PERSPECTIVE
https://doi.org/10.1038/s43588-021-00023-0

nature computational science

Check for updates

The digital revolution of Earth-system science

Peter Bauer[1]✉, Peter D. Dueben[1], Torsten Hoefler[2], Tiago Quintino[3], Thomas C. Schulthess[4] and Nils P. Wedi[1]

Computational science is crucial for delivering reliable weather and climate predictions. However, despite decades of high-performance computing experience, there is serious concern about the sustainability of this application in the post-Moore/Dennard era. Here, we discuss the present limitations in the field and propose the design of a novel infrastructure that is scalable and more adaptable to future, yet unknown computing architectures.

The human impact on greenhouse gas concentrations in the atmosphere and the effects on the climate system have been documented and explained by a vast resource of scientific publications, and the conclusion—that anthropogenic greenhouse gas emissions need to be drastically reduced within a few decades to avoid a climate catastrophe—is accepted by more than 97% of the Earth-system science community today[1]. The pressure to provide skillful predictions of extremes in a changing climate, for example, the number and intensity of tropical cyclones and the likelihood of heatwaves and drought co-occurrence, is particularly high because the present-day impact of natural hazards at a global level is staggering. In the period 1998–2017, over 1 million fatalities and several trillion dollars in economic loss have occurred[2]. The years between 2010 and 2019 have been the costliest decade on record with the economic damage reaching US$2.98 trillion—US$1.19 trillion higher than 2000–2009[3]. Both extreme weather and the potential failure to act on climate change rank as the leading risks combining maximum likelihood and impact for our future[4].

These losses do not invalidate the steady progress achieved in weather prediction over the past decades, that is, the combined result of improved observing systems, a better understanding of the relevant physical processes occurring and interacting in the Earth system and the exponential growth of general-purpose computing technology performance at nearly constant cost[5]. However, continuing at this pace is being questioned right now for two major reasons. First, the apparent effects of climate change on our environment—in particular on the frequency of occurrence and the intensity of environmental extremes—require urgent political response and much faster progress in delivering skillful predictions of future change[6,7]. Earth-system models need more than steady progress and make a leap to very high resolution, a more realistic representation of processes at all scales and their interaction between atmosphere, ocean, cryosphere, land surfaces and the biosphere. This leap will inevitably translate into a leap in our computational and data handling capacity needs. Second, the explosion of data challenges[8] and the demise of the 'laws' of Dennard and Moore[9] require a rethinking of the way we approach Earth-system modeling and high-performance computing (HPC) at extreme scales. These laws have been driving the development of microchips for decades. Dennard scaling states that shrinking feature sizes of transistors also decreases their power consumption such that the frequency could be increased from processor generation to the next while the heat dissipation per chip area remained approximately constant. Dennard scaling ended nearly 15 years ago and led to the 'multicore crisis' and the advent of commodity parallel processing. Moore's law drove the economics of computing by stating that every 18 months, the number of transistors on a chip would double at approximately equal cost. However, the cost per transistor starts to grow with the latest chip generations, indicating an end of this law. Therefore, in order to increase the performance while keeping the cost constant, transistors need to be used more efficiently.

In this Perspective, we will present potential solutions to adapt our current algorithmic framework to best exploit what new digital technologies have to offer, thus paving the way to address the aforementioned challenges. In addition, we will propose the concept of a generic, scalable and performant prediction system architecture that allows advancement of our weather and climate prediction capabilities to the required levels. Powerful machine learning tools can accelerate progress in nearly all parts of this concept.

The perfect application for extreme computing

Weather prediction has been a pioneering application of numerical computer simulations since John von Neuman's 'Meteorology Project' in the late 1940s[10,11]. Much has been achieved since then and today's operational global predictions are completed within an hour for models with about 10 million grid points, 100 vertical layers and 10 prognostic variables, initialized using 100 million observations per day. These calculations run on hundreds of nodes of general-purpose central processing units (CPU) offered by vendors in the US, Asia and Europe. The need to run simulation ensembles for predicting both state and uncertainty[12] multiplies both compute and data burden—but has proven hugely beneficial for decision-making[13].

Figure 1 illustrates the elements of an operational weather prediction workflow, in which steps 2–4 are very compute- (peta-flops) and data- (100 terabytes per day) intensive. Weather simulations are different from climate simulations as they are run in burst mode at given times per day while climate predictions are run in steady-production mode to complete multi-decadal, centennial and millennial projections of the climate.

Given the computational constraints, weather and climate models have diverged in the past decades: climate models need to represent closed and stable energy, water and constituent cycles at the expense of small-scale process detail; weather models, on the other hand, need this level of detail for locally accurate forecasts, but choose to exclude those Earth-system processes that are less relevant for weather on day-to-season time scales. For example, the accurate description of water-cycle processes is highly relevant for

[1]Research Department, ECMWF, Reading, UK. [2]Computer Science Department, ETH Zürich, Zürich, Switzerland. [3]Forecast Department, ECMWF, Reading, UK. [4]CSCS, ETH Zürich, Lugano, Switzerland. ✉e-mail: peter.bauer@ecmwf.int

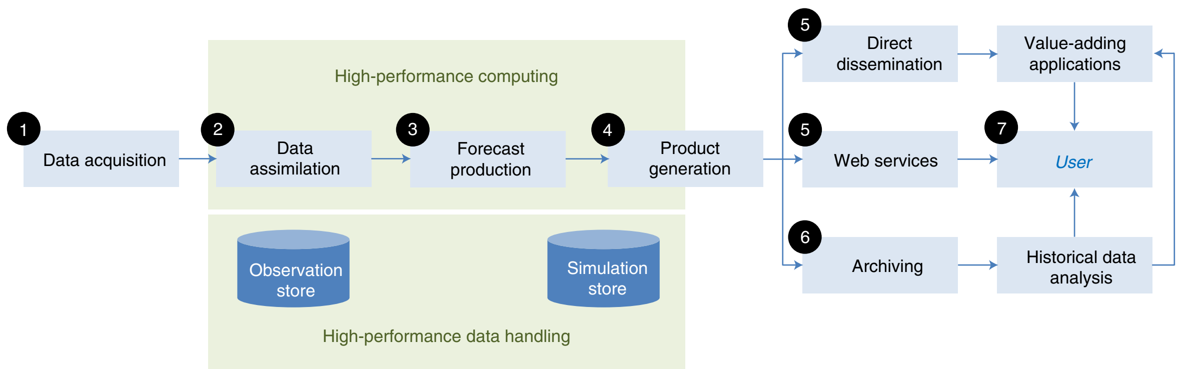


Fig. 1 | Typical production workflow in operational numerical weather prediction. (1) High-volume and high-speed observational data acquisition and pre-processing; (2) data assimilation into models to produce initial conditions for forecasts; (3) forecast production by Earth-system simulation models; (4) generation of output products tailored to the portfolio of weather and climate information users; (5) direct dissemination of raw output and web-products; (6) long-term archiving for reuse in statistical analyses and performance diagnostics; (7) user-specific applications and data-driven analytics.

both weather and climate models while the representation of the carbon cycle is only important at climate time scales. Increasingly though, the recognition that small scales matter for climate predictions[14] and that Earth-system complexity matters for weather prediction[15] dawns on our community and is leading to converging developments. As a consequence, we need both very high resolution and Earth-system process complexity.

Stretching the computing limits to what is available on the fastest supercomputers in the world allows us to gauge how much more realistic very high-resolution simulations become[16,17] (Fig. 2) but also what the computing footprint with existing codes would be[17,18]. These experiments show that—only for forecasts—these computers cannot fully deliver the throughput required to produce high-resolution simulations of fully coupled Earth-system models, and the data volumes these simulations would produce cannot be handled effectively. This makes future weather and climate predictions an extreme-scale computing and data-handling challenge.

The urgency of climate change and the need for much faster progress than in the past translates into much more than only a forecast model upgrade. To build an information system in support of policy- and decision-making, the workflow shown in Fig. 1 needs to be extended to weather- and climate-dependent applications like energy, food, water and disaster management and to add flexibility for testing both scientific and socio-economic scenarios. This information system is called a digital twin[19] (Box 1). The twin produces a digital replica of the real world through simulations and observations with much more physical realism than it is possible today and by fully integrating impact sectors and human behavior in the Earth system. With the advent of cyber-physical systems in the context of the fourth industrial revolution[20], this concept is being increasingly applied to other areas beyond engineering[21]—in our case, weather and climate prediction.

Code adaptation to new technologies

Traditional practices. The record of continual code adaptation to emerging technology reaches back to the 1970s when supercomputers became commercially available and used by prediction centers. The main disruption in technology—the move from vector to scalar processors in the 1990s[22]—coincided with a period where models substantially increased spatial resolution benefiting from much enhanced parallelism[23]. Since then, these codes have profited from Moore's law[24] and Dennard scaling[25] without much pressure to fundamentally revise numerical methods and programming paradigms.

This has led to very large legacy codes, primarily driven by scientific concerns, leaving very little room for computational science innovation[26]. The result is that such codes only achieve around 5% sustained floating-point performance on present-day CPU machines[27], which sufficed as long as CPU technology delivered exponential performance growth in clock-speed, memory size and access speed. Now, as this growth is stopping and energy cost is rising, a computing 'chasm' looms[28] that our community has to overcome to deliver better and more cost-effective predictions.

Earth-system models discretize the set of physical equations for the resolved processes in space and time[29] and use parameterizations for unresolved processes such as cloud microphysics and turbulence, which impact the prognostic variables at the resolved scales[30]. The same applies to data assimilation, whose computing performance is mostly driven by the forecast model and coupled components representing ocean processes, surface waves, sea-ice, land surfaces including vegetation and so forth in the Earth system[31]. Different choices of discretization imply different solvers with specific patterns for memory access and data communication per time step. The time step itself is an important cost factor and depends on the choice of discretization[32,33], but is also constrained by the non-linearity of the problem and the type and speed of motions to be resolved[34].

There have been several programs aiming to substantially accelerate weather and climate prediction code infrastructures in the past decade. However, one would call these improvements 'traditional' because they refrain from touching the basic algorithmic concepts and work along known science software development paths. The code is primarily written by scientists and then computer scientists extract performance by incrementally refactoring code, typically improving memory and communication handling, and by introducing code directives to exploit parallelism and vectorization based on standard programming models.

More recently, the option of precision reduction below the default of double precision has been investigated to improve bandwidth and computational throughput[35–37]. The precision reduction below single precision is non-trivial in a complex, non-linear weather model[38]. Another route has been to advance the concurrent execution of different model sub-components, thus breaking up the classical, strictly sequential execution of physical process calculations per time step[39]. This is also relevant where sea-ice and ocean dynamics calculations are co-executed[40]. More generally, overlapping computing and data transfer can speed up individual numerical algorithms that heavily rely on data communication[41,42], or accelerate workflows in which data analysis and post-processing run concurrently with the model[43].

Porting computing intensive code parts to novel architectures such as graphics processing unit (GPU)-accelerated systems and

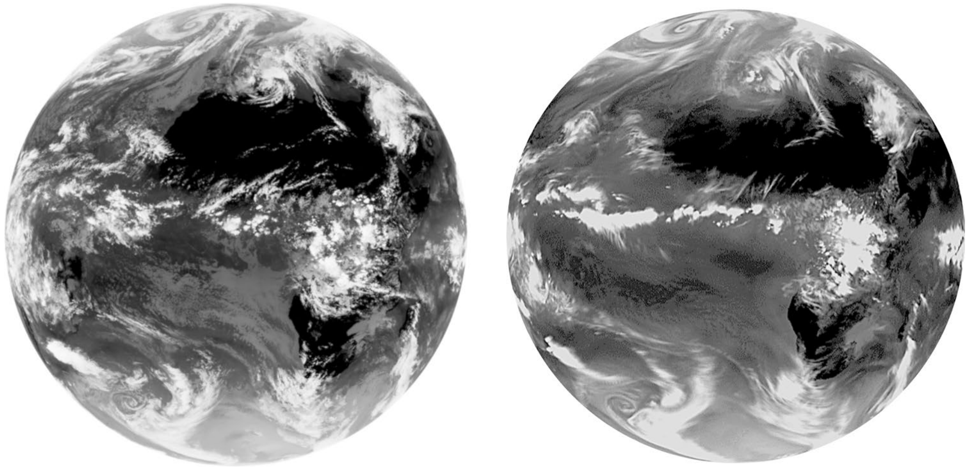


Fig. 2 | Comparison between observed and simulated satellite imagery. The satellite data (left) were obtained by Meteosat Second Generation's Spinning Enhanced Visible and Infrared Imager and represent the emitted radiances at infrared wavelengths. The simulation (right) was produced with the ECMWF Integrated Forecasting System at 1 km spatial resolution.

many-core processors has shown good results, but often requiring laborious code rewrites. An early effort based on Fortran to Compute Unified Device Architecture (CUDA) source-to-source translation succeeded in making the global Non-hydrostatic Icosahedral Model (NIM) of the National Oceanic and Atmospheric Administration (NOAA) portable across multiple architectures, including NVDIA GPUs[44]. A rewrite of the Consortium for Small-scale Modeling (COSMO) dynamical core along with porting of physics parameterization[45] resulted in the first fully operational, limited-area climate and weather model running on GPU-accelerated systems. A very large effort is presently underway in the US Department of Energy's (DoE) Exascale Computing Project (ECP) to evolve the Energy Exascale Earth System Model (ESMD/E3SM) to novel computing architectures[46]. The US National Center for Atmospheric Research (NCAR) high-resolution version of the Community Earth System Model (CESM) code has been extensively adapted and optimized for the heterogeneous management/computing processing element architecture on the Sunway TaihuLight supercomputer[47]. Furthermore, the Met Office is leading a large project in the UK to implement the successor to the Unified Model (UM) in such a way that any conceivable future architecture can be supported[48,49]. In Japan, both high-resolution modeling and large ensemble data assimilation developments break similar barriers on the world's largest supercomputing facilities[50,51]. The following modern code design practices are likely to emerge from these efforts.

Modern practices in co-designing algorithms and computing. Recent performance assessments show that present codes fall way short of the throughput targets needed for operational production[18]. Traditional code adaptation will not be sufficient to achieve the necessary efficiency gains and manual adaptation is not sustainable as technology keeps changing. Therefore, the suitability of the basic algorithmic framework needs to be scrutinized[52] and new data-driven methodologies like machine learning need be incorporated where they promise savings without loss of quality[53]. Since digital technologies evolve rapidly, both performance and portability matter. The ultimate goal is to avoid technology lock-in as well as algorithmic lock-in.

Data structures and discretization. When investing in more intrusive measures to enhance performance, a few basic architectural building blocks require attention, such as spatial discretization, forward-in-time time stepping and the (intrinsic) coupling of Earth-system components, all of which strongly rely on data structures. The actual performance metrics should reflect the complexity of the entire problem, and this goes well beyond achievable floating-point operation rates[18,54].

As time-stepping algorithms and choices of spatial discretization combined with particular advection transport schemes are not independent[30,34], substantial speedups can be obtained by making the appropriate choice. On existing architectures, (semi-)implicit numerical schemes offer such speedups because large time steps produce stable solutions despite the drawback of additional communications[18,55]. This is in comparison to inherently local, explicit schemes with higher-order discretization stencils, which pay a high price for achieving numerical stability with small time steps to capture fast evolving processes.

Both efficiency and accuracy can be achieved by combining large-time-step methods with higher-order discretizations[55]. Other solutions offer efficiencies through different time steps used for different Earth-system components, splitting vertical from horizontal advective transport[56], full coupling of the discretized dynamical equations, and the same computational pattern being repeatedly applied, for example, for the advection scheme or the vertical-column physical process simulation across atmosphere and ocean.

As for reduced precision, it is still unknown how this will affect slow error growth in the global mean model state at long time scales. It does not help that many different time- and length-scales of weather and climate processes interact non-linearly with each other leading to a continuous rather than well separated spectrum of motions[57], in contrast with other multi-physics applications where processes and their computations can be readily split due to their vastly different time and length scales.

Another approach concerns parallel-in-time methods, which have received renewed interest because of the advent of massively parallel computers[58]. In contrast with spatial discretization, the particular problem for parallelizing time in weather and climate applications is to consider the dependence on the time history of the flow and maintaining the accuracy and numerical stability of the integrations[59,60].

Tightly linked to discretization and the connectivity choices of grids and meshes is the overall data structure of models. The complexity of weather and climate models does not readily allow flexibly changing data structure or use asynchronous data-flow programming models[61]. Existing structures are often explicitly or implicitly

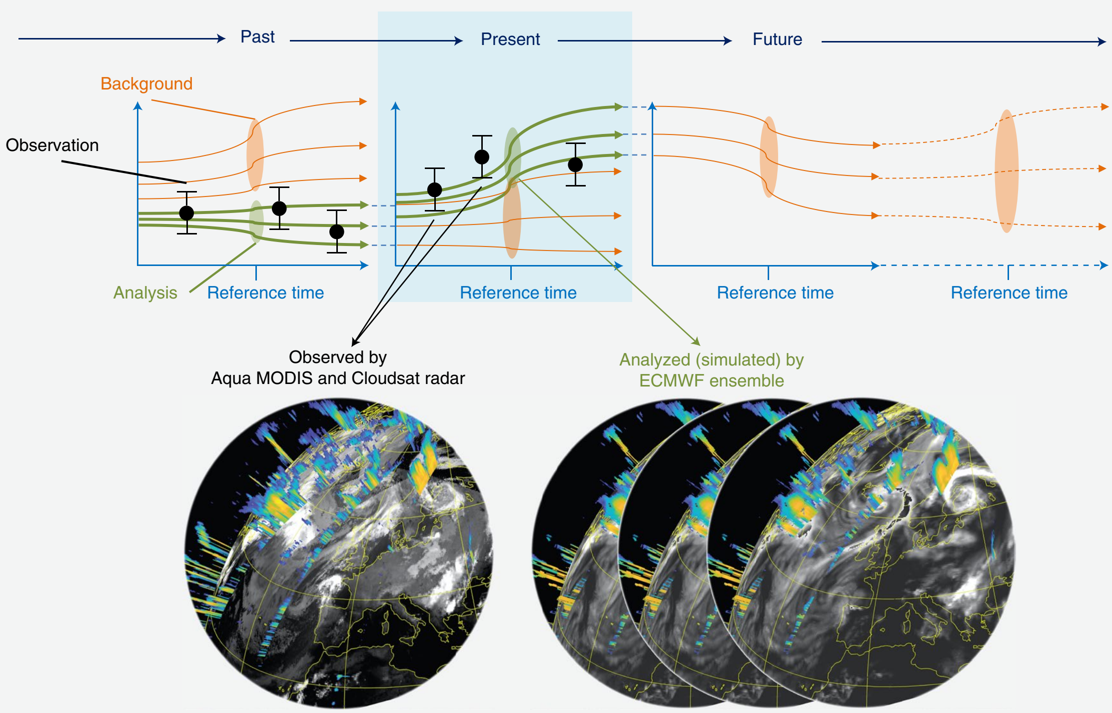
Box 1 | Digital twins

Digital twins were created for industrial production and space technology engineering processes. Their goal is to optimize design and operations of complex processes through a highly interconnected workflow combining a digital replica of the process with real-time observations of the physical system. The observations allow supervision of performance and health of its components, so that performance optimization, autotuning and resilience measures can be applied on the fly.

The Earth-system digital twin, shown in the figure, optimally combines simulations and near-real-time observations to monitor the evolution of the Earth system. For each cycle, the simulation generates a background forecast ensemble (orange arrows) of the Earth system, which is compared to observations (black dots) throughout a time window and eventually corrected to an analysis ensemble (green arrows), which fits the observations better than the background. Uncertainties of forecasts (ellipses) and observations (error bars) are fully taken into account from ensembles, which are multiple, perturbed realizations of both model and observations. Analysis uncertainties become smaller than background uncertainties from using the observational constraint (smaller spread of green versus orange trajectories). The pictures below show a real example from this procedure using observations from the Moderate Resolution Imaging Spectroradiometer (MODIS) infrared radiometer on board the National Aeronautics and Space Administration's (NASA) Aqua satellite and CloudSat cloud radar reflectivity cross-sections (bottom left) and the digital-twin simulation from assimilating this data (bottom right). This methodology is based on data assimilation and has been used in weather forecasting since the 1990s[107,108] and, more recently, in support of climate prediction[109].

The simulation–observation fusion is performed in space and over a time window whereby the model ensures that the optimum, physically consistent evolution is produced accepting that observations do not measure all state variables everywhere all the time. This optimization has a huge computing footprint as the Earth-system state comprises billions of degrees of freedom and deals with a non-linear and chaotic system.

The extension of present-day capabilities to digital twins that operate at much higher resolution, complexity, with much more diverse observations and that include weather- and climate-dependent impact models and observations produces one of the most challenging applications for digital technologies and requires sustained international research and development programs[110,111].



tied to a specific structured or unstructured grid arrangement on which the algorithms operate. More generic approaches can anticipate where data resides and where it will be next, but can also help exploit an increasing hierarchy of memory layers on emerging hardware platforms[62].

Performance and portability. Digging even deeper into model and data assimilation architectures requires breaking up codes into domain-specific key algorithmic motifs and encapsulating them in separate, mid-sized applications with well-defined application programming interfaces (API). This has greatly helped to identify their specific performance bottlenecks and to adapt them to different hardware architectures with alternative programming models and alternative algorithms[63]. Building such mid-sized applications and sharing them with vendors and academia has been a popular approach[64] also to widen the perspective on the key elements of weather and climate models, while extending such research beyond atmospheric applications to numerical algorithms used in ocean, wave, sea-ice and biogeochemistry models.

While a bespoke implementation on a specific HPC architecture can return substantial speedups[65], achieving performance without sacrificing portability is seen as crucial to avoid a solution where

Table 1 | Stepwise estimate of hybrid CPU–GPU machine size for digital-twin computing based on COSMO benchmark[18,30] accounting for known, near-future technology upgrades and methodological redesign as described in this paper

Upgrade and technology	Acceleration factor	Remaining shortfall factor	Reference
5,000 Intel Xeon E5-2690 v3/Nvidia P100	1	100	[18]
Data structures, grids, numerical methods, mixed precision, machine learning	4	25	[18]
GPU bandwidth efficiency and bandwidth	2	13	Example, NVIDIA V100 vs P100[104]
GPU peak bandwidth and memory	1.5	8	Example, NVIDIA A100 vs V100[105]
High-bandwidth memory	2	4	Example, HBM3 vs HBM2[106]

Acceleration and shortfall factors describe the expected acceleration of a digital-twin benchmark delivered by each technology upgrade and the remaining shortfall of achievable time to solution, respectively.

one has to continuously rewrite complex software for a particular hardware option. Today, most models and data assimilation systems are still based on millions of lines of Fortran code. In addition, they adopt a fairly rigid block-structure in the context of a hybrid parallelization scheme using the Message Passing Interface (MPI) and Open Multiprocessing (OpenMP) combined with a domain knowledge-inspired or convenience-driven data flow within the model application[66].

The basis for entirely revising this approach are again generic data structures and domain-specific software concepts that separate scientific code from hardware-dependent software layers—distinguishing between the algorithmic flexibility concern of the front-end and the hardware flexibility concern of the back-end[67]. Ideally, this results in a single data structure view of the entire complex coupled application across a range of architectures, which is used in the entire workflow of observation handling, simulation, assimilation, I/O, post processing and archiving data[68–71].

Such domain-specific software framework developments are currently being pursued by the DoE-supported rewrite of the E3SM climate model[72] based on the C++ library Kokkos[73], to achieve performance portability on GPUs and CPUs. The UK Met Office, in collaboration with partners in academia, has developed a software framework called PsyClone[49]. MeteoSwiss and the Swiss National Supercomputing Centre CSCS pioneered the use of embedded domain-specific language constructs through their COSMO adaptation based on the C++ STELLA/Gridtools library[74], all with performance portability on energy efficient, heterogeneous hardware in mind. This has increased the popularity of code-generation tools and a fundamental rethinking of the structure and separation of concerns in future model developments, promising a route to radically rewrite the present monolithic and domain-specific codes. Beyond CPU and GPU, this approach would also support specialized data-flow processors like field-programmable gate arrays (FPGA) or application specific integrated circuits (ASIC).

Machine learning. Despite the recent flurry of machine learning projects, it is still difficult to predict how the application of machine learning will shape future developments of weather and climate models. There are approaches to build prediction models based on machine learning that beat existing predictions systems, in particular for very short (for example, now-casting[75]) and very long (for example, multi-seasonal[76]) forecasts, but also for medium-range prediction[77]. However, the majority of the weather and climate community remains skeptical regarding the use of black-box deep-learning tools for predictions and aims for hybrid modeling approaches that couple physical process models with the versatility of data-driven machine learning tools to achieve the best results[53].

In any case, machine learning is here to stay and has already had a notable impact on the development of all of the components of the prediction workflow that is visualized in Fig. 1, for example, in now-casting and observation processing[78], data assimilation[79,80], the forecast model (for the emulation of parameterization schemes[81,82] and parameter tuning[83]), and post-processing (for example, in feature detection and downscaling applications[84,85], and uncertainty quantification[86,87]).

Still, the impact of machine learning on weather and climate modeling goes beyond the development of tools to improve prediction systems. Artificial intelligence is a multi-trillion US$ market[88]—a multiple of the same value for the entire supercomputing market[89]—and machine learning will keep having a strong impact on hardware developments in the future. While co-designed processors are developed for deep-learning applications—such as the tensor processing unit (TPU)—commodity hardware for the general HPC market will have accelerators for deep learning, such as the Tensor Cores on NVIDIA Volta GPUs. Machine learning also has a strong impact on CPU and interconnect technologies, and compute system design.

Special machine learning hardware is optimized for dense linear algebra calculations at low numerical precision (equal or less than half precision) and allows for substantial improvements in performance for applications that can make use of this arithmetic. While the training and inference of complex machine learning solutions show the best performance on GPU-based systems[84] at the moment, most weather and climate centers still rely on conventional CPU-based systems. While the reduction of precision to three significant decimal digits—as available in IEEE half precision—is challenging but not impossible[90], no weather and climate model is able to run with less than single precision arithmetic yet. As tests to use machine learning accelerators within Earth-system models are in their infancy[37], the weather and climate community is largely unprepared to use hardware optimized for machine learning applications. On the other hand, the use of machine learning accelerators and low numerical precision comes naturally when using deep-learning solutions within the prediction workflow, in particular if used to emulate and replace expensive model components that would otherwise be very difficult to port to an accelerator, such as the physical parameterization schemes or tangent linear models in data assimilation[91,92]. Thus, machine learning, and in particular deep learning, also shows the potential to act as a shortcut to HPC efficient code and performance portability.

The Earth simulation machine

Proposing a computing infrastructure that optimally serves all aspects of weather and climate prediction is nearly impossible as the workflows are extremely complex given the large variety of data pre-/postprocessing and high-throughput computing steps—exemplified by the digital-twin concept. Box 1 explains the digital-twin concept and its foundation on the continuous fusion of simulations and observations based on information theory.

Given these constraints, we focus on a machine and software ecosystem that addresses the extreme-scale aspects of the digital twin most effectively. For this, we pose three questions: (1) What are the digital-twin requirements? (2) What is the most effective and sustainable software ecosystem? (3) What technology and machine size can run digital twins in the near future?

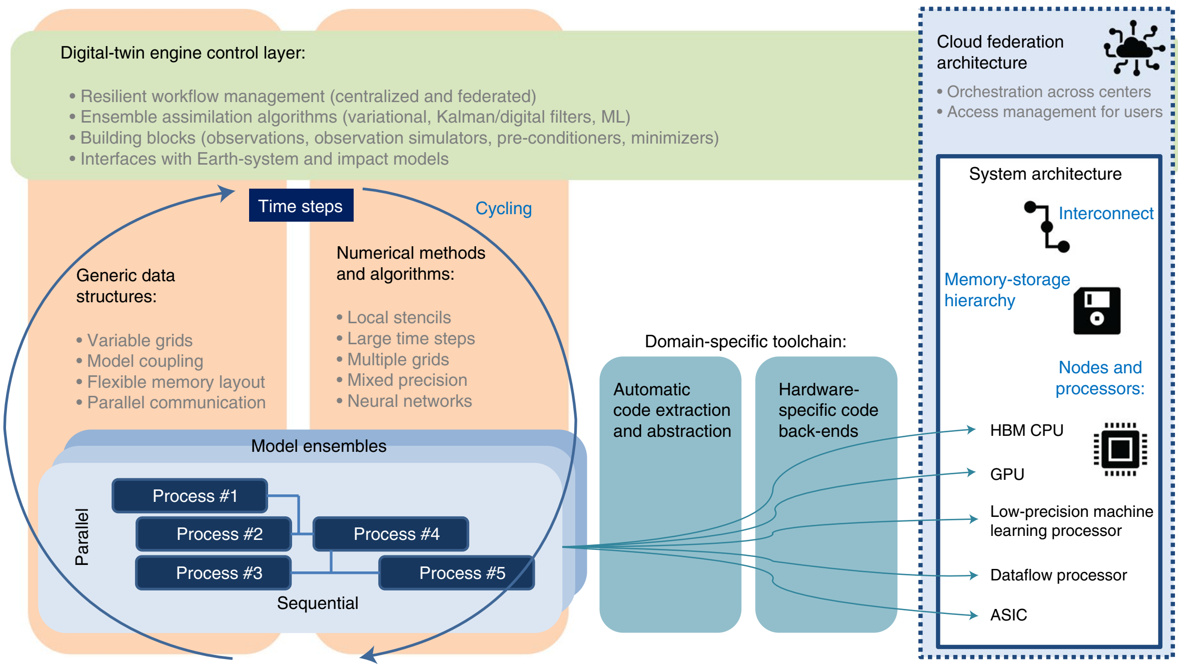


Fig. 3 | Conceptual view of an efficient software infrastructure for the Earth-system digital twin. The digital-twin control layer drives flexible workflows for Earth-system modeling and data assimilation using generic data structures and physical process simulations that exploit parallelism and are based on algorithms minimizing data movement. DSLs map the algorithmic patterns optimally on the memory and parallel processing power of heterogeneous processor architectures. The computing architecture is based on heterogeneous, large-scale architectures within federated systems.

Application requirements. Following the digital-twin definition in Box 1, its extreme-scale computing requirement is mostly driven by the forecast model itself. Even though the twin is based on a huge ensemble optimization problem using both simulations and observations, its efficiency and scalability is determined by the model. Observation processing and matching observations with model output is comparably cheap. The optimization procedure itself is mostly based on executing model runs in various forms and performing memory-intensive matrix operations. The digital-twin benchmark would use a very high resolution, coupled Earth-system model ensemble noting that a spatial resolution increase has the largest footprint on computing and data growth[18]. When refining the simulation grid by a factor of two in the horizontal dimensions, the computational demand roughly grows by a factor of eight, since doubling the resolution in each of the two spatial dimensions requires a commensurate increase in the number of time steps taken by the simulation. The ensemble mode multiplies the requirement by as many ensemble members as are required; however, lagged ensembles and using machine learning as a cheaper alternative for characterizing uncertainty[87] can produce substantial efficiency gains.

Software ecosystem. According to what we covered in the previous sections, a computing and data aware algorithmic framework based on flexible control and data structures can drastically reduce the computing and data footprint. In addition, such framework must overlap the execution of individual model components, focus on stencil operations with little data movement overhead, stretch time steps as much as possible and reduce arithmetic precision. Machine learning will produce further savings through surrogate models.

Apart from producing cost savings, the revised algorithmic framework also facilitates the implementation of more generic software infrastructures making future codes more portable and therefore sustainable. However, it is important to note that implementing high-performance codes in low-level environments is not simple and requires strong human expertise. We propose a strict separation of concerns of the programming problem into a productive front-end (for example, a Python-based domain-specific software framework for the relevant computational patterns) and an intermediate representation (for example, the multi-level Intermediate Representation (MLIR)[93] or Stateful DataFlow multi-Graphs (SDFG)[94]) for optimization that can then generate tuned code for the target architectures. A similar approach is used in machine learning where models are written with either PyTorch or TensorFlow and then compiled into optimized library calls using tools such as Accelerated Linear Algebra (XLA) or TensorRT. We expect that the design of the front-end will be specialized to our domain or at least to certain computational patterns, while many of the optimizations and transformations on the intermediate representation (for example, loop tiling and fusion) can be re-used across multiple domains. Thus, the performance engineering work can utilize existing investments and also benefit from other science disciplines as well as machine learning.

A candidate machine. The end of Moore's law and Dennard scaling forces us to consider different architectural variants in order to use each transistor most efficiently. A domain-specific, weather and climate architecture design would need to be manufactured in an advanced silicon process to be competitive in terms of energy consumption and performance. To maximize performance and cost effectiveness it is necessary to use the latest, smallest fabrication processes. While manufacturing costs grow very quickly towards the latest processes, performance grows even faster. For example, reducing transistor size from 16 nm to 5 nm results in a five-fold cost growth[95] while the transistor density and performance grows by a factor of six. As low-cost commoditization only happens at the low-performance end, building high-performance domain-specific architectures today would require a huge market such as deep learning where hundreds of millions of dollar investments can be made. This means that true weather and climate domain architecture

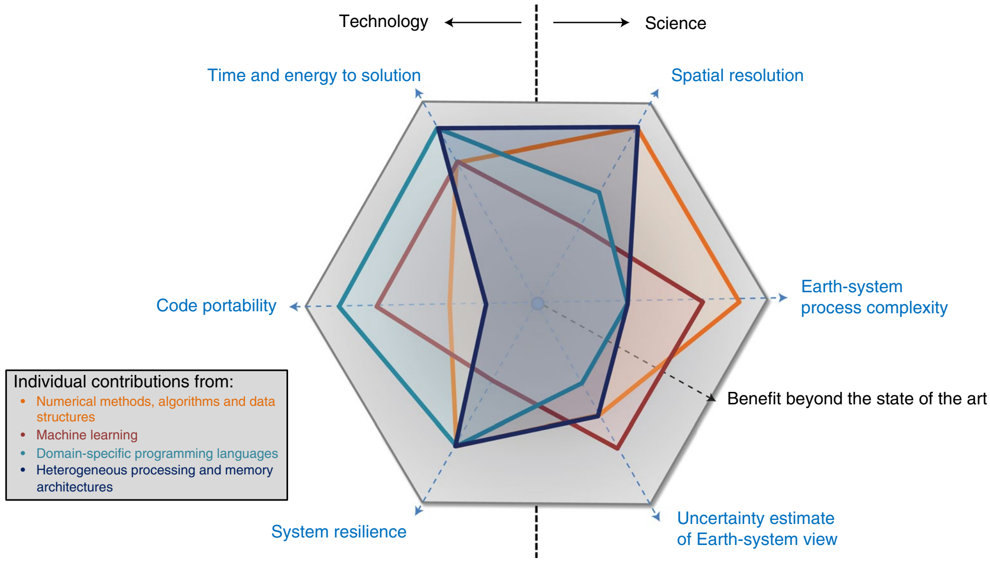


Fig. 4 | Expected contribution of main system developments necessary to achieve key science and computing technology performance goals. The distance from the center of the hexagon indicates the magnitude of the individual contributions towards enhanced efficiency for increased spatial resolution, more Earth-system complexity and better uncertainty information provided by ensembles as well as resilient, portable and efficient code and workflow execution, respectively.

co-design may not be possible unless funding commensurate with the scale of the climate change impact cost would become available.

If we resort to commodity devices that have a large volume market and enable high-performance specialized computations, we are limited to either vectorized CPUs, highly threaded GPUs or reconfigurable FPGAs. All these devices are manufactured in the latest silicon processes and offer high-performance solutions. Most of the energy in information processing systems is spent moving data between chips or on the chip[96]. Only a very small fraction of the energy is actually consumed to perform calculations. This is due to various control overheads in today's architectures, and innovations in accelerators mainly aim to reduce these control overheads[97]. Two prime examples are wide vectorization as implemented in the Fujitsu A64FX CPU or wide single instruction, multiple thread (SIMT)-style GPU machines as in NVIDIA's A100 accelerator. From investigating bounds for stencil programs that are common in weather and climate codes on each of these device types[98] we can conclude that the latest highly vectorized CPUs can be competitive with GPUs if their memory bandwidths match. Unfortunately, high-bandwidth memory was only recently added to FPGAs so that they will still be outperformed by GPUs in the near future[99].

Thus, a pragmatic option for today is a CPU–GPU-based solution. However, if industry continues the road of hardening floating-point logic on a reconfigurable fabric (similar to Intel's Stratix 10) and adding high-bandwidth memory connections, then the resulting CGRA-style (coarse-grained reconfigurable architectures) devices could surpass GPU and CPU performance and energy efficiency. This technological uncertainty also makes it imperative to implement new codes in a performance-portable language, which we suggested already above. The most competitive architecture for the next years will therefore likely be GPU-accelerated systems for which we need a rough size estimate now.

The previously cited benchmark runs used a single, high-resolution model forecast and estimated efficiency gain factors of 100 to comply with the operational one-year-per-day simulation throughput requirement[17,18,27]. This estimate included a model using today's algorithms and a nearly optimal, yet manual code adaptation to 5,000 GPU accelerators on the Piz Daint[100] system with one CPU host and one P100 GPU accelerator per node and an overall power envelope of 4 MW. Extrapolating this to near-future technology produces an estimate of a remaining shortfall factor of four thus requiring about 20,000 GPUs to perform the digital-twin calculations with the necessary throughput (Table 1). This machine would have a power envelope of about 20 MW. Whether the 5,000 GPU estimate can simply be extrapolated depends on the benchmark's strong scaling limit. Several of these systems are already in production to inspire a detailed machine design. For example, Summit and its successor Frontier present advanced CPU–GPU technology solutions at extreme scale. The European Large Unified Modern Infrastructure (LUMI), Leonardo, and MareNostrum5 systems provide similar technology options[101].

An important consideration in machine design is balance. Specifically, our machine would need to balance well computation, memory, and storage performance given that the storage/compute price trade-off can easily be adjusted given partial recomputation[102]. The specific design should be tuned to our domain with an emphasis on data movement over raw floating-point performance given the available hardware at the specific time.

An HPC system of sufficient size also creates an environmental footprint that needs to be taken into account. According to the US Environmental Protection Agency, which accounts about 1,000 lb CO_2 output per MWh, such a simulation machine, if it was built in places where only 'dirty' power is available, would produce substantial amounts of CO_2 per year. Performance and efficiency therefore need to make the operation not only economical but also environmentally friendly due to large power consumption rates.

Conclusion and outlook

The synergy of these developments is summarized as a conceptual view of the entire proposed infrastructure in Fig. 3. Workflow and algorithmic flexibility are provided by generic control layers and data structures supporting a variety of grid lay-outs, numerical methods and overlapping as well as parallelizing model component (process) execution and their coupling. Machine learning can deliver both computational efficiency and better physical process descriptions derived from data analytics. Codes follow the separation-of-concerns paradigm whereby front-end, highly legible science code is separated from hardware specific, heavily optimized code back-ends. The link is provided by a domain-specific software tool-chain. The system architecture maximizes both time and energy to solution and exploits both centralized and cloud-based deployments. It is important to understand that computing hardware and software advance on vastly different time scales. The lifetime of software can be decades while high-performance hardware is usually used for less than five years. The proposed algorithmic and software investments should therefore provide utmost flexibility and openness to new, fast evolving technology.

By how much all these factors will reduce the cost has not yet been fully quantified, but Fig. 4 gives our estimate of the potential relative impacts of the contributions outlined in this paper. The optimum system design requires these contributions to be developed together—as they are co-dependent—so that the resulting overall benefit beyond the state of the art can be fully achieved.

Computer system development and innovation never stop. The best price–performance point will quickly shift and in three years, a system design will likely look very different. For example, we could imagine software breakthroughs to happen that will make very low precision arithmetic viable in Earth-system science computations, thus drastically reduce memory and data communication overheads. Hardware breakthroughs in reconfigurable or spatial[103] as well as analog computing[63] may also become competitive.

The societal challenges arising from climate change require a step-change in predictive skill that will not be reachable with incremental enhancements, and the time is ripe for making substantial investments at the interface between Earth-system and computational science to promote the revolution in code design that is described in this paper. The cost of this effort is small compared to the benefits.

Received: 7 September 2020; Accepted: 12 January 2021;
Published online: 22 February 2021

References

1. Cook, J. et al. Quantifying the consensus on anthropogenic global warming in the scientific literature. *Environ. Res. Lett.* **8**, 024024 (2013).
2. Wallemacq, P., Below, R. & McLean, D. *Economic Losses, Poverty and Disasters: 1998–2017* (UNISDR, CRED, 2018).
3. *Weather, Climate and Catastrophe Insight* Report GDM05083 (AON, 2019).
4. Franco, E. et al. *The Global Risks Report 2020* (World Economic Forum, 2020).
5. Bauer, P., Thorpe, A. & Brunet, G. The quiet revolution of numerical weather prediction. *Nature* **525**, 47–55 (2015).
6. Hausfather, Z., Drake, H. F., Abbott, T. & Schmidt, G. A. Evaluating the performance of past climate model projections. *Geophys. Res. Lett.* **47**, e2019GL085378 (2020).
7. Sillmann, J. et al. Understanding, modeling and predicting weather and climate extremes: challenges and opportunities. *Weather Clim. Extremes* **18**, 65–74 (2017).
8. Asch, M. 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. *Int. J. High Perform. Comput. Appl.* **32**, 435–479 (2018).
9. Khan, H. N., Hounshell, D. A. & Fuchs, E. R. Science and research policy at the end of Moore's law. *Nat. Electron.* **1**, 14–21 (2018).
10. Platzman, G. W. The ENIAC computations of 1950—gateway to numerical weather prediction. *Bull. Amer. Meteorol. Soc.* **60**, 302–312 (1979).
11. Lynch, P. *The Emergence of Numerical Weather Prediction: Richardson's Dream* (Cambridge Univ. Press, 2006).
12. Leutbecher, M. & Palmer, T. N. Ensemble forecasting. *J. Comput. Phys.* **227**, 3515–3539 (2008).
13. Zhu, Y., Toth, Z., Wobus, R., Richardson, D. & Mylne, K. The economic value of ensemble-based weather forecasts. *Bull. Amer. Meteorol. Soc.* **83**, 73–84 (2002).
14. Palmer, T. & Stevens, B. The scientific challenge of understanding and estimating climate change. *Proc. Natl Acad. Sci. USA* **116**, 24390–24395 (2019).
15. Brunet, G. et al. Collaboration of the weather and climate communities to advance subseasonal-to-seasonal prediction. *Bull. Amer. Meteorol. Soc.* **91**, 1397–1406 (2010).
16. Stevens, B. et al. DYAMOND: the dynamics of the atmospheric general circulation modeled on non-hydrostatic domains. *Prog. Earth Planet. Sci.* **6**, 61 (2019).
17. Wedi, N.P. et al. A baseline for global weather and climate simulations at 1 km resolution. *J. Adv. Model. Earth Syst.* **12**, e2020MS002192 (2020).
18. Schulthess, T. C. et al. Reflecting on the goal and baseline for exascale computing: a roadmap based on weather and climate simulations. *Comput. Sci. Eng.* **21**, 30–41 (2018).
19. Bauer, P., Stevens, B., Hazeleger, W. A digital twin of Earth for the green transition.*Nat. Clim. Change* https://doi.org/10.1038/s41558-021-00986-y (2021).
20. Davis, N. What is the fourth industrial revolution? *World Economic Forum* https://www.weforum.org/agenda/2016/01/what-is-the-fourth-industrial-revolution/ (19 January 2016).
21. Tao, F. & Qi, Q. Make more digital twins. *Nature* **573**, 490–491 (2019).
22. Bell, G. *Supercomputers: The Amazing Race (A History of Supercomputing, 1960–2020)* (2014).
23. Lynch, P. J. The origins of computer weather prediction and climate modeling. *Comput. Phys.* **227**, 3431–3444 (2008).
24. Bondyopadhyay, P. K. Moore's law governs the silicon revolution. *Proc. IEEE* **86**, 78–81 (1998).
25. Frank, D. J. et al. Device scaling limits of Si MOSFETs and their application dependencies. *Proc. IEEE* **89**, 259–288 (2001).
26. Easterbrook, S. M. & Johns, T. C. Engineering the software for understanding climate change. *Comput. Sci. Eng.* **11**, 65–74 (2009).
27. Fuhrer, O. et al. Near-global climate simulation at 1 km resolution: establishing a performance baseline on 4888 GPUs with COSMO 5.0. *Geosci. Model Dev.* **11**, 1665–1681 (2018).
28. Lawrence, B. N. et al. Crossing the chasm: how to develop weather and climate models for next generation computers. *Geosci. Model Dev.* **11**, 1799–1821 (2018).
29. Williamson, D. L. The evolution of dynamical cores for global atmospheric models. *J. Meteorol. Soc. Jpn Ser. II* **85**, 241–269 (2007).
30. McFarlane, N. Parameterizations: representing key processes in climate models without resolving them. *Wiley Interdiscip. Rev. Clim. Change* **2**, 482–497 (2011).
31. Flato, G. M. Earth system models: an overview. *Wiley Interdiscip. Rev. Clim. Change* **2**, 783–800 (2011).
32. Steppeler, J., Hess, R., Schättler, U. & Bonaventura, L. Review of numerical methods for nonhydrostatic weather prediction models. *Meteorol. Atmos. Phys.* **82**, 287–301 (2003).
33. Mengaldo, G. et al. Current and emerging time-integration strategies in global numerical weather and climate prediction. *Arch. Comput. Meth. Eng.* **26**, 663–684 (2019).
34. Teixeira, J., Reynolds, C. A. & Judd, K. Time step sensitivity of nonlinear atmospheric models: numerical convergence, truncation error growth, and ensemble design. *J. Atmos. Sci.* **64**, 175–189 (2007).
35. Dueben, P. D. & Palmer, T. Benchmark tests for numerical weather forecasts on inexact hardware. *Mon. Weather Rev.* **142**, 3809–3829 (2014).
36. Vána, F. et al. Single precision in weather forecasting models: an evaluation with the IFS. *Mon. Weather Rev.* **145**, 495–502 (2017).
37. Hatfield, S. et al. Choosing the optimal numerical precision for data assimilation in the presence of model error. *J. Adv. Model. Earth Syst.* **10**, 2177–2191 (2018).
38. Dueben, P. D. & Dawson, A. An approach to secure weather and climate models against hardware faults. *J. Adv. Model. Earth Syst.* **9**, 501–513 (2017).
39. Balaji, V., Benson, R., Wyman, B. & Held, I. Coarse-grained component concurrency in Earth system modeling: parallelizing atmospheric radiative transfer in the GFDL AM3 model using the flexible modeling system coupling framework. *Geosci. Model Dev.* **9**, 3605–3616 (2016).
40. Koldunov, N. V. et al. Scalability and some optimization of the Finite-volumE Sea ice–Ocean Model, Version 2.0 (FESOM2). *Geosci. Model Dev.* **12**, 3991–4012 (2019).
41. Mozdzynski, G., Hamrud, M., Wedi, N., Doleschal, J. & Richardson, H. *2012 SC Companion: High Performance Computing, Networking Storage and Analysis* 652–661 (2012).
42. Sanan, P., Schnepp, S. M. & May, D. A. Pipelined, flexible Krylov subspace methods. *SIAM J. Sci. Comput.* **38**, C441–C470 (2016).
43. Maisonnave, E. et al. *CDI-pio & XIOS I/O Servers Compatibility with HR Climate Models* TR/CMGC/17/52 (CERFACS, 2017).
44. Govett, M. W., Middlecoff, J. & Henderson, T. Running the NIM next-generation weather model on GPUs. In *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing* 792–796 (2010).
45. Thaler, F. et al. Porting the cosmo weather model to manycore CPUS. In *Proc. Platform for Advanced Scientific Computing Conference* 1–11 (2019).
46. Alexander, F. et al. Exascale applications: skin in the game. *Phil. Trans. R. Soc. A* **378**, 20190056 (2020).
47. Zhang, S. et al. Optimizing high-resolution community Earth system model on a heterogeneous many-core supercomputing platform. *Geosci. Model Dev.* **13**, 4809–4829 (2020).
48. Melvin, T. et al. A mixed finite-element, finite-volume, semi-implicit discretization for atmospheric dynamics: Cartesian geometry. *Q. J. Royal Meteorol. Soc.* **145**, 2835–2853 (2019).

49. Adams, S. V. et al. LFRic: Meeting the challenges of scalability and performance portability in weather and climate models. *J. Parallel Distrib. Comput.* **132**, 383–396 (2019).
50. Satoh, M. et al. Nonhydrostatic icosahedral atmospheric model (NICAM) for global cloud resolving simulations. *J. Comput. Phys.* **227**, 3486–3514 (2008).
51. Miyoshi, T., Kondo, K. & Imamura, T. The 10,240-member ensemble Kalman filtering with an intermediate AGCM. *Geophys. Res. Lett.* **41**, 5264–5271 (2014).
52. Washington, W. M., Buja, L. & Craig, A. The computational future for climate and Earth system models: on the path to petaflop and beyond. *Phil. Trans. R. Soc. A* **367**, 833–846 (2009).
53. Reichstein, M. et al. Deep learning and process understanding for data-driven Earth system science. *Nature* **566**, 195–204 (2019).
54. Balaji, V. et al. CPMIP: measurements of real computational performance of Earth system models in CMIP6. *Geosci. Model Dev.* **10**, 19–34 (2017).
55. Tumolo, G. & Bonaventura, L. A semi-implicit, semi-Lagrangian discontinuous Galerkin framework for adaptive numerical weather prediction. *Q. J. R. Meteorol. Soc.* **141**, 2582–2601 (2015).
56. Kühnlein, C. et al. FVM 1.0: a nonhydrostatic finite-volume dynamical core for the IFS. *Geosci. Model Dev.* **12**, 651–676 (2019).
57. Nastrom, G., Gage, K. S. & Jasperson, W. Kinetic energy spectrum of large-and mesoscale atmospheric processes. *Nature* **310**, 36–38 (1984).
58. Gander, M. J. in *Multiple Shooting and Time Domain Decomposition Methods* 69–113 (Springer, 2015).
59. Hamon, F. P., Schreiber, M. & Minion, M. L. Parallel-in-time multi-level integration of the shallow-water equations on the rotating sphere. *J. Comput. Phys.* **407**, 109210 (2020).
60. Fisher, M. & Gürol, S. Parallelization in the time dimension of four-dimensional variational data assimilation. *Q. J. R. Meteorol. Soc.* **143**, 1136–1147 (2017).
61. Duran, A. et al. OmpSs: a proposal for programming heterogeneous multi-core architectures. *Parallel Proc. Lett.* **21**, 173–193 (2011).
62. Weiland, M., Jackson, A., Johnson, N. & Parsons, M. Exploiting the performance benefits of storage class memory for HPC and HPDA workflows. *Supercomput. Front. Innov.* **5**, 79–94 (2018).
63. Müller, A. et al. The ESCAPE project: energy-efficient scalable algorithms for weather prediction at exascale. *Geosci. Model Dev.* **12**, 4425–4441 (2019).
64. Heroux, M. et al. *Improving Performance via Mini-Applications* SAND2009-5574 (Sandia, 2009).
65. Yang, C. et al. 10M-core scalable fully-implicit solver for nonhydrostatic atmospheric dynamics. In *SC16: Proc. International Conference for High Performance Computing, Networking, Storage and Analysis* 57–68 (2016).
66. Mozdzynski, G., Hamrud, M. & Wedi, N. A partitioned global address space implementation of the European Centre for Medium Range Weather Forecasts Integrated Forecasting System. *Int. J. High Perform. Comput. Appl.* **29**, 261–273 (2015).
67. Schulthess, T. C. Programming revisited. *Nat. Phys.* **11**, 369–373 (2015).
68. Deconinck, W. et al. Atlas: a library for numerical weather prediction and climate modelling. *Comput. Phys. Comm.* **220**, 188–204 (2017).
69. Trèmolet, Y. The Joint Effort for Data Assimilation Integration (JEDI). *JCSDA Q.* **66**, 1–5 (2020).
70. Hill, C., DeLuca, C., Balaji, V., Suarez, M. & da Silva, A. The architecture of the Earth system modeling framework. *Comput. Sci. Eng.* **6**, 18–28 (2004).
71. Smart, S., Quintino, T. & Raoult, B. A high-performance distributed object-store for exascale numerical weather prediction and climate. In *Proc. Platform for Advanced Scientific Computing Conference* 1–11 (2019).
72. Bertagna, L. et al. HOMMEXX 1.0: a performance-portable atmospheric dynamical core for the energy exascale Earth system model. *Geosci. Model Dev.* **12**, 1423–1441 (2019).
73. Edwards, H. C. & Sunderland, D. Kokkos array performance-portable manycore programming model. In *Proc. 2012 International Workshop on Programming Models and Applications for Multicores and Manycores* 1–10 (2012).
74. Gysi, T., Osuna, C., Fuhrer, O., Bianco, M. & Schulthess, T. C. STELLA: a domain-specific tool for structured grid methods in weather and climate models. In *Proc. International Conference for High Performance Computing, Networking, Storage and Analysis* 1–12 (2015).
75. Sønderby, C. K. et al. MetNet: a neural weather model for precipitation forecasting. Preprint at https://arxiv.org/abs/2003.12140 (2020).
76. Ham, Y.-G., Kim, J.-H. & Luo, J.-J. Deep learning for multi-year ENSO forecasts. *Nature* **573**, 568–572 (2019).
77. Rasp, S. et al. WeatherBench: a benchmark data set for data-driven weather forecasting. *J. Adv. Model. Earth Syst.* **12**, e2020MS002203, https://doi.org/10.1029/2020MS002203 (2020).
78. Prudden, R. et al. A review of radar-based nowcasting of precipitation and applicable machine learning techniques. Preprint at https://arxiv.org/abs/2005.04988 (2020).
79. Bonavita, M., Laloyaux, P. Machine learning for model error inference and correction. *Earth Space Sci. Open Arch*. https://doi.org/10.1002/essoar.10503695.1 (2020).
80. Brajard, J., Carassi, A., Bocquet, M. & Bertino, L. Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: a case study with the Lorenz 96 model. *J. Comput. Sci.* **44**, 101171 (2020).
81. Chevallier, F., Morcrette, J.-J., Chéruy, F. & Scott, N. Use of a neural-network-based long-wave radiative-transfer scheme in the ECMWF atmospheric model. *Q. J. R. Meteorol. Soc.* **126**, 761–776 (2000).
82. Krasnopolsky, V. M., Fox-Rabinovitz, M. S. & Chalikov, D. V. New approach to calculation of atmospheric model physics: accurate and fast neural network emulation of longwave radiation in a climate model. *Mon. Weather Rev.* **133**, 1370–1383 (2005).
83. Schneider, T., Lan, S., Stuart, A. & Teixeira, J. Middle atmosphere dynamical sources of the semiannual oscillation in the thermosphere and ionosphere. *Geophys. Res. Lett.* **44**, 12–21 (2017).
84. Kurth, T. et al. Exascale deep learning for climate analytics. In *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis* 649–660 (2018).
85. Vandal, T. et al. DeepSD: Generating high resolution climate change projections through single image super-resolution. In *Proc. 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 1663–1672 (2017).
86. Rasp, S. & Lerch, S. Neural networks for postprocessing ensemble weather forecasts. *Mon. Weather Rev.* **146**, 3885–3900 (2018).
87. Grönquist, P. et al. Deep learning for post-processing ensemble weather forecasts. Preprint at https://arxiv.org/abs/2005.08748 (2020)
88. Chui, M. Notes from the AI frontier: Applications and value of deep learning. *McKinsey* https://www.mckinsey.com/featured-insights/artificial-intelligence/notes-from-the-ai-frontier-applications-and-value-of-deep-learning (17 April 2018).
89. Black, D. HPC Market Update from Hyperion Research. *insideHPC* https://insidehpc.com/2019/09/hpc-market-update-from-hyperion-research (2019).
90. Klöwer, M., Dueben, P.D., Palmer, T.N. *J. Adv. Model. Earth Syst*., e2020MS002246, https://doi.org/10.1029/2020MS002246 (10 September 2020).
91. Brenowitz, N. D. & Bretherton, C. S. Bretherton, prognostic validation of a neural network unified physics parameterization. *Geophys. Res. Lett.* **45**, 6289–6298 (2018).
92. Rasp, S., Pritchard, M. S. & Gentine, P. Deep learning to represent subgrid processes in climate models. *Proc. Natl Acad. Sci. USA* **115**, 9684–9689 (2018).
93. Gysi, T. et al. Domain-specific multi-level IR rewriting for GPU. Preprint at https://arxiv.org/abs/2005.13014 (2020).
94. Ben-Nun, T., de Fine Licht, J., Ziogas, A. N., Schneider, T. & Hoefler, T. Stateful dataflow multigraphs: a data-centric model for performance portability on heterogeneous architectures. In *Proc. International Conference for High Performance Computing, Networking, Storage and Analysis* 1–14 (2019).
95. Hruska, J. As chip design costs skyrocket, 3nm process node is in jeopardy. *ExtremeTech* https://www.extremetech.com/computing/272096-3nm-process-node (22 June 2018).
96. Unat, D. et al. Trends in data locality abstractions for HPC systems. *IEEE Trans. Parallel Distrib. Syst.* **28**, 3007–3020 (2017).
97. Horowitz, M. 1.1 computing's energy problem (and what we can do about it). In *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* 10–14 (2014).
98. Gysi, T., Grosser, T., Hoefler, T. Modesto: Data-centric analytic optimization of complex stencil programs on heterogeneous architectures. In *Modesto: Proceedings of the 29th ACM on International Conference on Supercomputing* 177–186 (2015).
99. de Fine Licht, J. et al. StencilFlow: Mapping Large Stencil Programs to Distributed Spatial Computing Systems. Preprint at https://arxiv.org/abs/2010.15218 (2020)
100. Piz Daint. *Swiss National Supercomputing Centre* https://www.cscs.ch/computers/piz-daint (2020).
101. EuroHPC supercomputer systems. *European Commission* http://eurohpc.eu/systems (2020).
102. Girolamo, S. D., Schmid, P., Schulthess, T. & Hoefler, T. SimFS: a simulation data virtualizing file system interface. In *Proc. 33st IEEE International Parallel & Distributed Processing Symposium (IPDPS'19)* (2019).
103. Yang, C., Wu, H., Huang, Q., Li, Z. & Li, J. Using spatial principles to optimize distributed computing for enabling the physical science discoveries. *Proc. Natl Acad. Sci. USA* **108**, 5498–5503 (2011).
104. Jia, Z., Maggioni, M., Staiger, B. & Scarpazza, D. P. Dissecting the NVIDIA Volta GPU architecture via microbenchmarking. Preprint at https://arxiv.org/abs/1804.06826 (2018).
105. Tsai, Y. M., Cojean, T. & Anzt, H. Evaluating the performance of NVIDIA's A100 ampere GPU for sparse linear algebra computations. Preprint at https://arxiv.org/abs/2008.08478 (2020).

106. Voskuilen, G. R., Gimenez, A., Peng, I., Moore, S. & Gokhale, M. *Milestone M1 Report: HBM2/3 Evaluation on Many-core CPU WBS 2.4, Milestone ECP-MT-1000* SAND2018-6370R (SANDIA, 2018).
107. Buehner, M., Houtekamer, P., Charette, C., Mitchell, H. L. & He, B. Intercomparison of variational data assimilation and the ensemble Kalman filter for global deterministic NWP. Part I: description and single-observation experiments. *Mon. Weather Rev.* **138**, 1550–1566 (2010).
108. Blayo, É., Bocquet, M., Cosme, E. & Cugliandolo, L. F. *Advanced Data Assimilation for Geosciences: Lecture Notes of the Les Houches School of Physics* (Oxford Univ. Press, 2014).
109. Palmer, T. Short-term tests validate long-term estimates of climate change. *Nature* **582**, 185–186 (2020).
110. Voosen, P. Europe is building a 'digital twin' of Earth to revolutionize climate forecasts. *Science* https://doi.org/10.1126/science.abf0687 (1 October 2020)
111. Palmer, T., Stevens, B. & Bauer, P. We Need an International Center for Climate Modeling. *Scientific American* https://blogs.scientificamerican.com/observations/we-need-an-international-center-for-climate-modeling/ (18 December 2019)

Acknowledgements

The authors are grateful to P. Lopez for providing Fig. 2, M. Fielding and M. Janiskova for the illustrations of simulation-observation fusion in Box 1, and to the European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT) and NASA for providing the satellite data used to produce Fig. 2 and the figure in Box 1.

Author contributions

P.B. conceived and organized the manuscript. P.B., P.D., T.H., T.Q., T.S. and N.W. contributed to the writing and revision of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence should be addressed to P.B.

Peer review information *Nature Computational Science* thanks Jana Sillmann and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Fernando Chirigati was the primary editor on this Perspective and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.