc-project.eu

[45] MaX benchmark repository, https://gitlab.com/max-centre/benchmarks

[46] H2020-EU.1.2.2 "*DEEP - Extreme Scale Technologies*" (DEEP-EST) project, grant agreement No. 754304, http://www.deep-project.eu/deep-project/EN/Home/home_node.html

[47] H2020-EU.1.2.2 "*Exascale Programming Models for Heterogeneous Systems*" (EPiGRAM-HS) project, grant agreement No. 801039, https://epigram-hs.eu/

[48] H2020-EU.1.2.2 "*European joint Effort toward a Highly Productive Programming Environment for Heterogeneous Exascale Computing*" (EPEEC) project, grant agreement No. 801051, https://epeec-project.eu/

[49] H2020-EU.1.4.1.3 "*Centre of Excellence for Biomolecular Research*" (BioExcel) project, grant agreement No. 675728, https://bioexcel.eu/

[50] H2020-EU.1.4.1.3 "*Concerted action for the European HPC CoEs*" (Focus-CoE) project, grant agreement No. 823964, https://www.focus-coe.eu/

[51] H2020-EU.1.4.1.3 "*PRACE 5th Implementation Phase Project*" (PRACE-6IP), grant agreement No. 730913, http://www.prace-ri.eu/prace-5ip/