An ExaScale Programming, Multi-objective Optimisation and Resilience Management Environment Based on Nested Recursive Parallelism

The AllScale project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 671603

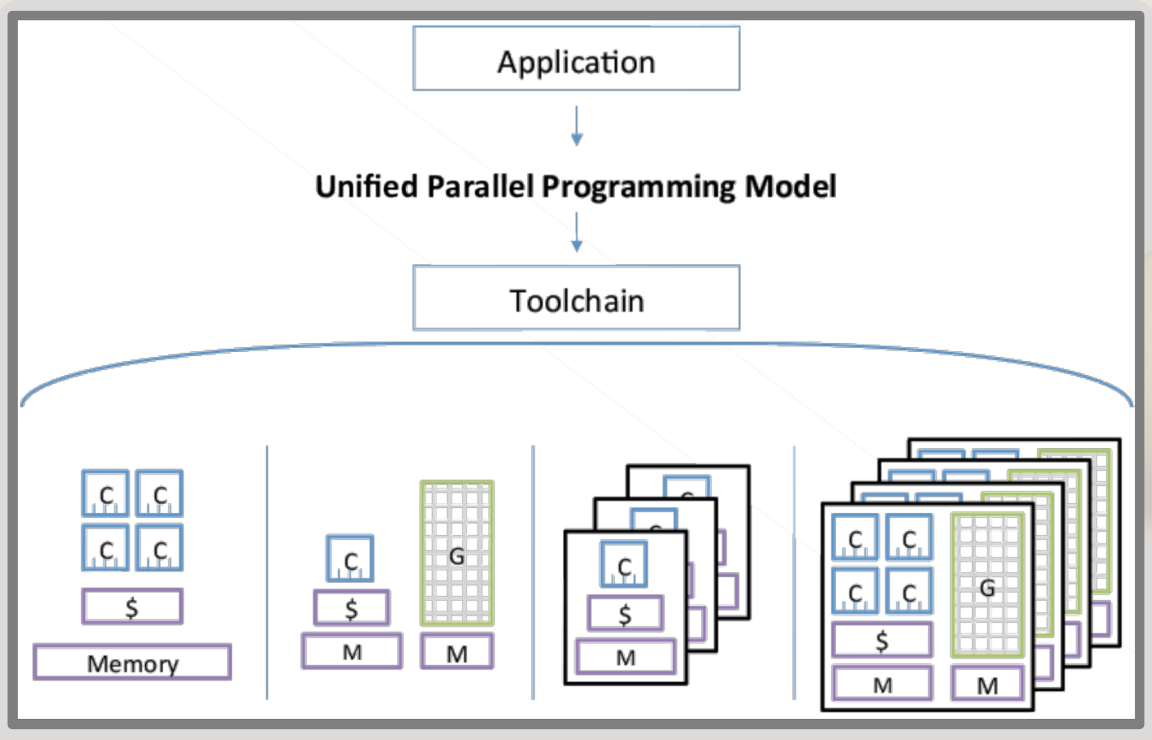
AllScale enables developers to be productive and to port applications to any scale of system

AllScale is an innovative programming approach for ExaScale, decoupling program specification parallelism from management tasks during runtime. The parallel programming model is based on the nested recursive parallelism, focusing on the following developments:

- Automated applications porting from small- to extreme scale architectures
- Flexible tuning of program execution to fulfil trade-offs among execution time, energy and resource usage,
- Efficient management of hardware resources and associated parameters (e.g. clock speed),
- Seamless Integration of resilience management measures to compensate for isolated hardware failures,
- Online performance monitoring and analysis.

The above mentioned developments will provide an application independent, reusable codes, taking advantage of combining sophisticated, modular and customizable compiler and runtime system based solutions.

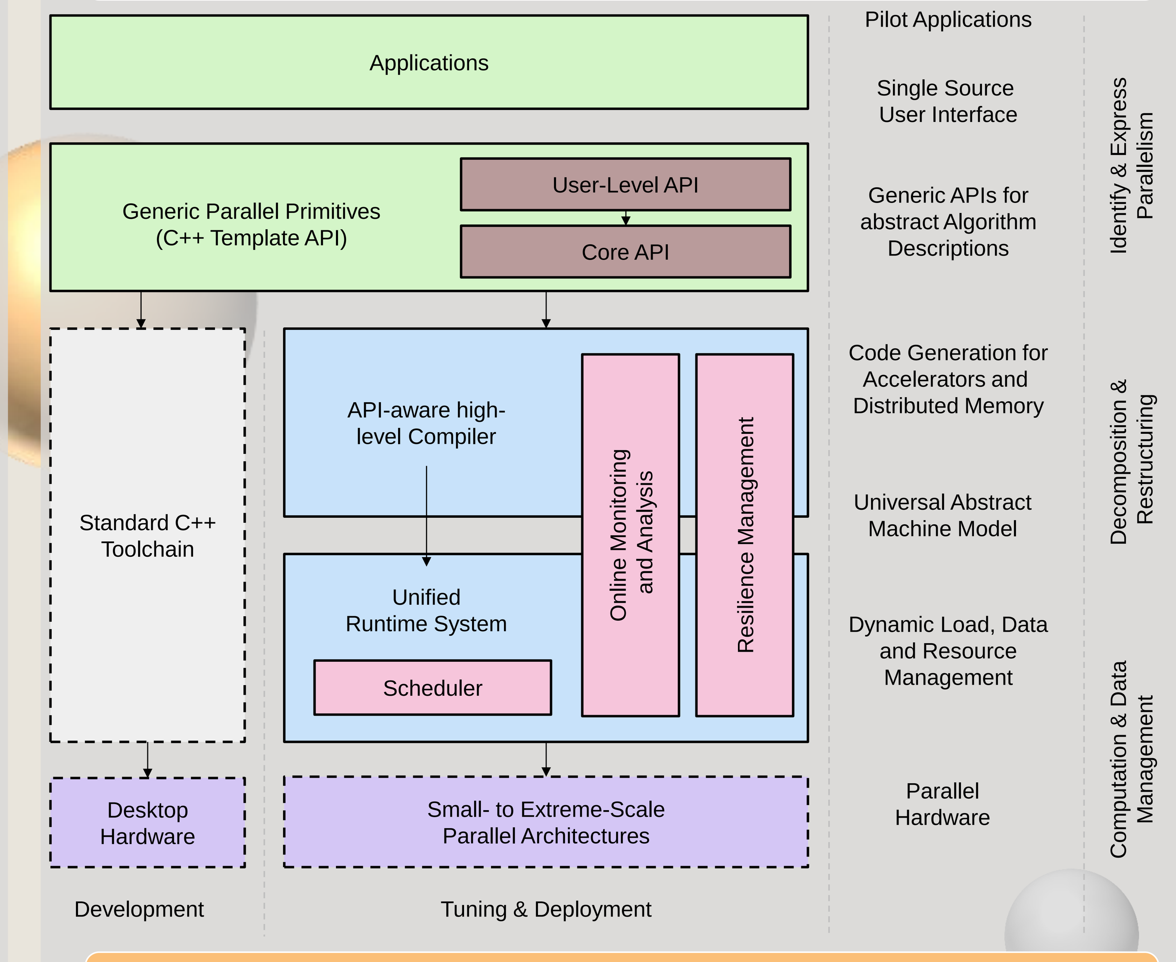
AllScale is expected to boost the parallel applications development productivity, their portability, and runtime efficiency. It will reduce energy needs, thus improving the resource efficiency utilisation of small to extreme scale parallel systems. The achieved outcomes are validated with applications from fluid dynamics, environmental hazard and space weather simulations; provided from SME, industry and academic consortium partners.



AllScale Objectives

- Single Source to Any Scale
- Facilitate Recursive Parallelism
- Multi-Objective Optimization for time, energy and resource usage
- Unified Runtime System
- Mitigating Hardware Failures
- Integrated Online/Offline Monitoring

AllScale Architecture



AllScale API based on C++ Templates

User-Level API

- High-level abstractions (e.g. grids, meshes, stencils, channels)
- Familiar interfaces (e.g. parallel for loops, map-reduce)

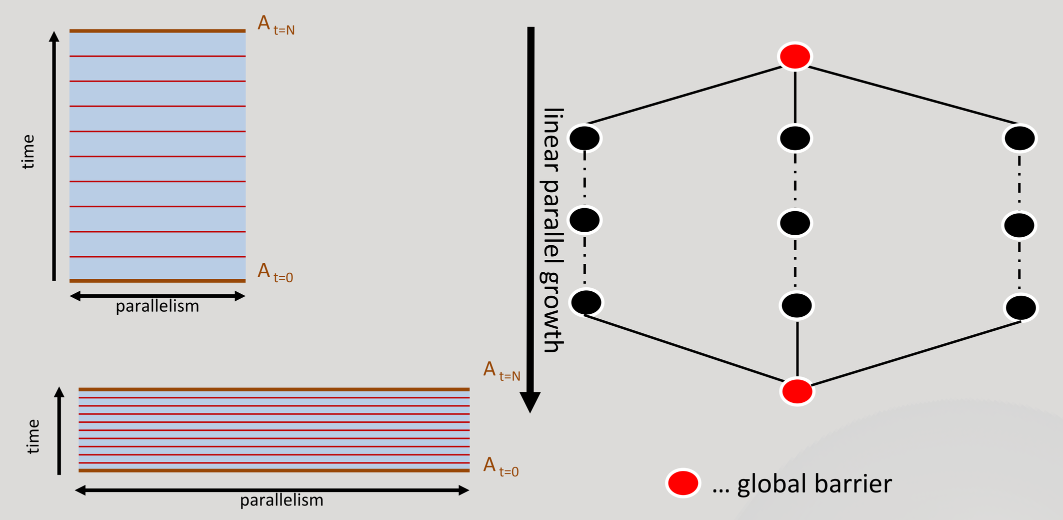
implemented based on

Core API

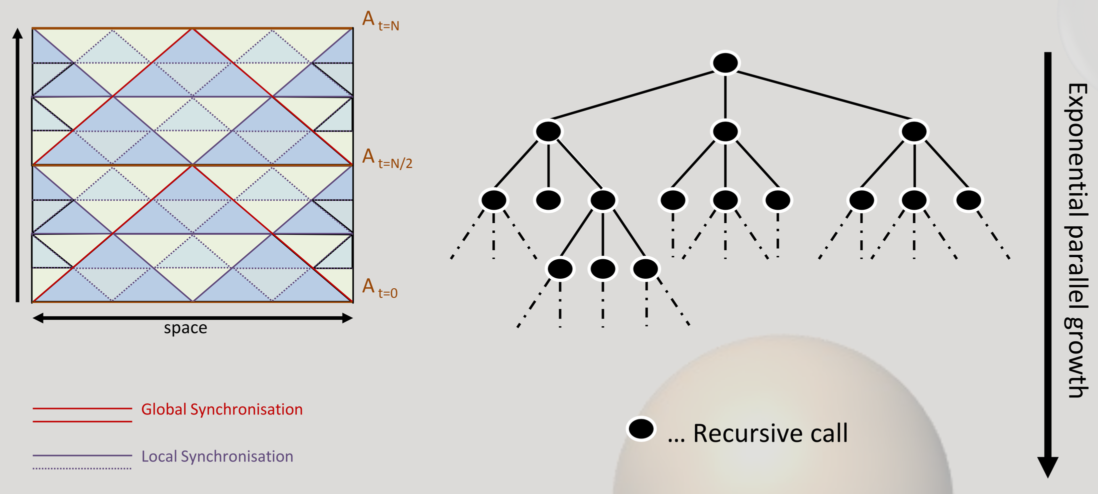
- Generic function template for recursive parallelism
- Set of recursive data structure templates
- Synchronization, control- and data-flow primitives

```
prec(a,b,c)   =>

future fun( auto in ) {
    return spawn
     seq_f(in).get() or
     par_f(in).get()
}

future par_f( auto in ) {
    if ( a(in) ) return spawn b(in) ;
    return spawn c( in, fun ) );
}

value seq_f( auto in ) {
    if ( a(in) ) return value( b(in) );
    return value( c( in, seq_f ) );
}
```

Conventional Flat Parallelism

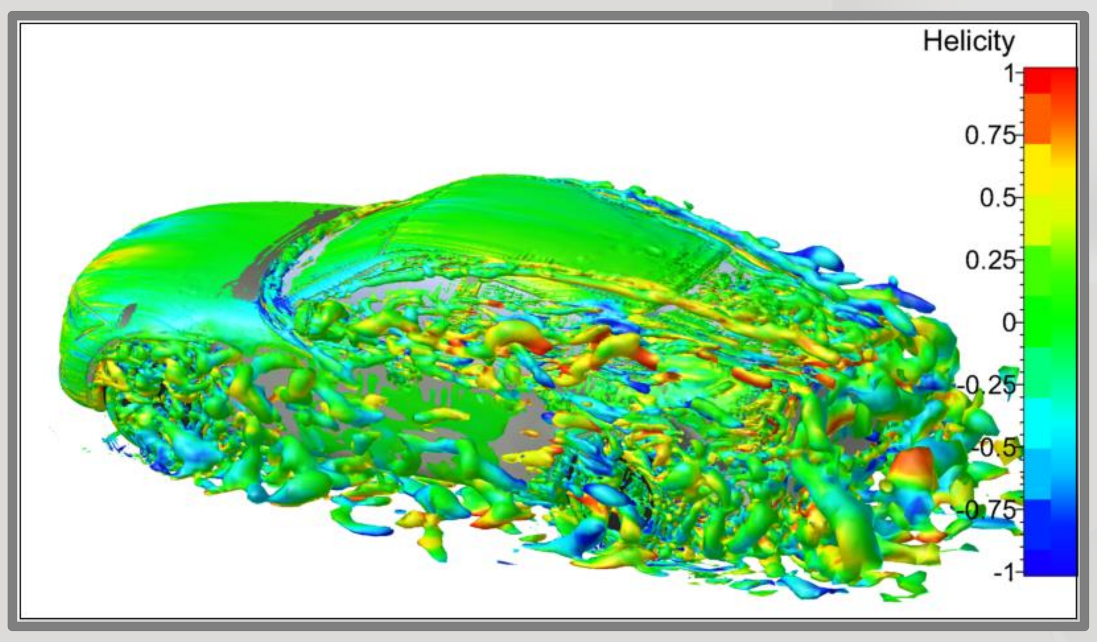


Recursively Nested Parallelism



FINE/Open™: Towards ExaScale HPC Industrial Unsteady CFD simulations (NUMECA)

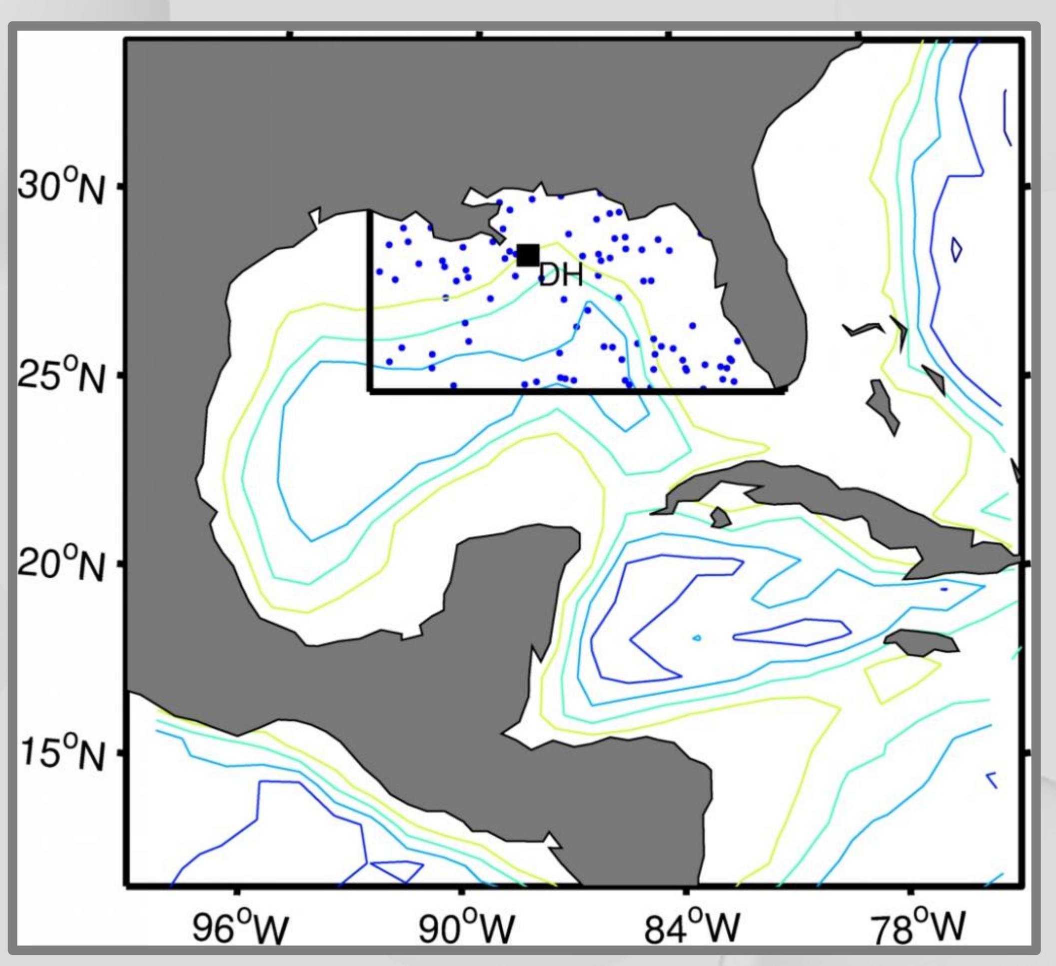
The request from industry towards higher fidelity CFD and increased simulation reliability is more and more pressing. The use of unsteady **LES** (Large Eddy Simulation) or **DNS** (Direct Numerical Simulation) provides an appropriate approach to fulfil such imposed industry demand. However, when applied to industrial cases on complex geometries, such LES/DNS simulation implies very large meshes, **ranging from 1 to 10 billion mesh points, yielding problems of 50-100 billions+ degrees of freedom as well as thousands of unsteady time steps**.



To fulfil those requirements, the cost, in terms of CPU time, needs to be significantly reduced, and the best way to do it, is to increase the number of processors, towards ExaScale HPC (High Performance Computing). The current distributed parallel implementation in FINE/Open™ is based on MPI, allowing reaching reasonable scalability up to a few thousand cores. The partitioning and agglomeration of the mesh are done in parallel, as well as the solution output. In the current implementation, when attempting to increase the number of processors by 1 or 2 orders of magnitude, **severe scalability problems** are encountered. To answer these bottlenecks, the AllScale Environment, whose research focuses towards **massively parallel simulations**, is being applied.

AMDADOS: Adaptive Meshing and Data Assimilation for the Deepwater Horizon Oil Spill (IBM)

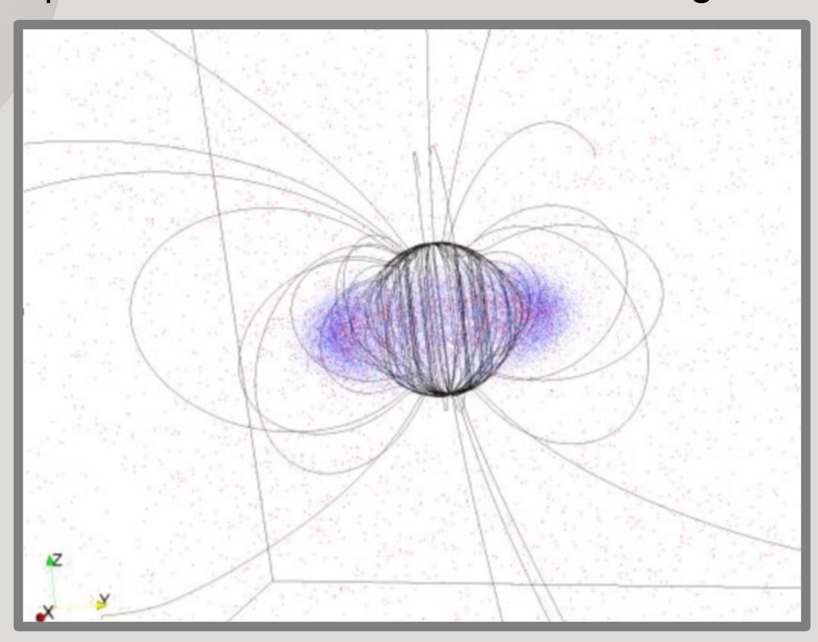
The Deepwater Horizon oil spill is the largest accidental spill in the history of the petroleum industry. The **BP blow-out lasted for 87 days, releasing approximately 4.9 million barrels (780,000 m3)** into the surrounding environment. Authorities collected huge volumes of data concerning the extent and evolution of the oil spill. While previous research has made use of some of the data, a system that harnesses the full potential of the dataset by integrating it with a set of highly accurate, adaptive models and meta-models is yet to be put in place.



In AMDADOS, Data Assimilation (DA) and Adaptive Meshing (AM) are used jointly for simulating the Deepwater Horizon accident. The system autonomously increases with AM the resolution at targeted locations, while DA incorporates observations into the model forecast. The HPC requirements and simulated data make AMDADOS an ExaScale application.

iPIC3D: implicit Particle-In-Cell 3D application for Space Weather Simulation (KTH)

Space Weather is the study of the processes originating on the Sun and propagating through the Solar System, which affects people and technology in Space and on the Earth. KTH has implemented the massively parallel Particle-in-Cell code iPIC3D, which simulates the plasma particles interaction with the **Earth's magnetic field**.



Some highly energetic particles are trapped within the Van Allen radiation belts while others escape the confinement of the Earth's magnetic field. This leads to large load imbalances since most of the particles are concentrated close to the Earth (violet cloud in the figure).

The current implementation of iPIC3D has considerable load imbalance problems. The redesign of iPIC3D will not only allow leveraging the nested recursive parallelism, but will also **reduce the total amount of global communication and synchronization by 80%**, which will lead to a much better scaling behaviour for iPIC3D on massively parallel computing systems. The AllScale Environment will enable reduction of either energy or resource consumption, by at least 25%, while not causing an execution penalty of more than 10%. Finally, the AllScale iPIC3D will allow us to perform the very first PIC simulation of radiation belts formation, on a grid with 5,000x5,000x5,000 cells with at least 10,000 particles per cell, requiring **10^{22}-10^{23} FLOPS or 27-hour-simulation** on an ExaScale super-computer.

General Coordinator :
Thomas Fahringer tf@dps.uibk.ac.at

Scientific Coordinator :
Herbert Jordan herbert@dps.uibk.ac.at

WebSite
http://www.allscale.eu