



Soleil-X : Turbulence, Particles, and Radiation in the Regent Programming Language

Hilario Torres, Manolis Papadakis, Lluís Jofre, Wonchan Lee,
Alex Aiken, Gianluca Iaccarino

Stanford University

November 17, 2019



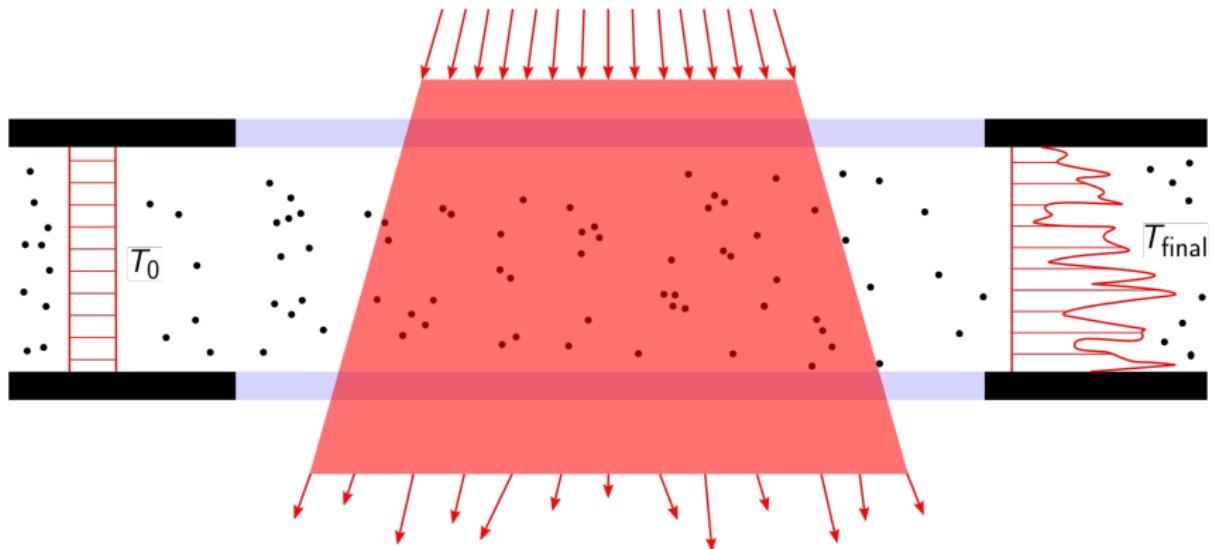
Overview

1. The PSAAP Program at Stanford
2. Task Based Parallelism
3. Soleil X - A Task Based Multiphysics Solver



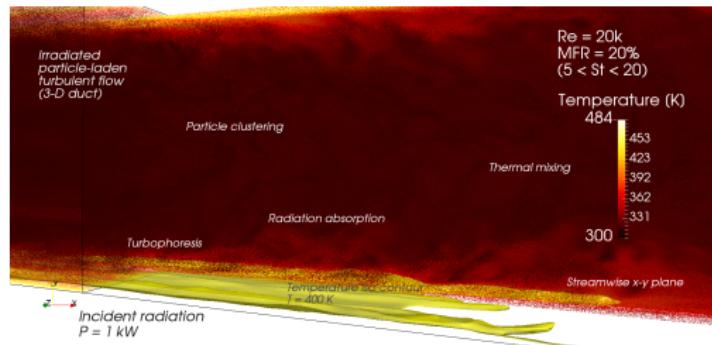
<https://ths.rwth-aachen.de/wp-content/uploads/sites/4/ge0masolar.jpg>

Particle Laden Turbulence In a Radiation Environment



PSAAP Project

Mission: To develop and demonstrate predictive multi-physics simulations of novel particle-based solar receivers on next-generation exascale compute systems

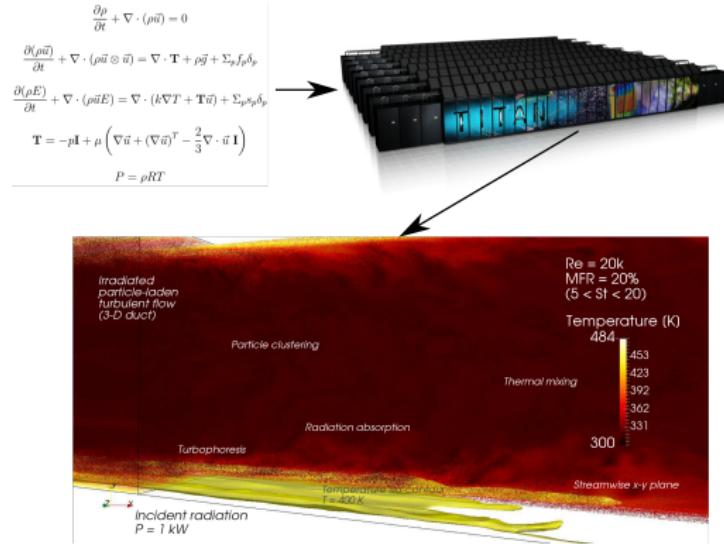


Research Components

- Physical Modeling
- Computational Methods
- Parallelism & Scalability
- Programming Environment
- Verification & Validation
- Uncertainty Quantification

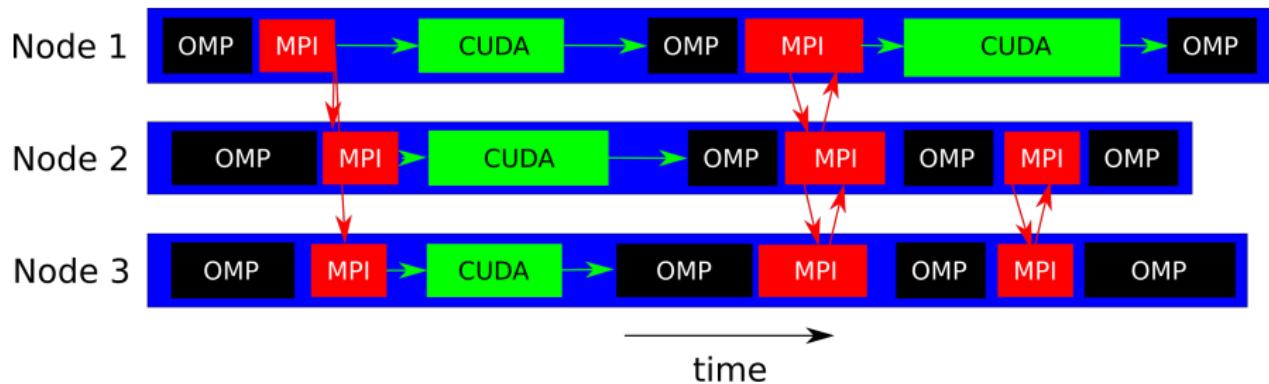
Computational Challenge of Particle Laden Turbulence In A Radiation Environment

- Predictive simulations of multi-scale multi-physics problems are expensive
 - Target large scale heterogeneous machines, Petascale to Exascale



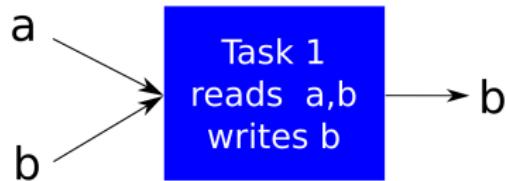
Computational Challenge of Multi-Physics HPC

- HPC applications usually rely on:
 - several different programming systems for parallelism (MPI, Pthreads, OpenMP, CUDA, etc...)
 - programmer coordinated communication in complex memory hierarchy
 - machine dependent parallelization strategies to maintain performance



Task Based Parallelism

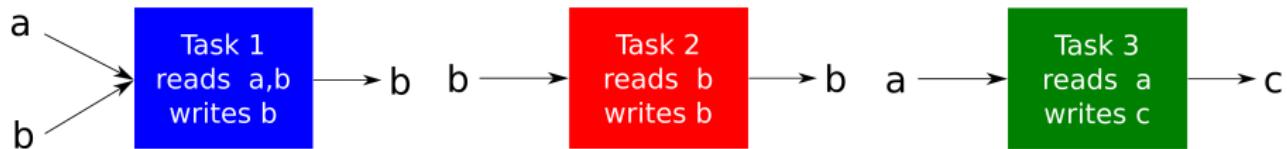
- Legion
 - A Data-Centric Programming System
 - Understands program structure and required data movement.
- Regent = High Level Legion
- Computational kernels are called “tasks”
 - Task arguments read and write permissions are specified by the programmer



- The main task in regent looks sequential
 - The parallelism is determined by data dependencies between tasks.
 - The task graph is built and executed at run time

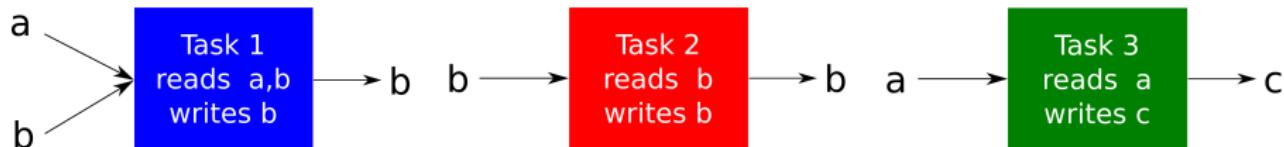
Determining Parallelism via Task Data Dependencies

- Task arguments read and write permissions are specified by the programmer

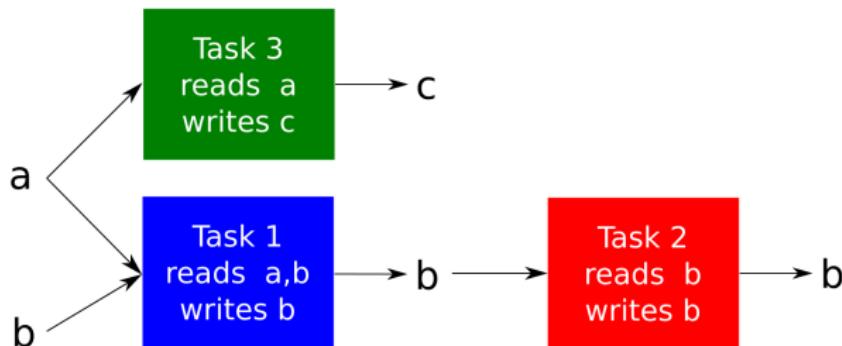


Determining Parallelism via Task Data Dependencies

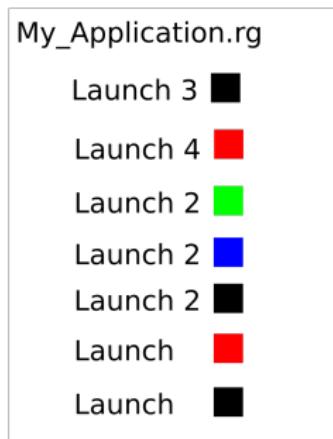
- Task arguments read and write permissions are specified by the programmer



- The parallelism is determined by data dependencies between tasks.



Building and Executing the Task Graph at Run Time



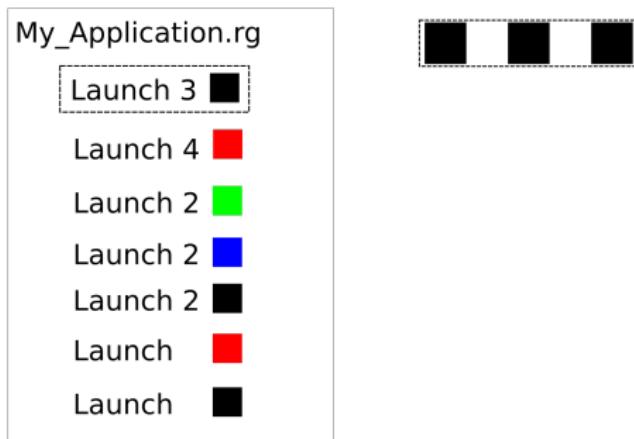
Node 1

Node 2

Node 3

→
time

Building and Executing the Task Graph at Run Time



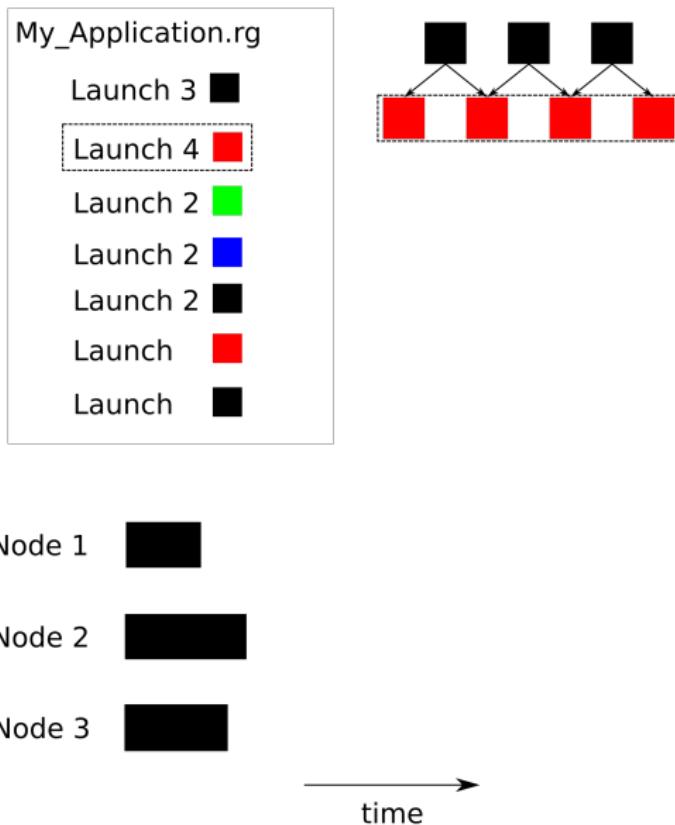
Node 1

Node 2

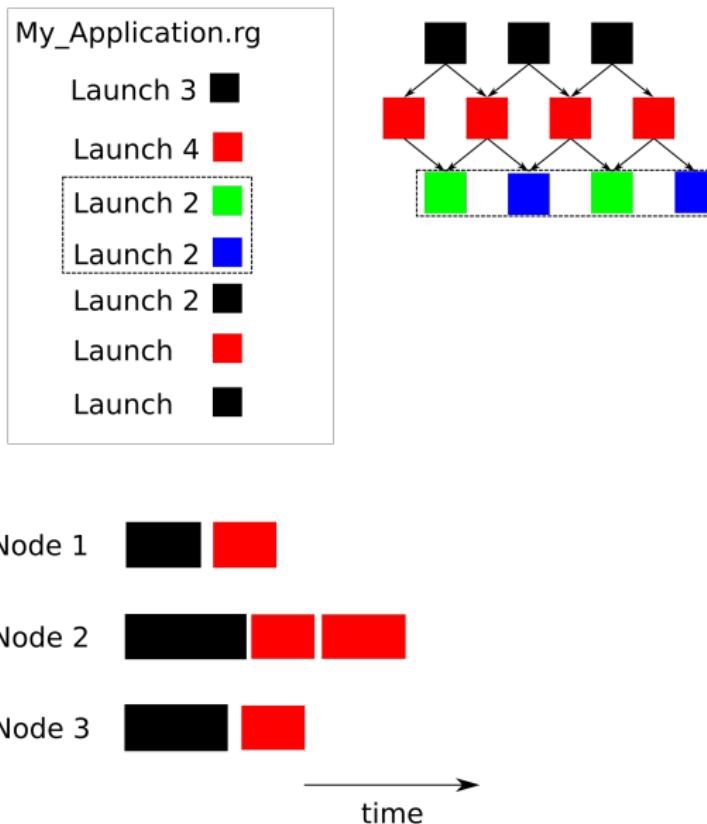
Node 3

→
time

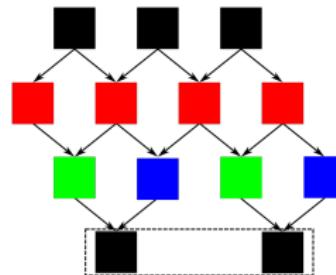
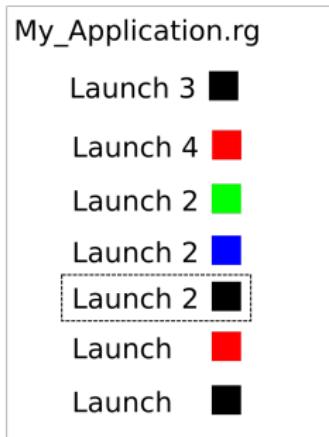
Building and Executing the Task Graph at Run Time



Building and Executing the Task Graph at Run Time

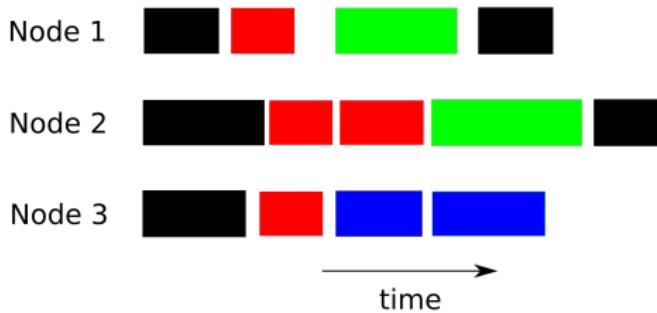
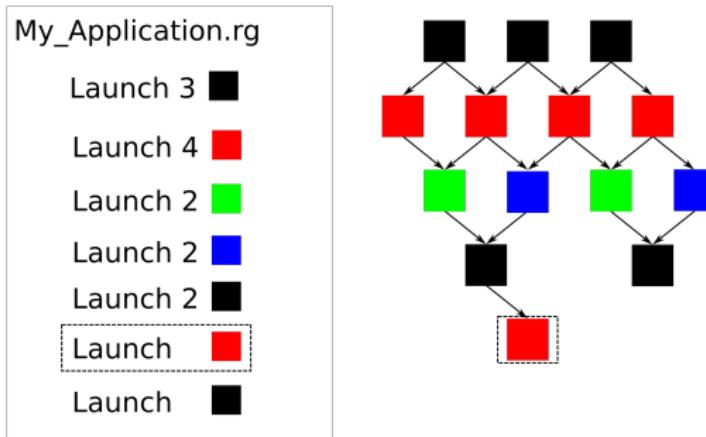


Building and Executing the Task Graph at Run Time

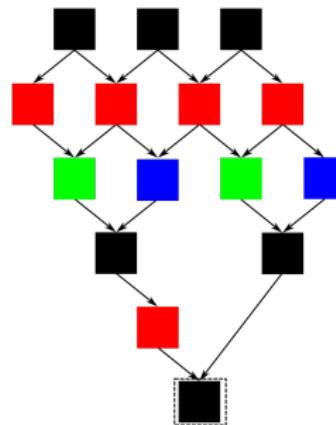
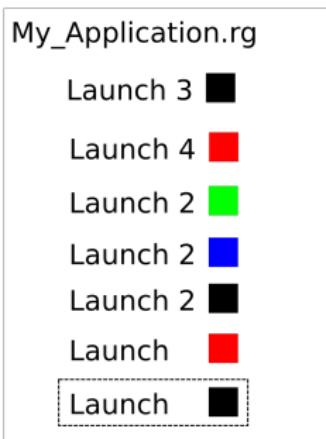


→
time

Building and Executing the Task Graph at Run Time

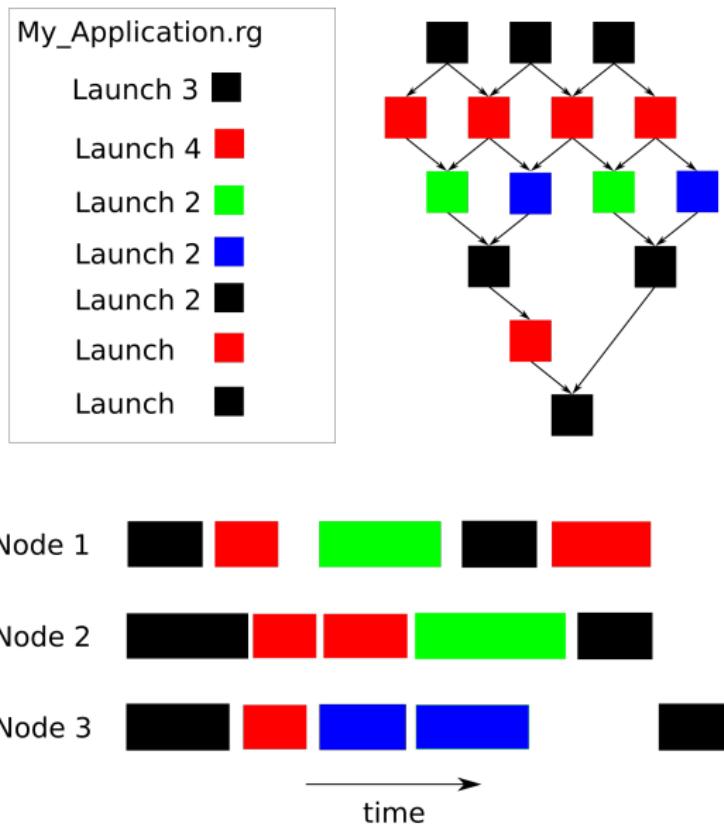


Building and Executing the Task Graph at Run Time



→
time

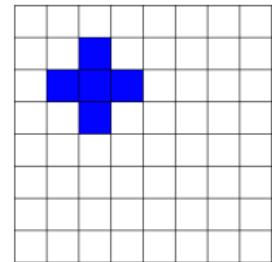
Building and Executing the Task Graph at Run Time



Soleil X - A Task Based Multi-Physics Solver

- Flow

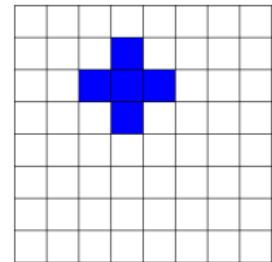
- Compressible Navier Stokes
- 2nd order finite difference
- Cell centered, flux based
- 3D cartesian grids



Soleil X - A Task Based Multi-Physics Solver

- Flow

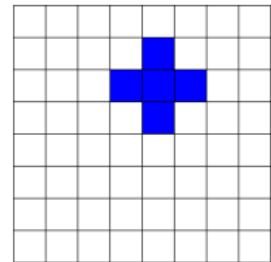
- Compressible Navier Stokes
- 2nd order finite difference
- Cell centered, flux based
- 3D cartesian grids



Soleil X - A Task Based Multi-Physics Solver

- Flow

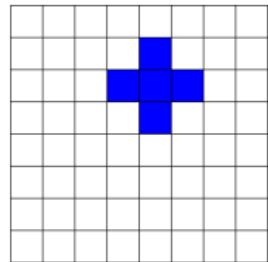
- Compressible Navier Stokes
- 2nd order finite difference
- Cell centered, flux based
- 3D cartesian grids



Soleil X - A Task Based Multi-Physics Solver

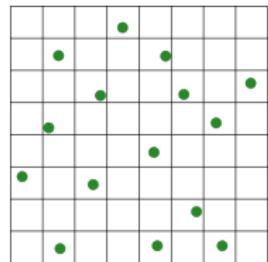
- Flow

- Compressible Navier Stokes
- 2nd order finite difference
- Cell centered, flux based
- 3D cartesian grids



- Particles

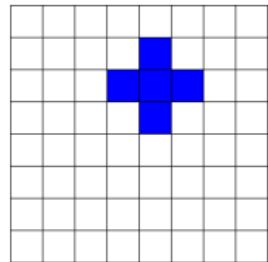
- Lagrangian point particles
- Particle-Fluid coupling in momentum and energy
- Particle-Particle collisions



Soleil X - A Task Based Multi-Physics Solver

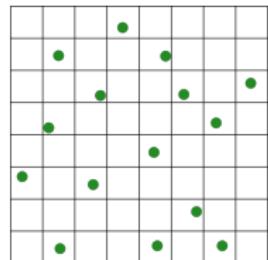
- Flow

- Compressible Navier Stokes
- 2nd order finite difference
- Cell centered, flux based
- 3D cartesian grids



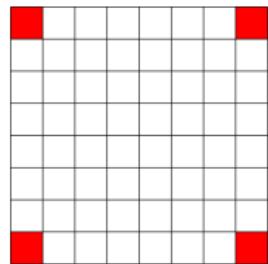
- Particles

- Lagrangian point particles
- Particle-Fluid coupling in momentum and energy
- Particle-Particle collisions



- Radiation

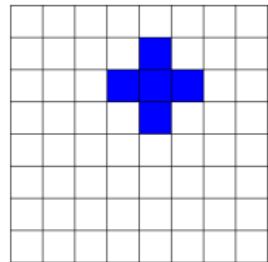
- Optically thin model
- Discrete Ordinates Method (sweeps)



Soleil X - A Task Based Multi-Physics Solver

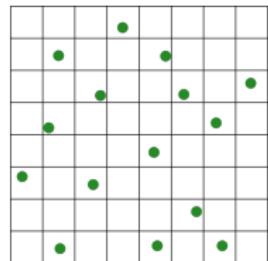
- Flow

- Compressible Navier Stokes
- 2nd order finite difference
- Cell centered, flux based
- 3D cartesian grids



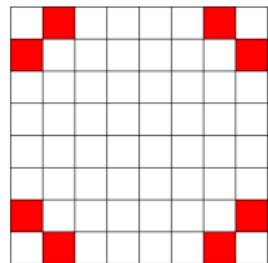
- Particles

- Lagrangian point particles
- Particle-Fluid coupling in momentum and energy
- Particle-Particle collisions



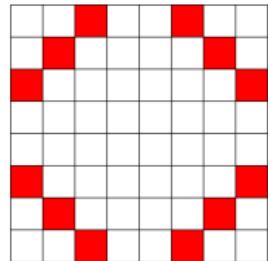
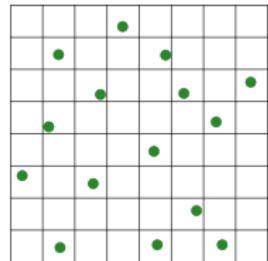
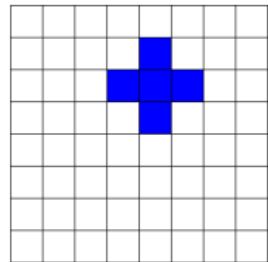
- Radiation

- Optically thin model
- Discrete Ordinates Method (sweeps)



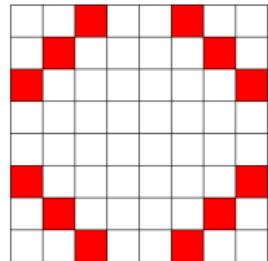
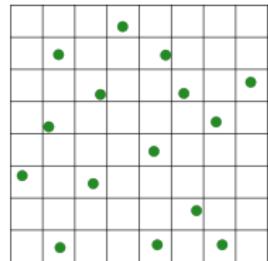
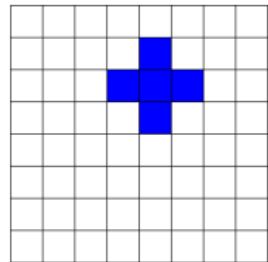
Soleil X - A Task Based Multi-Physics Solver

- Flow
 - Compressible Navier Stokes
 - 2nd order finite difference
 - Cell centered, flux based
 - 3D cartesian grids
- Particles
 - Lagrangian point particles
 - Particle-Fluid coupling in momentum and energy
 - Particle-Particle collisions
- Radiation
 - Optically thin model
 - Discrete Ordinates Method (sweeps)

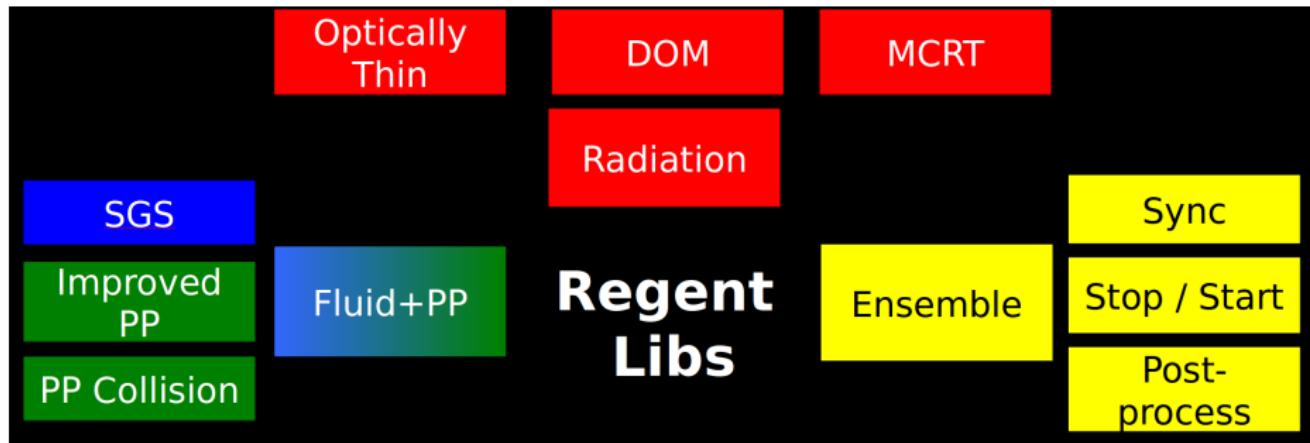


Soleil X - A Task Based Multi-Physics Solver

- Flow
 - Compressible Navier Stokes
 - 2nd order finite difference
 - Cell centered, flux based
 - 3D cartesian grids
- Particles
 - Lagrangian point particles
 - Particle-Fluid coupling in momentum and energy
 - Particle-Particle collisions
- Radiation
 - Optically thin model
 - Discrete Ordinates Method (sweeps)
- Time Integration
 - RK4 Time Advancement



Simulation Framework



Regent (Terra/Lua)

Legion

Mapper

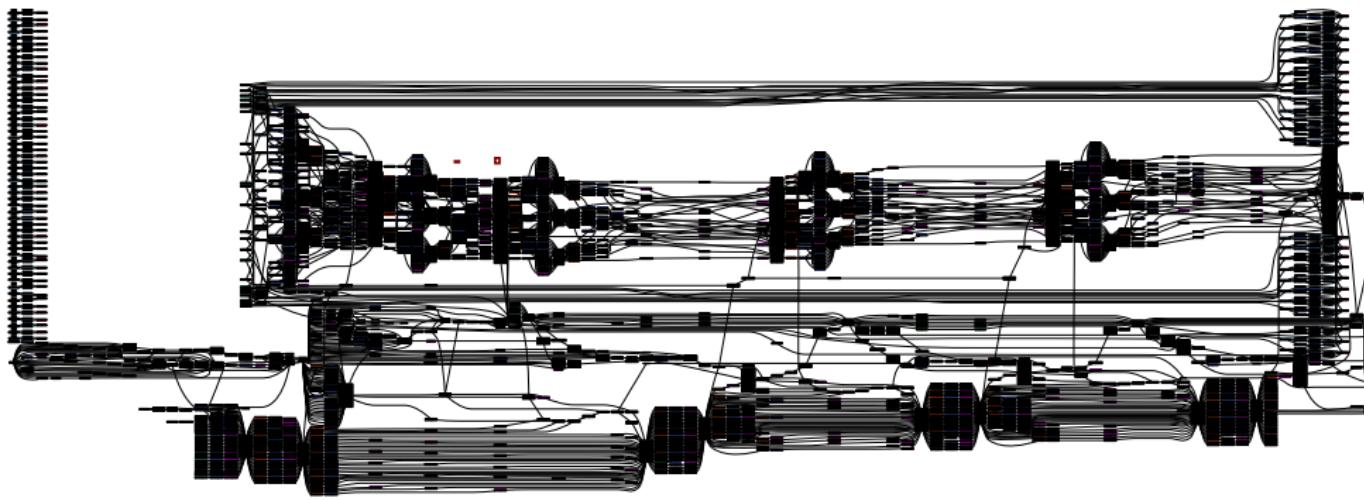
Task Example

```
task ComputeKineticEnergy(cells : region(ispace(int3d),Cell))
where
    reads (cells.Rho, cells.Velocity)
    writes(cells.Kinetic_Energy)
do
    for cell in cells do
        cell.Kinetic_Energy = 0.5 * cell.Rho * cell.Velocity^2.0
    end
end
```

GPU Task Example

```
--demand(__cuda)
task ComputeKineticEnergy(cells : region(ispace(int3d),Cell))
where
    reads (cells.Rho, cells.Velocity)
    writes(cells.Kinetic_Energy)
do
    for cell in cells do
        cell.Kinetic_Energy = 0.5 * cell.Rho * cell.Velocity^2.0
    end
end
```

Soleil-X Task Graph and Profile



CPU Proc 1



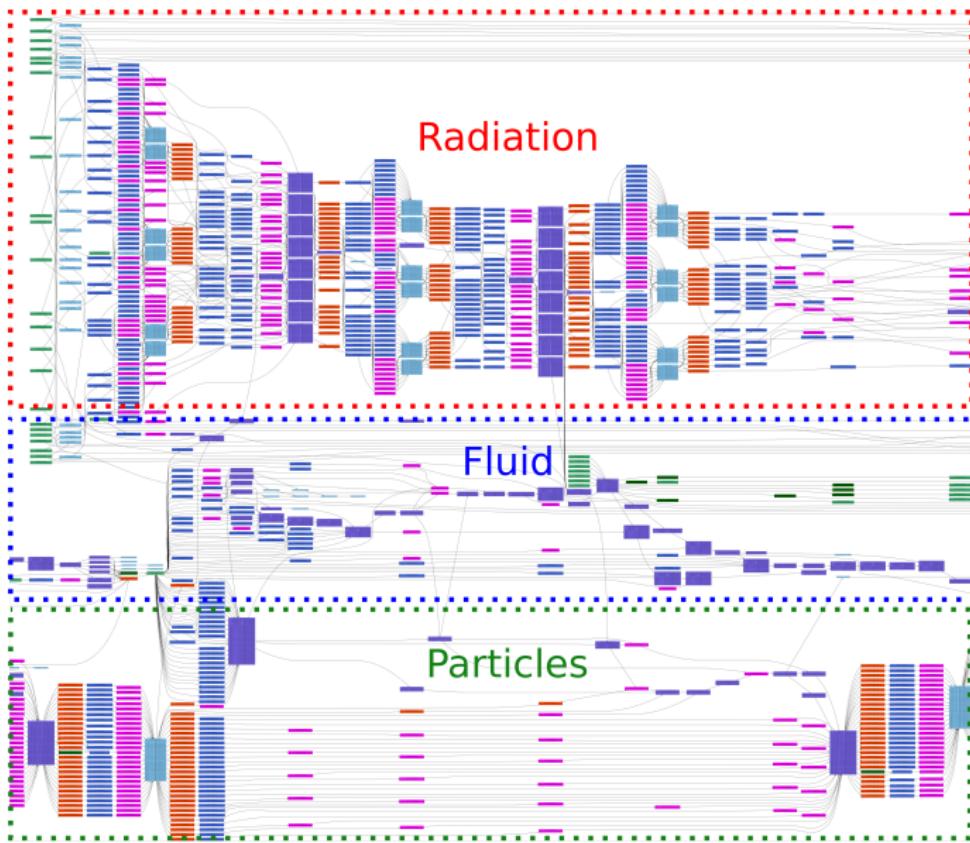
CPU Proc 2



CPU Proc 3



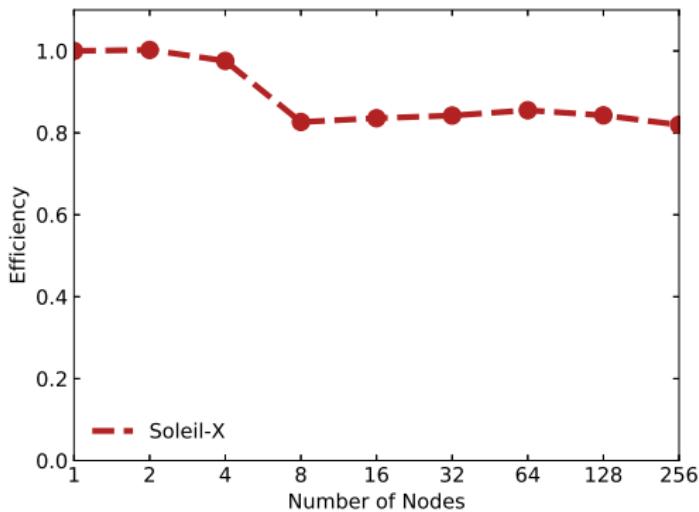
Soleil-X Task Graph



Soleil-X Scaling and Portability

Weak Scaling

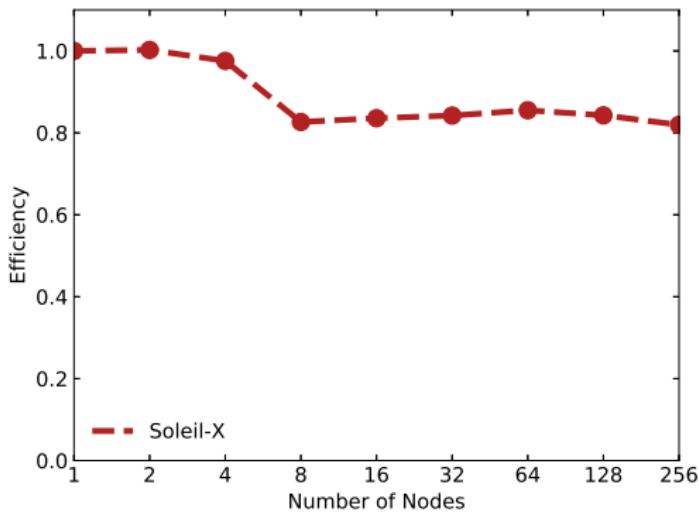
- Sierra with 256^3 fluid cells and 32 million particles per node



Soleil-X Scaling and Portability

Weak Scaling

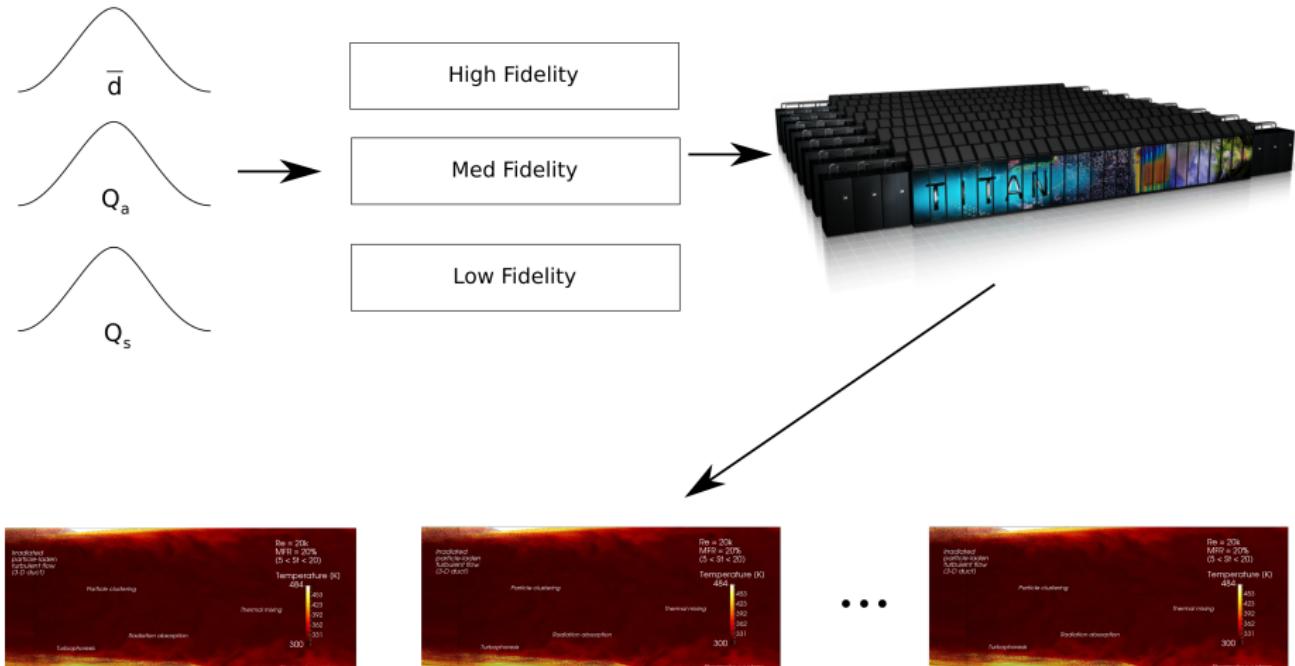
- Sierra with 256^3 fluid cells and 32 million particles per node



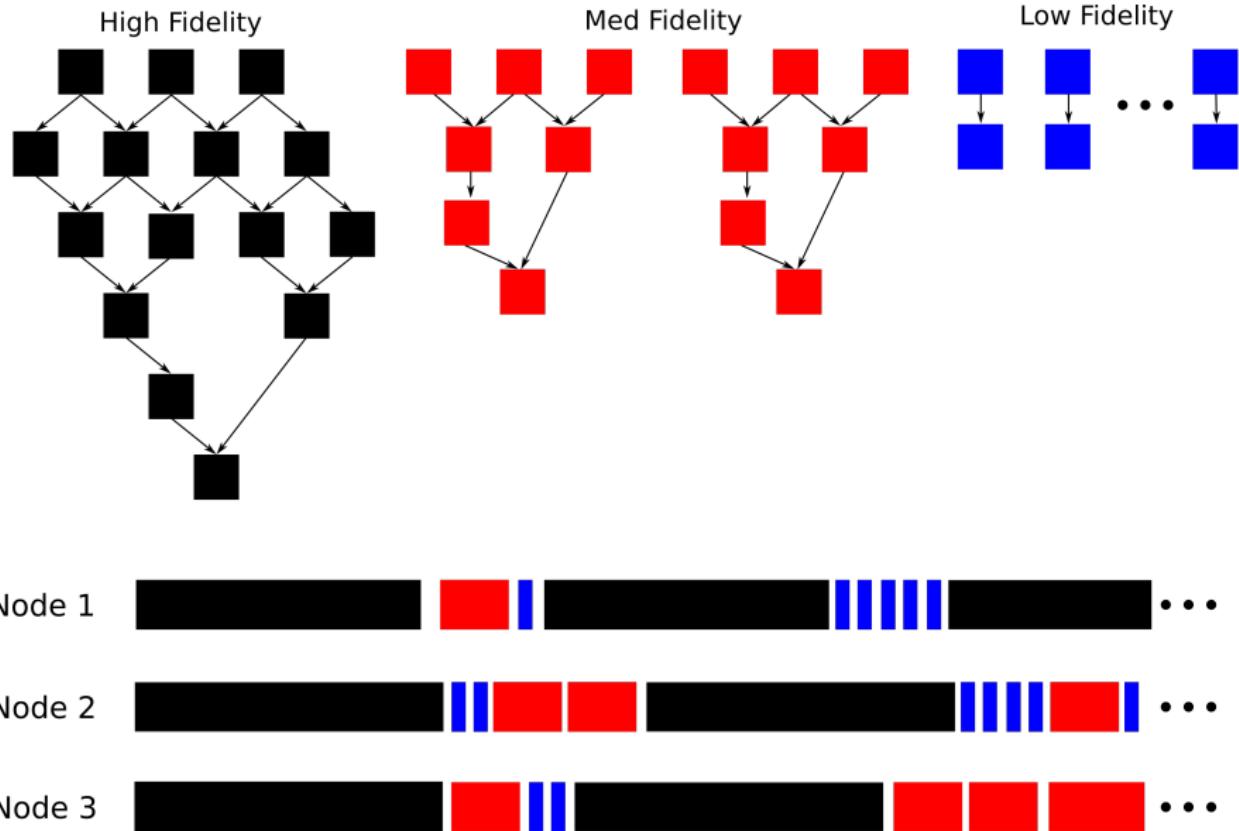
Portability

- Laptops to Workstations
- Local clusters at Stanford: Certainty, Sherlock
- Leadership class systems: Titan, Piz Daint, Lassen, Sierra, Summit

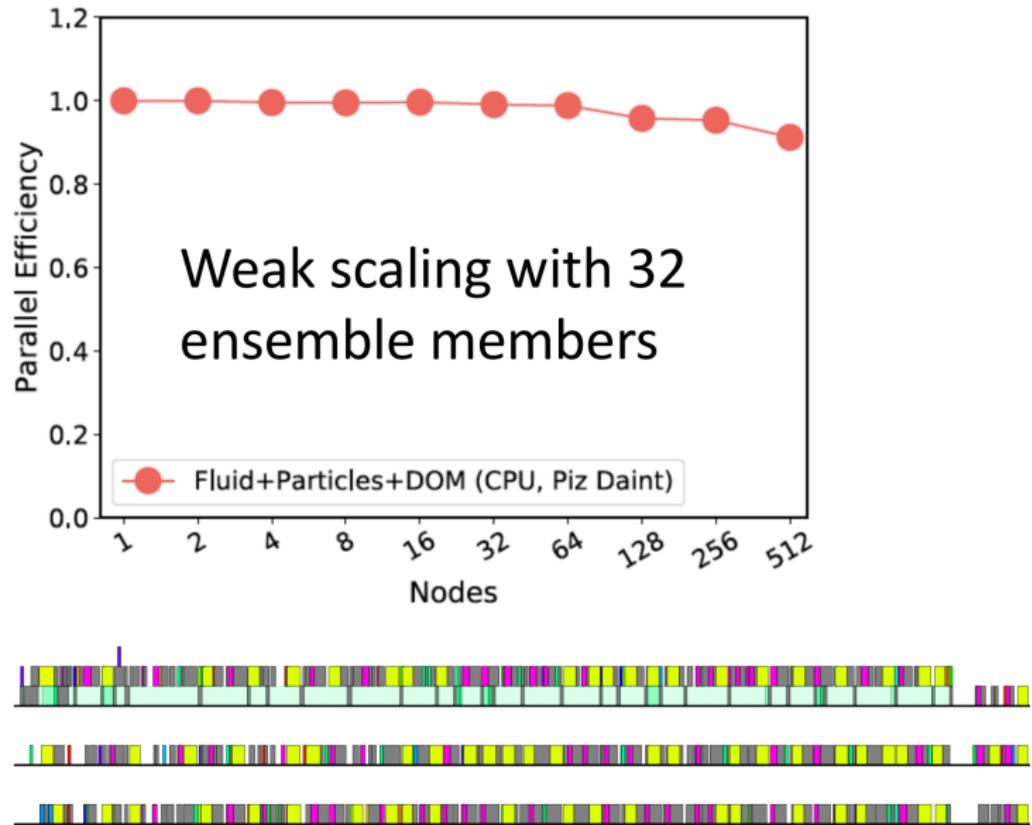
Ensembles for UQ



Ensembles in Regent



Soleil Ensemble Profile



Summary

- Interested in multi-scale multi-physics problems (turbulence, particles, radiation)
- Targeting large heterogeneous machines.
- Using Regent, a high-level counterpart to the task based programming system Legion, to accomplish this.
- Our solver, Soleil, has been deployed on a diverse set of HPC systems.
- Ensembles of Soleil simulations fit into task based framework well.



Questions?



hctorres@stanford.edu

<http://exascale.stanford.edu>

<http://legion.stanford.edu>

