MB2020 D3.1– Define a set of (Mini) Applications
Version 1.0

Document Information

Contract Number	779877
Project Website	www.montblanc-project.eu
Contractual Deadline	M3
Dissemination Level	Public
Nature	Document
Authors	Adrià Armejach (BSC), Marc Casas (BSC), Guillaume Colin de Verdière (CEA), Miquel Moretò (BSC), Dirk Pleiter (JUELICH), Fatima Rizvi (Bull)
Contributors	
Reviewers	Ludovic Saugé (Bull)
Keywords	BigData and HPC applications, Architecture simulation

Notices: *This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 779877.*

Change Log

Version	Description of Change
v0.1	Initial version of the deliverable
v0.2	Draft complete for internal review
v1.0	Finalized report

Contents

Executive Summary

This document reports the results of the activities planned in Mont-Blanc 2020 WP3 under task T3.1:

> **Task 3.1: Define a Set of (Mini) Applications:** In close cooperation with CEA and JUELICH, BSC will identify a set of real life applications that exhibit the characteristic demands of BigData-HPC applications. These applications will run on a simulator towards the end of the project, and as they may be too large to be processed in a reasonable time, representative mini applications (miniapps) will be chosen to replace the full applications. The optimal selection of these applications is essential in order that the project can deliver a competitive solution.

We report a list of (mini-)applications that exhibit the characteristics needed to stress all the segments targeted by MB2020.

1 Introduction

The project will follow a co-design approach to ensure the final hardware design meets real-world High-Performance Computing (HPC) application requirements. The applications selected for project testing need to address complex challenges in science and engineering, as well as stress the requirements the project is aiming to study, namely:

- **Requirements for Scalable Vector Extension (SVE):** A key aspect will be the analysis of the trade-offs for the different vector sizes (128 to 2048 bits) permitted in the SVE Instruction Set Architecture (ISA).
- **Memory hierarchy design requirements:** Analyse the memory performance and capacity requirements of the selected applications.
- **Requirements for a scalable Network-on-Chip (NoC):** Analyse the requirements for a scalable implementation that interconnects the different clusters of cores in an efficient manner with regards to coherence, speed, and extensibility.
- **Reliability Requirements for HPC:** Identify the reliability features required for future generations of MB2020 processors.

These requirements will be defined within Task 3.2, which starts in project month 4.

The set of applications has been selected based on the knowledge of today's use of supercomputers at JUELICH, CEA, Bull, and BSC; as well as on expectations on how future application mix might look like. The applications also cover a broad range of computational characteristics, from stencil-type applications to highly irregular applications.

In order to provide realistic data of applications currently running on a production machine, Table 1 lists the applications that run on the Curie supercomputer. Curie is a PRACE research infrastructure designed by Bull, owned by GENCI and operated by CEA at the *Très Grand Centre de Calcul* (TGCC). The table shows two lists that reflect the activities of the different communities using the machine. In the first list we have the codes developed by independent software vendors (ISVs), which are most often closed sources programs. The second list covers academic and community codes for which the source code is accessible.

All of these applications have large code bases ($\geq 100,000$ lines of code). Despite being large applications, which may be challenging or not feasible to simulate in its entirety, we have included some of them in our selected set because of their relevance - e.g., GROMACS (Section 2.7), NAMD (Section 2.17), or WRF (Section 2.26). We have also selected the ABINIT application in which CEA is involved (Section 2.1).

In addition, we have identified several applications that consume significant supercomputing resources in today's machines - e.g., LQCD (Section 2.13) and Quantum Espresso (Section 2.23); as well as applications that are expected to be used extensively in the near future - e.g., KKR-nano (Section 2.11) and Arbor (Section 2.2).

Our list also includes well-known mini-applications and synthetic kernels representative of HPC applications that are more amenable for simulations, e.g., MiniAMR (Section 2.15), LULESH (Section 2.14), and HPCG (Section 2.9).

The following section presents our extensive list of selected applications in alphabetic order. From this list we will chose a representative subset for each of the different application requirement studies in order to stress the requirement in question. Finally, Section 3 contains a summary table with the applications and their high-level characteristics.

Table 1: Applications running on the Curie supercomputer.

ISV applications	**Academic applications**
ANSYS Fluent/Autodyn	ABINIT
CD-adapco STAR-CCM+	CEA CAST3M
LSTC LS-DYNA	VASP University of Vienna
DASSAULT ABAQUS	GROMACS
MSC Nastran	GROMOS
COMSOL Multiphysics	CHARMM
ESI Pam-Crash	Cerfacs AVBP
Gaussian	CORIA Yales2
SCHRODINGER	EDF Code_Aster
Material Studio	EDF Code_Saturne
	BigDFT
	CP2K
	Quantum Espresso
	Siesta
	Gamess
	Lammps
	NAMD
	OpenFoam
	WRF

2 List of Selected Applications

2.1 ABINIT

ABINIT (`https://www.abinit.org`) is a software suite to calculate the optical, mechanical, vibrational, and other observable properties of materials. Starting from the quantum equations of density functional theory (DFT), you can build up to advanced applications with perturbation theories based on DFT, and many-body Green's functions (GW and DMFT).

This is a large application with about 895000 lines of code.

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP (written in FORTRAN)
- **Suitable for simulations:** Maybe - Might be necessary to focus on specific kernels.
- **Characteristics:** Scalable code, extensive use of FFT, vectorizes (SVE compatible), memory intensive.

2.2 Arbor

Arbor is a new framework for brain simulators, which is partially developed in the context of the Human Brain Project. Due to its approach for modelling neurons it is very compute intensive. The main kernel is a linear solver where dense matrices play a key role.

As of today the application is not yet used extensively. This is however expected to change in the near future.

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP
- **Suitable for simulations:** Yes
- **Characteristics:** Some of the main kernels are expected to be very suitable for SVE. Initial results towards an SVE port are already available

2.3 BT-MZ

The NAS multi-zone block tridiagonal (BT) benchmark [dWJ03] solves multiple, independent systems of equations. BT-MZ represents workloads similar to many flow solver codes (3D Navier-Stokes equations).

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP and MPI plus OmpSs
- **Suitable for simulations:** Yes
- **Characteristics:** Computation intensive, low memory contention that leads to good scalability.

2.4 CoMD

CoMD [HDC+09] is a reference implementation of typical classical molecular dynamics algorithms also included in the Mantevo suite. It evaluates all forces between atom pairs within a cutoff distance. It scales well with core counts.

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP
- **Suitable for simulations:** Yes
- **Characteristics:** Highly scalable code representative of n-body methods.

2.5 DL_POLY_Classic

DL_POLY_Classic is a general purpose classical molecular dynamics (MD) simulation software developed at Daresbury Laboratory [DLP].

- **Code accessibility:** Public
- **Programming model:** MPI
- **Suitable for simulations:** Maybe - Might be necessary to focus on specific kernels.
- **Characteristics:** Has high memory bandwidth and MPI communication requirements. It can run on one node or multi-node in function of the test-case.

2.6 graph500

The Breadth-First Search (BFS) kernel in graph500 [MWBA10] starts with a single source vertex, then, in phases, finds and labels its neighbors, then the neighbors of its neighbors, etc. Many graph algorithms are based on this method, where performance reflects an architectures throughput when executing concurrent threads, each of low memory concurrency and high memory reference density.

- **Code accessibility:** Public
- **Programming model:** OpenMP
- **Suitable for simulations:** Yes
- **Characteristics:** The memory access pattern is data-dependent, presenting low memory concurrency and high memory reference density. Suitable to explore memory hierarchy requirements for irregular access patterns.

2.7 GROMACS

GROMACS [gro] is a versatile package to perform molecular dynamics, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles. It is primarily designed for biochemical molecules like proteins, lipids and nucleic acids that have a lot of complicated bonded interactions, but since GROMACS is extremely fast at calculating the non-bonded interactions (that usually dominate simulations) many groups are also using it for research on non-biological systems, e.g. polymers.

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP
- **Suitable for simulations:** Maybe - Might be necessary to focus on specific kernels.
- **Characteristics:** Gromacs is suitable for vectorization and optimized with intrinsics. Has high memory bandwidth and MPI communication requirements.

2.8 High Performance Linpack (HPL)

The HPL benchmark [Don88] is a method of measuring the floating point rate of execution of a computer by running a program that solves a system of linear equations. It is well-known for being used to list the top500 fastest supercomputers.

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP
- **Suitable for simulations:** Yes
- **Characteristics:** Computation intensive, low memory contention.

2.9 HPCG

The High Performance Conjugate Gradients (HPCG) Benchmark [DHL15] was developed to create a new metric for ranking HPC systems. HPCG is intended as a complement to HPL and is designed to exercise computational and data access patterns that more closely match a different and broad set of important applications, as well as, to give incentive to computer system designers to invest in capabilities that will have impact on the collective performance of these applications. HPCG is a complete, stand-alone code that measures the performance of basic operations such as: sparse matrix-vector multiplication, vector updates, global dot products, and sparse triangular solvers.

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP
- **Suitable for simulations:** Yes - HPCCG Mantevo miniapp [HDC+09] is also avaliable
- **Characteristics:** Suitable for vectorization and with high memory contention.

2.10 HYDRO

HYDRO is a mini-application which implements a simplified version of RAMSES [Tey02], a code developed to study large scale structure and galaxy formation. HYDRO contains all performance relevant algorithms and communication patterns of the original application.

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP and MPI plus OmpSs
- **Suitable for simulations:** Yes
- **Characteristics:** Low memory intensity and relevant communication overheads that stress the communication network.

2.11 KKRnano

KKRnano implements a particular approach to Density Functional Theory (DFT) methods. DFT-based approaches play an increasingly important role in materials science. Currently this application is not yet used by a broader community, but this is expected to change.

The code has been developed with parallelism in mind. This allowed to achieve an exceptionally high scalability. The application can run with reasonable efficiency on all cores of the Blue Gene/Q at JUELICH.[1] Central to its performance is an iterative solver for the linear system and the application of the block-sparse operator. The blocks typically have a size of 16×16 resulting in many small matrix-matrix multiplications with complex operands. As a consequence the arithmetic intensity is high.

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP
- **Suitable for simulations:** Yes
- **Characteristics:** The main kernel is suitable for SVE and initial results towards an SVE port are already available.

[1] See: http://www.fz-juelich.de/ias/jsc/EN/Expertise/High-Q-Club/KKRnano/_node.html

2.12 Kripke

Kripke is a structured deterministic (Sn) transport using RAJA that contains wavefront algorithms. Is one of the miniapps featured in the new Coral benchmarks (`https://asc.llnl.gov/coral-2-benchmarks/`).

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP (written in C++)
- **Suitable for simulations:** Yes
- **Characteristics:** Stress memory latency and/or bandwidth, as well as network latency.

2.13 LQCD

Quantum Chromodynamics (QCD) is the theory of strong interactions. Its discretised version is called Lattice QCD (LQCD). In Europe, Japan and the US a very significant part of the supercomputing resources are used for LQCD simulations.

The main kernel, a iterative solver for a very sparse matrix, is highly scalable and falls in the class of regular grids. The Arithmetic Intensity (AI) is on today's architectures relative low, namely 0.45 or 0.9 bytes per flop (depending on whether double or single precision is being used).

LQCD here is still a placeholder for different possible applications from the area high energy physics. Possible candidates are: (i) BQCD[2], (ii) grid[3], (iii) tmLQCD[4]. For BQCD a mini-app version is already available. To all of these applications are highly scalable and the following applies:

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP
- **Suitable for simulations:** Yes
- **Characteristics:** The main kernel is suitable for SVE and initial results towards an SVE port are already available

2.14 LULESH

Livermore Unstructured Lagrangian Explicit Shock Hydrodynamics (LULESH) [HKG11] represents a typical hydrocode. LULESH approximates the hydrodynamics equations discretely by partitioning the spatial problem domain into a collection of volumetric elements defined by a mesh. It presents high instruction-level and memory-level parallelism with bursts of independent memory accesses.

- **Code accessibility:** Public
- **Programming model:** OpenMP and MPI plus OpenMP
- **Suitable for simulations:** Yes

[2] https://www.rrz.uni-hamburg.de/services/hpc/bqcd.html
[3] https://github.com/paboyle/Grid
[4] https://github.com/etmc/tmLQCD

- **Characteristics:** Burst of memory accesses that demand high bandwidth. It also stresses the core capacity to exploit instruction level parallelism.

2.15 MiniAMR

MiniAMR [HDC+09] applies a stencil calculation on a unit cube computational domain, which is divided into blocks. The blocks all have the same number of cells in each direction and communicate ghost values with neighboring blocks. With adaptive mesh refinement, the blocks can represent different levels of refinement in the larger mesh. A version with hand-coded SVE assembly is under development.

- **Code accessibility:** Public - SVE version private to BSC and ARM.
- **Programming model:** Serial and MPI
- **Suitable for simulations:** Yes
- **Characteristics:** SVE-friendly and memory bound as vector lenghts increase.

2.16 MontBlanc Benchmarks

The MontBlanc Benchmarks [RRV+13] are a collection of mini-applications and kernels representative of HPC workloads. The suite was developed within the MonBlanc project and includes: dense matrix multiplication, sparse matrix-vector multiplication, fast fourier transforms, stencil, n-body, and other kernels typically found in HPC applications.

- **Code accessibility:** Public
- **Programming model:** OpenMP and OmpSs
- **Suitable for simulations:** Yes
- **Characteristics:** The kernels are diverse and can be used to infer requirements for vectors (SVE), memory bandwidth, computational demands, as well as stress the network-on-chip.

2.17 NAMD

Recipient of a 2002 Gordon Bell Award and a 2012 Sidney Fernbach Award, is a parallel molecular dynamics code designed for high-performance simulation of large biomolecular systems [nam].

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP
- **Suitable for simulations:** Maybe - Might be necessary to focus on specific kernels.
- **Characteristics:** SVE can be suitable on specific kernels. High memory bandwidth and MPI communication requirements.

2.18 NEST

NEST is another already established brain simulator. It is based on point-neurons and mainly focusses on being able to maximise the size of the network that can be simulated. Due to its approach for modelling neurons it is very compute intensive. The application meanwhile is partially developed within the Human Brain Project.

The code has been proven to be scalable [5]. It should, however, be noted that for this application mainly the weak scaling limit is interesting, i.e. the limit of extremely large networks. Memory accesses are unstructured, control flow is irregular

As of today the application is not yet used extensively. This is however expected to change in the near future.

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP
- **Suitable for simulations:** Yes
- **Characteristics:** As of today it is unclear how this application could exploit SVE.

2.19 NTChem

NTChem [NKKN15] is a comprehensive software package that includes various high-performance computational methods and functions for quantum molecular simulations. It was designed for high-performance calculations on a computer with numerous compute nodes.

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP and MPI plus OmpSs
- **Suitable for simulations:** Yes
- **Characteristics:** Presents a wave-front communication pattern. Highly scalable to a large number of nodes.

2.20 PENNANT

PENNANT is a mini-app for hydrodynamics on general unstructured meshes in 2D (arbitrary polygons).

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP (written in C++)
- **Suitable for simulations:** Yes
- **Characteristics:** It makes heavy use of indirect addressing and irregular memory access patterns. Should vectorize (to be confirmed).

[5] http://www.fz-juelich.de/ias/jsc/EN/Expertise/High-Q-Club/NEST/_node.html

2.21 RAJA Performance Suite

The RAJA performance suite is designed to explore performance of loop-based computational kernels of the sort found in HPC applications. In particular, it is used to assess, monitor, and compare runtime performance of kernels implemented using RAJA and variants implemented using standard or vendor-supported parallel programming models directly.

- **Code accessibility:** Public
- **Programming model:** OpenMP (written in C++)
- **Suitable for simulations:** Yes
- **Characteristics:** HPC-specific computational loops. Potentially suitable for SVE vectorization.

2.22 Specfem3D

Specfem3D Cartesian [spe] simulates acoustic (fluid), elastic (solid), coupled acoustic/elastic, poroelastic or seismic wave propagation in any type of conforming mesh of hexahedra (structured or not.) It can, for instance, model seismic waves propagating in sedimentary basins or any other regional geological model following earthquakes. It can also be used for non-destructive testing or for ocean acoustics.

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP
- **Suitable for simulations:** Yes
- **Characteristics:** SVE can be suitable on specific kernels. High memory bandwidth and MPI communication requirements.

2.23 Quantum Espresso

Quantum Espresso is a significantly different type of DFT-method, which unlike KKRnano spends significant amounts of time in FFT. Other mainly BLAS level 3 operations are used.

Quantum Espresso is a very popular and used by various research groups in Europe but also in the US.

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP
- **Suitable for simulations:** Maybe - Might be necessary to focus on specific kernels.
- **Characteristics:** Some of the main kernels are suitable for SVE.

2.24 Quicksilver

Monte Carlo transport benchmark with multi-group cross section lookups.

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP (written in C++)
- **Suitable for simulations:** Yes
- **Characteristics:** Stresses memory latency, significant branching, and one large kernel that is 1000's of lines big. Poor vectorization potential.

2.25 SP-MZ

The NAS multi-zone scalar-pentadiagonal (SP-MZ) benchmark [dWJ03] is representative of many CFD applications. The system has Scalar Pentadiagonal bands of linear equations that are solved sequentially along each dimension.

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP and MPI plus OmpSs
- **Suitable for simulations:** Yes
- **Characteristics:** Stresses the memory hierarchy due to high contention.

2.26 WRF

The weather research and forecasting (WRF) model [WRF] is a numerical weather prediction system [wea] designed to serve both atmospheric research and operational forecasting needs.

- **Code accessibility:** Public
- **Programming model:** MPI plus OpenMP
- **Suitable for simulations:** Maybe - Might be necessary to focus on specific kernels.
- **Characteristics:** SVE can be suitable on specific kernels. High memory bandwidth and MPI communication requirements.

2.27 XSBench

The XSBench proxy app [TSIS14] models the most computationally intensive part of a typical MC transport algorithm - the calculation of macroscopic neutron cross sections - a kernel which accounts for around 85% of the total runtime of OpenMC. The essential computational conditions and tasks of fully featured MC neutron transport codes are retained in the mini-app.

- **Code accessibility:** Public
- **Programming model:** OpenMP
- **Suitable for simulations:** Yes
- **Characteristics:** Stresses system through memory capacity (including potential non-volatile random-access memory), random memory access, memory latency, threading, and memory contention. Thereby suitable to explore memory hierarchy requirements.

3 Summary Table of Selected Applications

Table 2 lists the selected applications and summarizes their characteristics. We have classified them according to the main computational dwarf [ABD+09] they belong, i.e., dense linear algebra, unstructured grids, etc. We cover all the HPC-relevant dwarfs with multiple representatives, to stress different workload characteristics that may appear within a dwarf, leading to an extensive coverage in terms of computational patterns. The characteristics we list are inline with the requirements to be studied within the project.

Table 2: Table of selected applications classified by computational dwarf.

Application	**HPC dwarf [ABD+09]**	**SVE friendly**	**Stress memory**		**Stress NoC**
			Bandwidth	**Capacity**	
ABINIT	spectral methods	✓	✓	✓	✓
Arbor	dense linear	✓	✗	✓	✗
BT-MZ	dense linear	✓	✗	✓	✗
CoMD	n-body methods	✓	✗	✗	✗
DL_POLY_Classic	n-body methods	✗	✓	✓	✓
graph500	graph algorithms	✗	✓	✓	✓
GROMACS	n-body methods	✓	✓	✓	✓
HPL	dense linear	✓	✗	✗	✗
HPCG	sparse linear	✓	✓	✓	✓
HPCCG	sparse linear	✓	✓	✓	✓
HYDRO	structured grids	✓	✗	✓	✓
KKRnano	dense linear	✓	✗	✗	✗
Kripke	structured grids	✗	✓	✗	✓
LQCD	structured grids	✓	✓	✓	✓
LULESH	unstructured grids	✓	✓	✓	✓
MiniAMR	structured grids	✓	✓	✓	✓
MontBlanc Benchs	all	✓	✓	✗	✓
NAMD	n-body methods	✓	✓	✓	✓
NEST	unstructured grids	✗	✗	✓	✗
NTChem	sparse linear	✗	✗	✗	✓
PENNANT	unstructured grids	✓	✓	✓	✓
RAJA Suite	set of HPC kernels	✓	✓	✗	✓
Specfem3D	unstructured grids	✓	✓	✓	✓
Quantum Espresso	spectral methods	✓	✓	✗	✓
Quicksilver	embarassingly parallel	✗	✓	✗	✓
SP-MZ	structured grids	✗	✓	✓	✓
WRF	structured grids	✓	✓	✓	✓
XSBench	embarassingly parallel	✗	✓	✓	✓

Acronyms and Abbreviations

- **BFS** **B**readth-**F**irst **S**earch
- **DFT** **D**ensity **F**unctional **T**heory
- **FFT** **F**ast **F**ourier **T**ransform
- **HPC** **H**igh **P**erformance **C**omputing
- **ISA** **I**nstruction **S**et **A**rchitecture
- **ISV** **I**ndependent **S**oftware **V**endors
- **MPI** **M**essage **P**assing **I**nterface
- **NoC** **N**etwork on **C**hip
- **OpenMP** **O**pen **M**ulti-**P**rocessing
- **SVE** **S**calable **V**ector **E**xtension

References

[ABD+09] Krste Asanovic, Rastislav Bodik, James Demmel, Tony Keaveny, Kurt Keutzer, John Kubiatowicz, Nelson Morgan, David Patterson, Koushik Sen, John Wawrzynek, David Wessel, and Katherine Yelick. A view of the parallel computing landscape. *Commun. ACM*, 52(10):56–67, oct 2009.

[DHL15] Jack J. Dongarra, Michael A. Heroux, and Piotr Luszczek. Hpcg benchmark: a new metric for ranking high performance computing systems. 2015.

[DLP] DL_POLY. `https://www.scd.stfc.ac.uk/Pages/DL_POLY.aspx`.

[Don88] Jack Dongarra. The linpack benchmark: An explanation. In *Proceedings of the 1st International Conference on Supercomputing*, pages 456–474, London, UK, UK, 1988. Springer-Verlag.

[dWJ03] Rob F. Van der Wijngaart and Haoqiang Jin. NAS Parallel Benchmarks, Multi-Zone Versions. Technical Report NAS-03-010, NAS, 2003.

[gro] GROMACS. `http://www.gromacs.org`.

[HDC+09] Michael A Heroux, Douglas W Doerfler, Paul S Crozier, James M Willenbring, H Carter Edwards, Alan Williams, Mahesh Rajan, Eric R Keiter, Heidi K Thornquist, and Robert W Numrich. Improving Performance via Mini-applications. Technical Report SAND2009-5574, Sandia National Laboratories, 2009.

[HKG11] R. D. Hornung, J. A. Keasler, and M. B. Gokhale. Hydrodynamics Challenge Problem. Technical Report LLNL-TR-490254, Lawrence Livermore National Laboratory, 2011.

[MWBA10] R. C. Murphy, K. B. Wheeler, B. W. Barrett, and J. A. Ang. Introducing the graph 500. Technical report, Cray User's Group, May 2010.

[nam] NAMD. `http://www.ks.uiuc.edu/Research/namd/`.

[NKKN15] Takahito Nakajima, Michio Katouda, Muneaki Kamiya, and Yutaka Nakatsuka. Ntchem: A high-performance software package for quantum molecular simulation. *International Journal of Quantum Chemistry*, 115(5):349–359, 2015.

[RRV+13] Nikola Rajovic, Alejandro Rico, James Vipond, Isaac Gelado, Nikola Puzovic, and Alex Ramirez. Experiences with mobile processors for energy efficient hpc. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '13, pages 464–468, San Jose, CA, USA, 2013. EDA Consortium.

[spe] Specfem3D. `https://geodynamics.org/cig/software/specfem3d/`.

[Tey02] Romain Teyssier. Cosmological hydrodynamics with adaptive mesh refinement-a new high resolution code called ramses. *Astronomy & Astrophysics*, 385(1):337–364, 2002.

[TSIS14] John R Tramm, Andrew R Siegel, Tanzima Islam, and Martin Schulz. XS-Bench – The development and verification of a performance abstraction for Monte Carlo reactor analysis. *The Role of Reactor Physics toward a Sustainable Future (PHYSOR)*, 2014.

[wea] Numerical Weather Prediction. https://en.wikipedia.org/wiki/Numerical_weather_prediction.

[WRF] WRF. https://en.wikipedia.org/wiki/Weather_research_and_forecasting_model.