CENTER FOR EXASCALE SIMULATION OF COMBUSTION IN TURBULENCE

SAND2015-2853PE

Combining In-situ and In-transit Processing to Enable Extreme-Scale Scientific Analysis

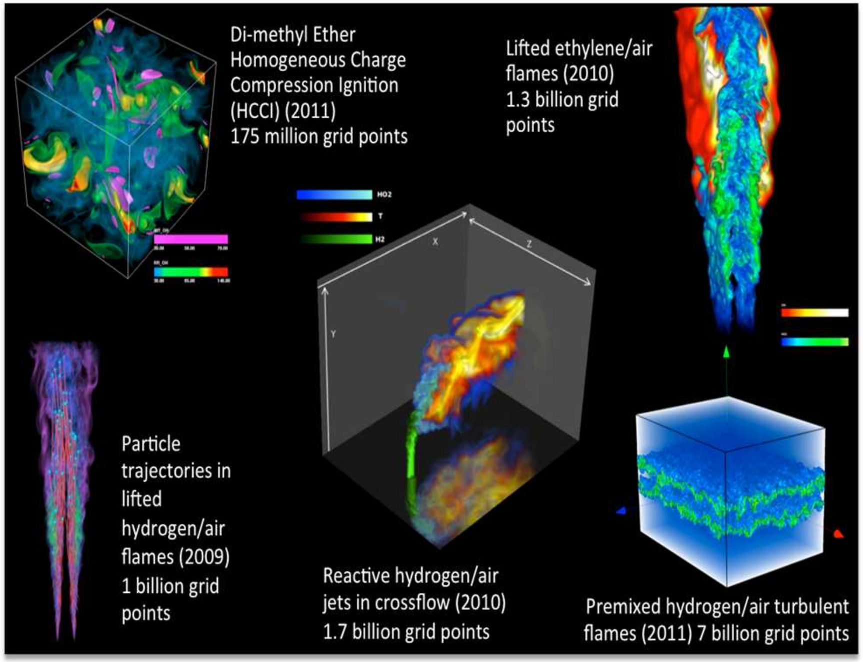
Janine Bennett[1]
Hasan Abbasi[2]
Peer-Timo Bremer[3]
Ray Grout[4]
Attila Gyulassy[5]
Tong Jin[6]
Scott Klasky[2]
Hemanth Kolla[1]
Manish Parashar[6]
Valerio Pascucci[5]
Philippe Pébay[7]
David Thompson[7]
Hongfeng Yu[8]
Fan Zhang[6]
Jacqueline Chen[1]

Sandia National Laboratories[1]
Lawrence Livermore National Laboratory[3]
University of Utah[5]
Kitware[7]
Oakridge National Laboratory[2]
National Renewable Energy Laboratory[4]
Rutgers[6]
University of Nebraska[8]

Science drivers are motivating the need for extreme-scale computing

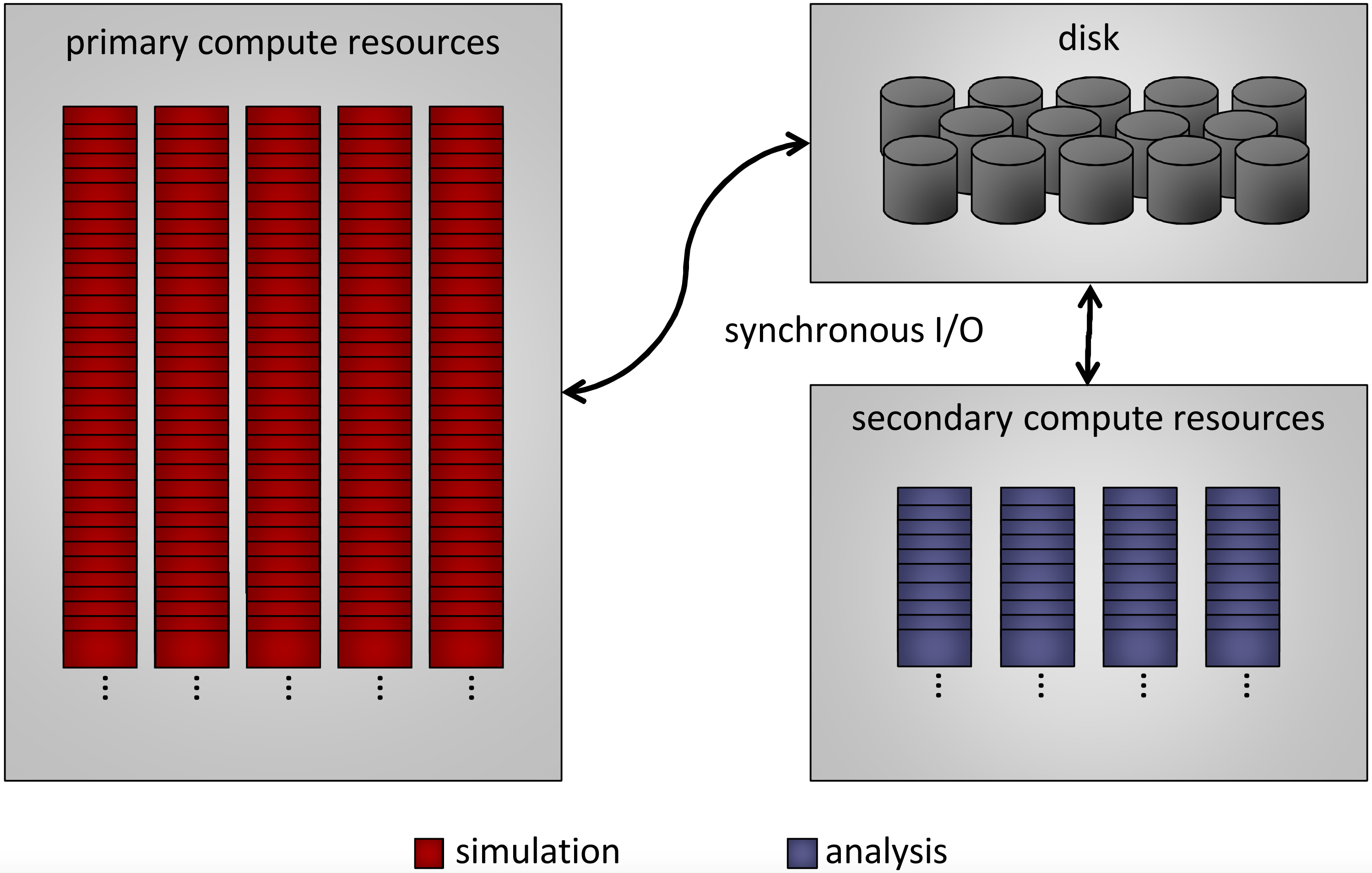
EXaCT CENTER FOR EXASCALE SIMULATION OF COMBUSTION IN TURBULENCE

- S3D: First-principles direct numerical simulation
- Simulation resolves features on the order of 10 simulation time steps
- Currently on the order of every 400th time step is written to disk
- Temporal fidelity is compromised when analysis is done as a post-process

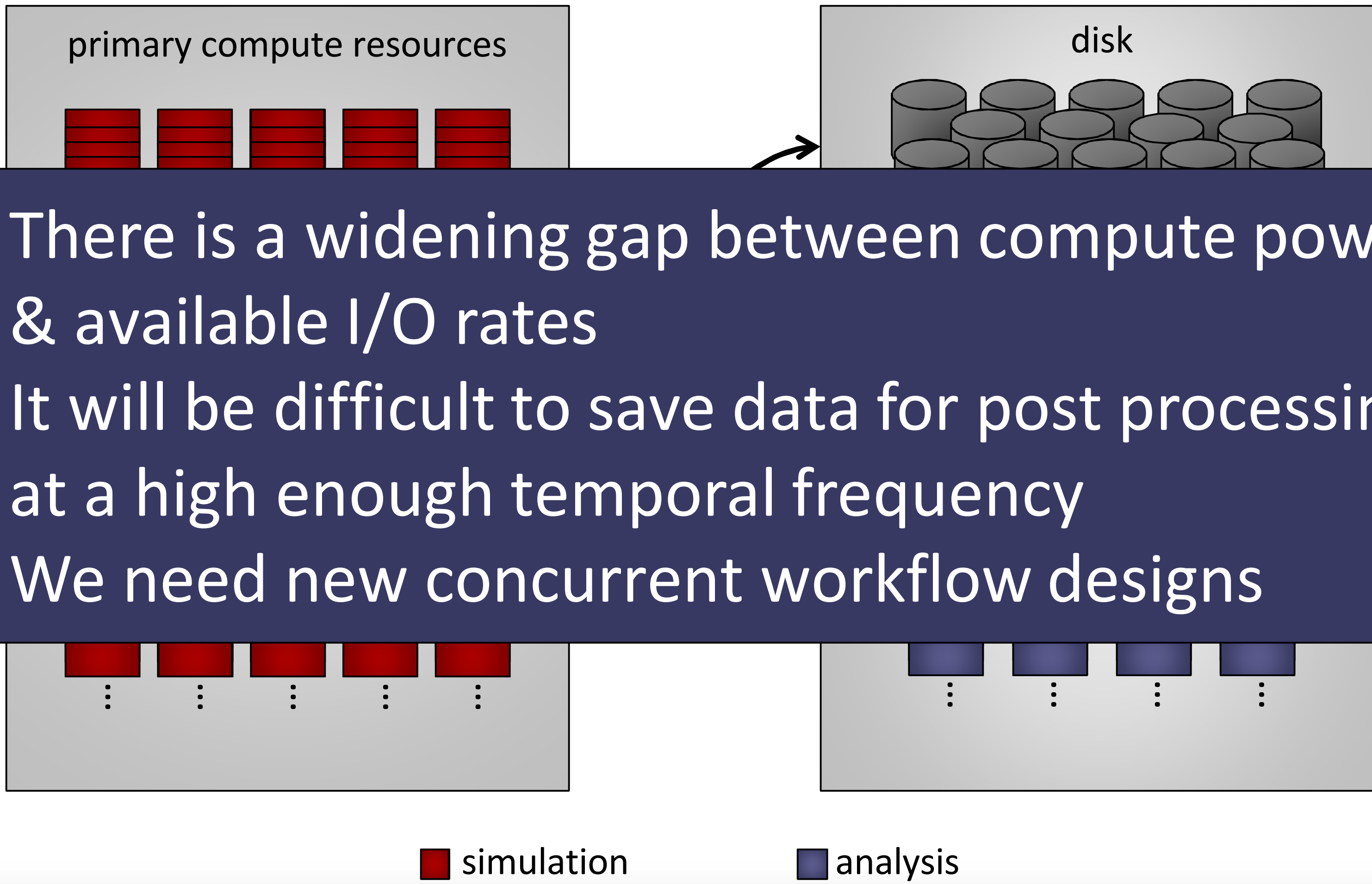


Recent data sets generated by S3D, developed at the Combustion Research Facility, Sandia National Laboratories

The current workflow of compute first, analyze later does not scale on projected high performance computing architectures



The current workflow of compute first, analyze later does not scale on projected high performance computing architectures



- There is a widening gap between compute power & available I/O rates
- It will be difficult to save data for post processing at a high enough temporal frequency
- We need new concurrent workflow designs

Related work

Scalable data movement

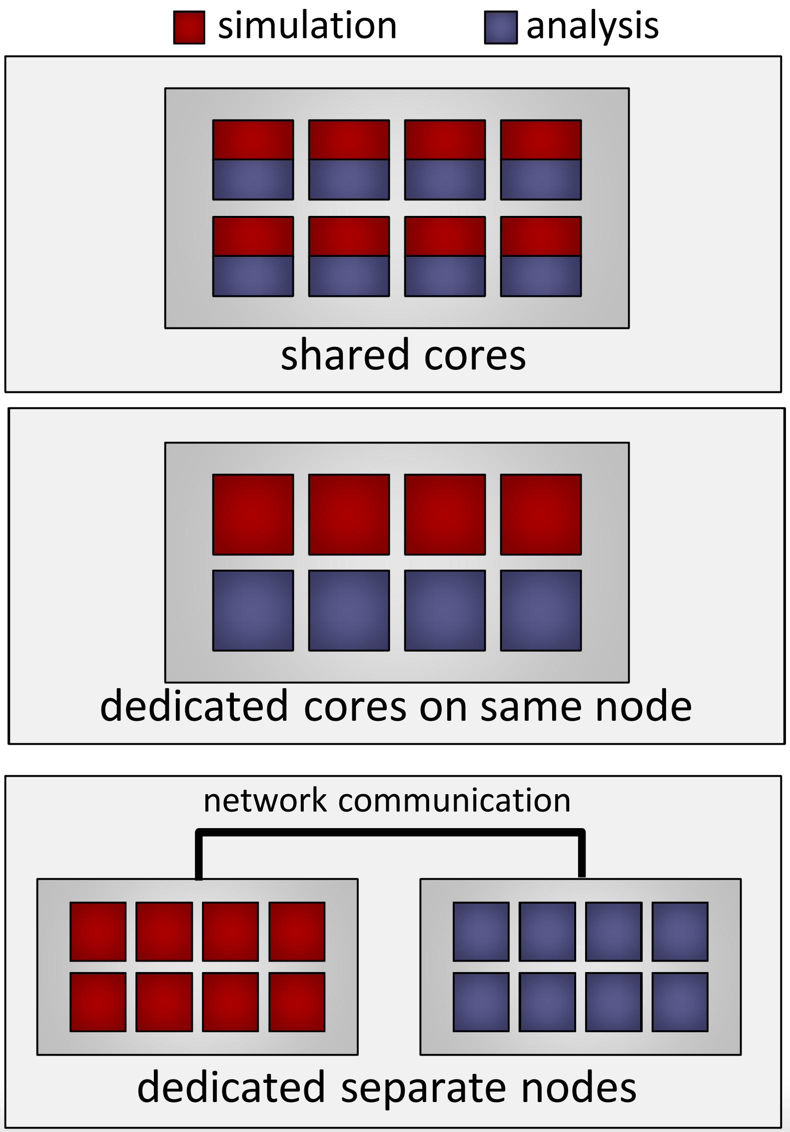
- In-situ
  - DIY: Peterka et al. 2011
  - CoDS: Zhang et al. 2012
  - FP: Li et al. 2010
- Staging
  - Glean: Vishwanath et al. 2011
  - JITStaging: Abbasi et al. 2011
  - PreDatA: Zheng et al. 2010
  - DataSpaces, ActiveSpaces, DART: Docan et al. 2011, 2010
  - Nessie: Loftsead et al. 2011
  - ADIOS: Loftstead et al. 2008

Analytics

- Visit (libsim):
  - Childs et al. 2011, Howison et al. 2010
  - http://wci.llnl.gov/codes/visit
- Paraview (catalyst):
  - Fabian et al. 2011, Cedilnik et al. 2006
  - http://www.paraview.org
- Other parallel analytics:
  - Tu et al. 2006, Yu et al. 2006, 2010, Gyulassy et al. 2012, Pébay et al. 2011, Camp et al. 2010, Pugmire et al 2009

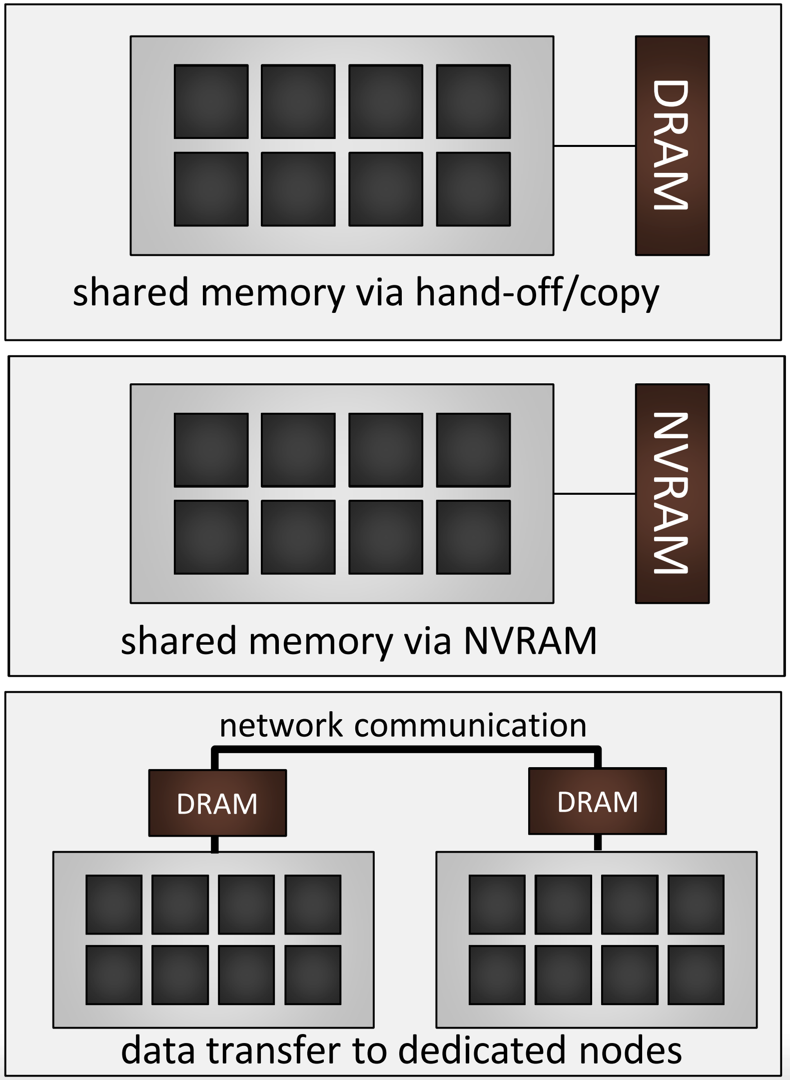
There is a rich design space of potential workflow designs on future HPC systems

- Location of analysis compute resources
  - Same cores as the simulation (in-situ)
  - Dedicated cores on the same node (in-situ)
  - Dedicated nodes on the same machine (in-transit)
  - Dedicated nodes on external resource (in-transit)



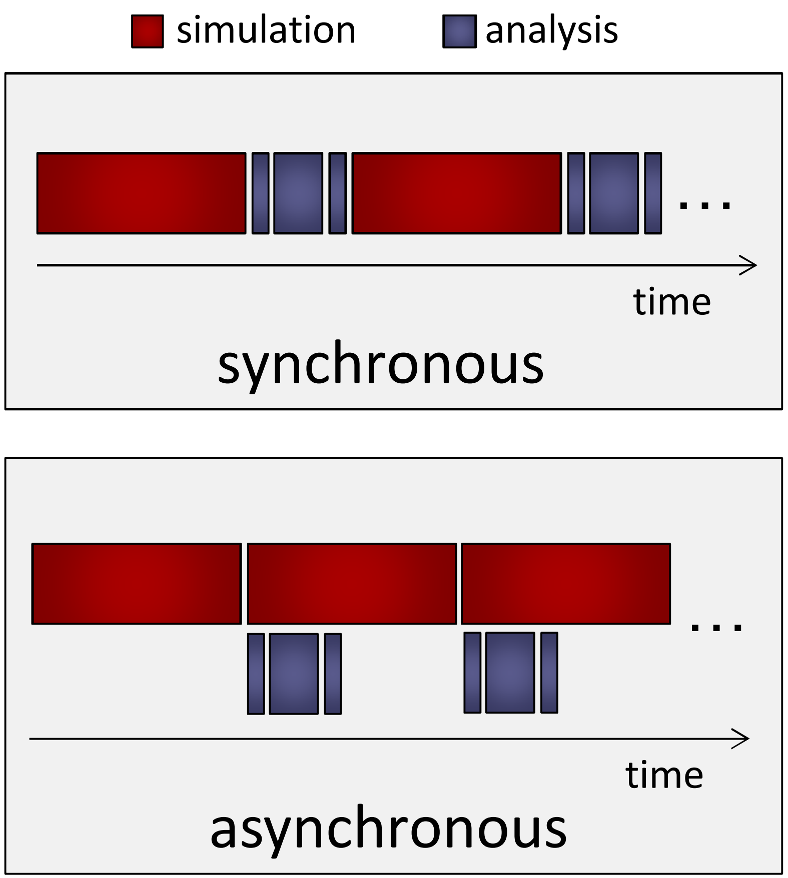
There is a rich design space of potential workflow designs on future HPC systems

- Location of analysis compute resources
  - Same cores as the simulation (in-situ)
  - Dedicated cores on the same node (in-situ)
  - Dedicated nodes on the same machine (in-transit)
  - Dedicated nodes on external resource (in-transit)
- Data access, placement, and persistence
  - Shared memory access via hand-off / copy
  - Shared memory access via non-volatile near node storage (NVRAM)
  - Data transfer to dedicated nodes or external resources



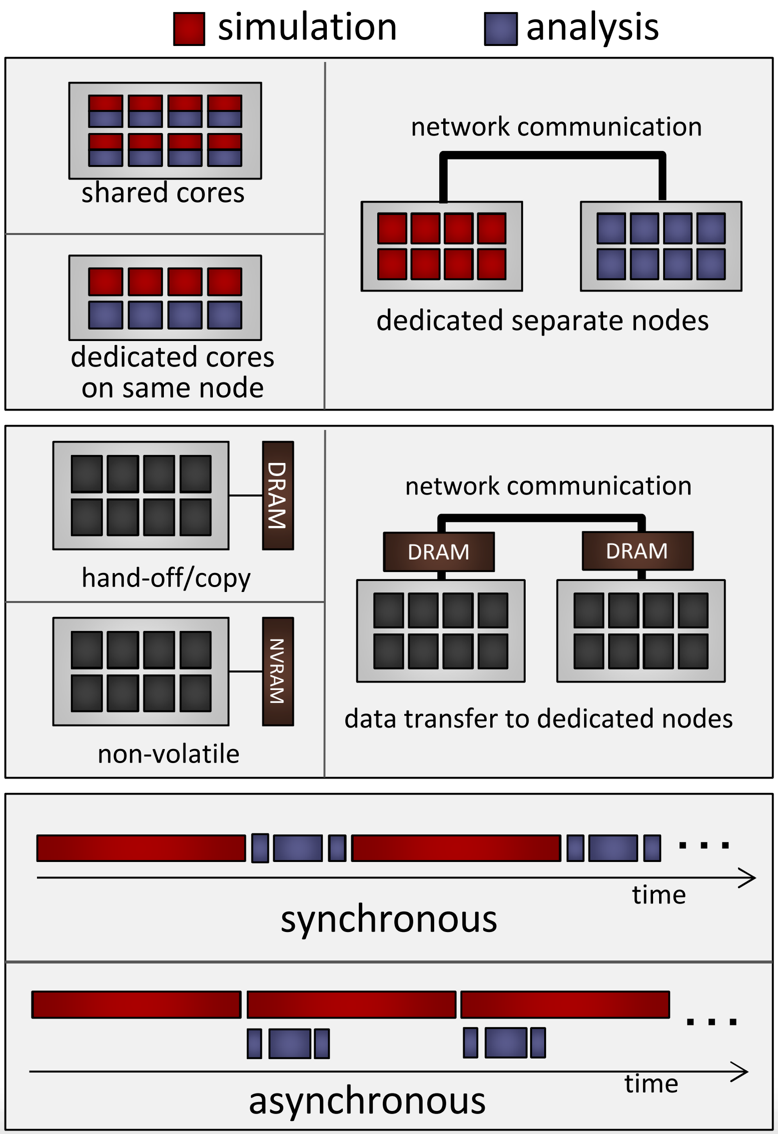
There is a rich design space of potential workflow designs on future HPC systems

- Location of analysis compute resources
  - Same cores as the simulation (in-situ)
  - Dedicated cores on the same node (in-situ)
  - Dedicated nodes on the same machine (in-transit)
  - Dedicated nodes on external resource (in-transit)
- Data access, placement, and persistence
  - Shared memory access via hand-off / copy
  - Shared memory access via non-volatile near node storage (NVRAM)
  - Data transfer to dedicated nodes or external resources
- Synchronization and scheduling
  - Execute synchronously with simulation every n^{th} simulation time step
  - Execute asynchronously



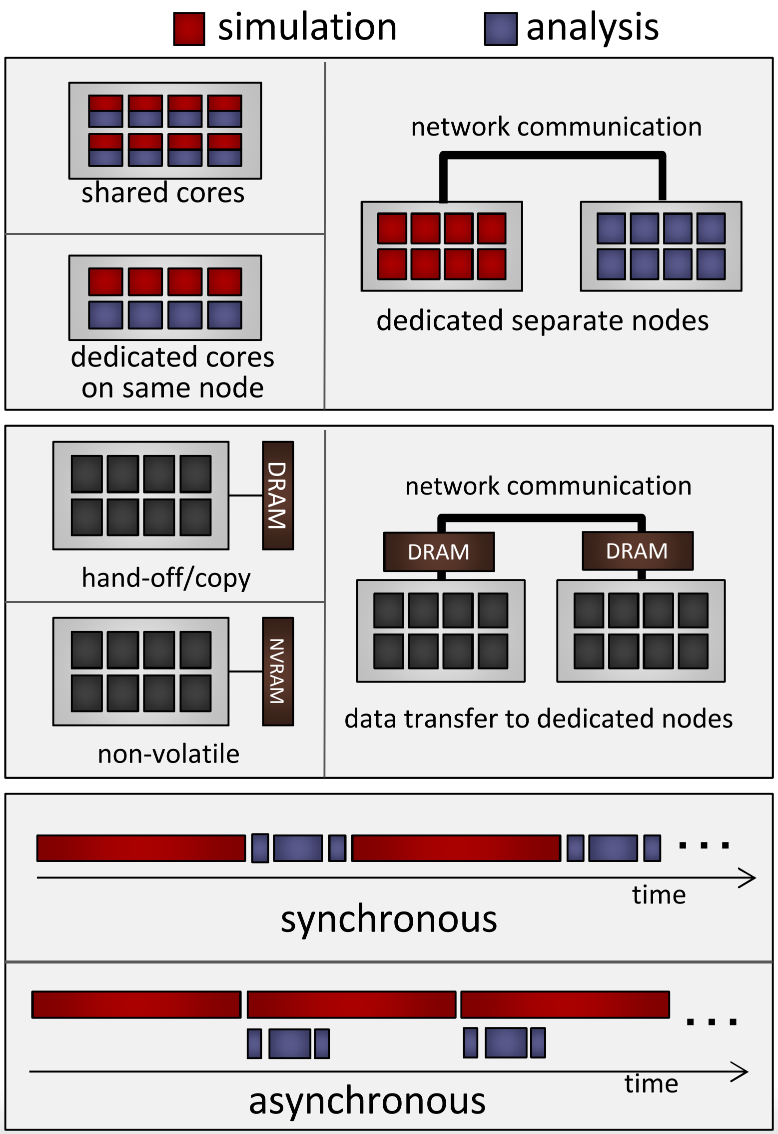
Workflow designs can have a significant impact on design and implementation of analysis algorithms

- **Explore the design space of new workflows**
  - Location of analysis compute resources
  - Data access, placement, and persistence
  - Synchronization and scheduling
- **Investigate impact of workflows on analysis algorithms**
  - In-situ
  - In-transit
  - Hybrid in-situ + in-transit



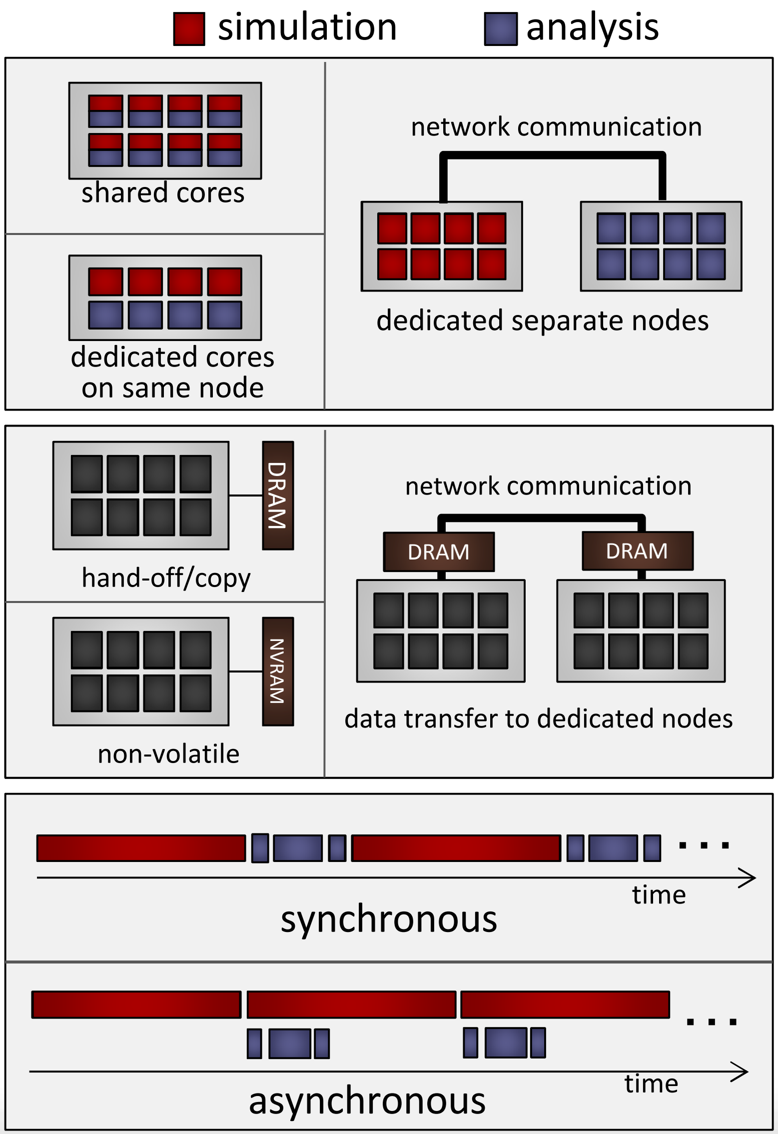
Exploring the design space of new workflows

- Location of analysis compute resources
  - **Same cores as the simulation (in-situ)**
  - Dedicated cores on the same node (in-situ)
  - **Dedicated nodes on the same machine (in-transit)**
  - Dedicated nodes on external resource (in-transit)
- Data access, placement, and persistence
  - **Shared memory access via hand-off / copy**
  - Shared memory access via non-volatile near node storage (NVRAM)
  - **Data transfer to dedicated nodes or external resources**
- Synchronization and scheduling
  - **Execute synchronously with simulation every n^{th} simulation time step**
  - **Execute asynchronously**



Investigating the impact of workflow designs on analyses

- Algorithmic variants
  - In-situ
  - In-transit
  - Hybrid in-situ + in-transit
- Which variant is best for a given algorithm?
- Is it dependent on workflow design?
- Where/how to decompose algorithms for hybrid analyses?

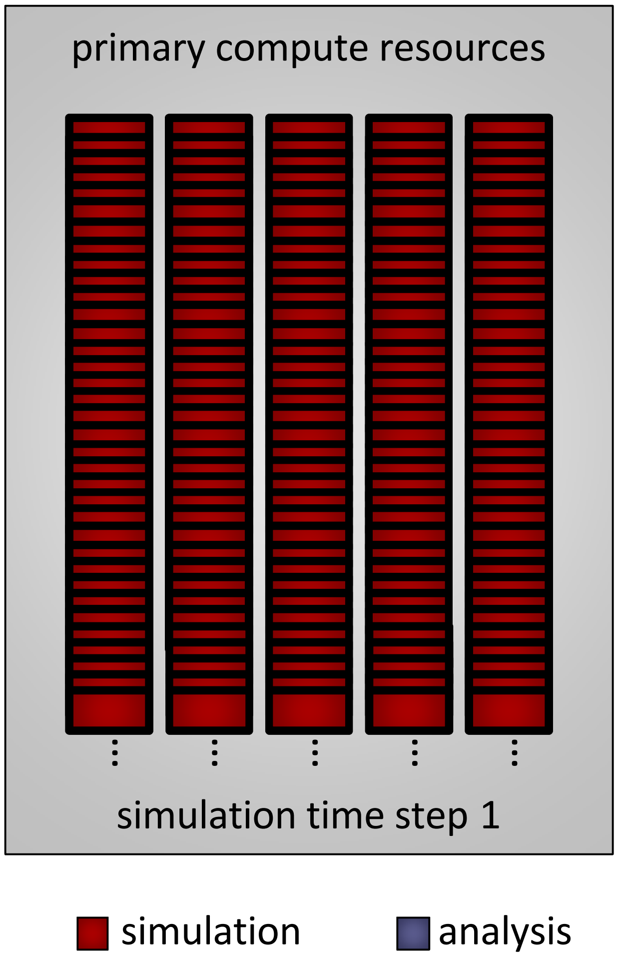


Exploring the design space of workflows: Constraints and observations

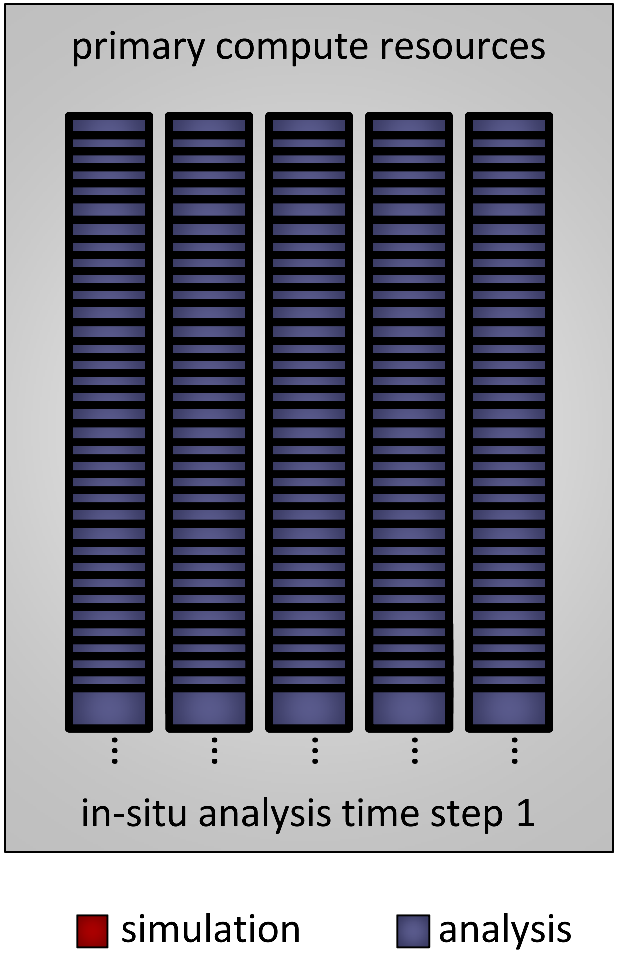
- We must minimize performance impact to the simulation
  - Work within time and memory constraints
  - Minimize cache impact

- The behavior of analysis algorithms varies widely
  - Data dependencies, communication patterns, scalability, instruction mixes, time and memory requirements
  - Data dependent algorithms are very hard to characterize

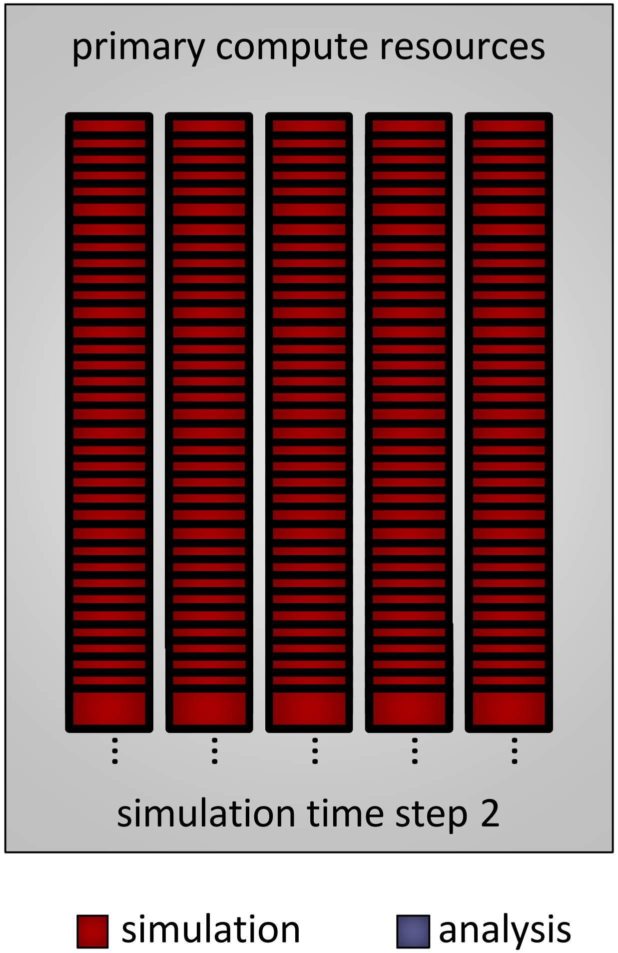
Exploring the design space of workflows: In-situ workflow performed synchronously with simulation



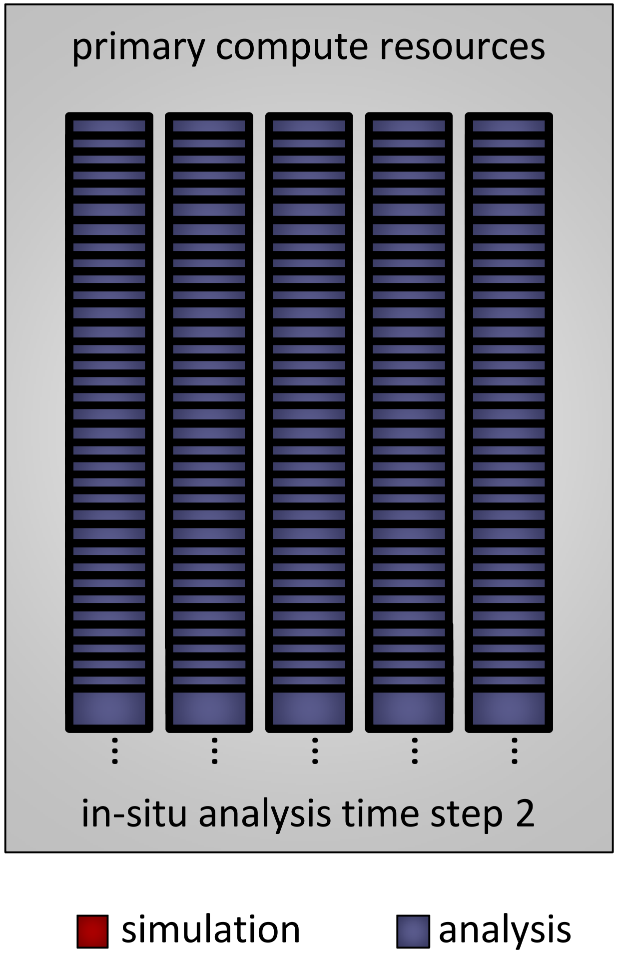
Exploring the design space of workflows: In-situ workflow performed synchronously with simulation



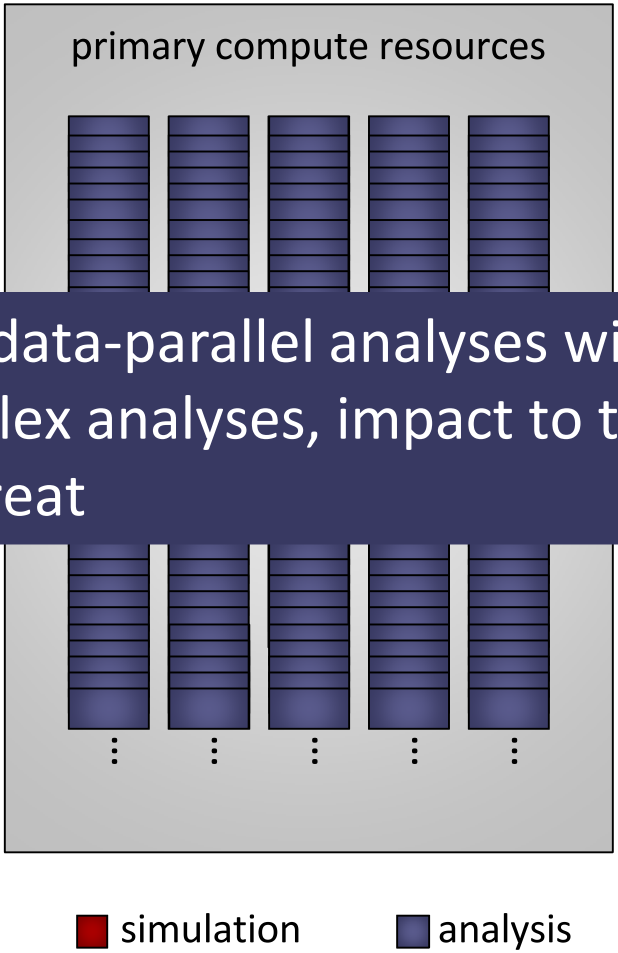
Exploring the design space of workflows: In-situ workflow performed synchronously with simulation



Exploring the design space of workflows: In-situ workflow performed synchronously with simulation

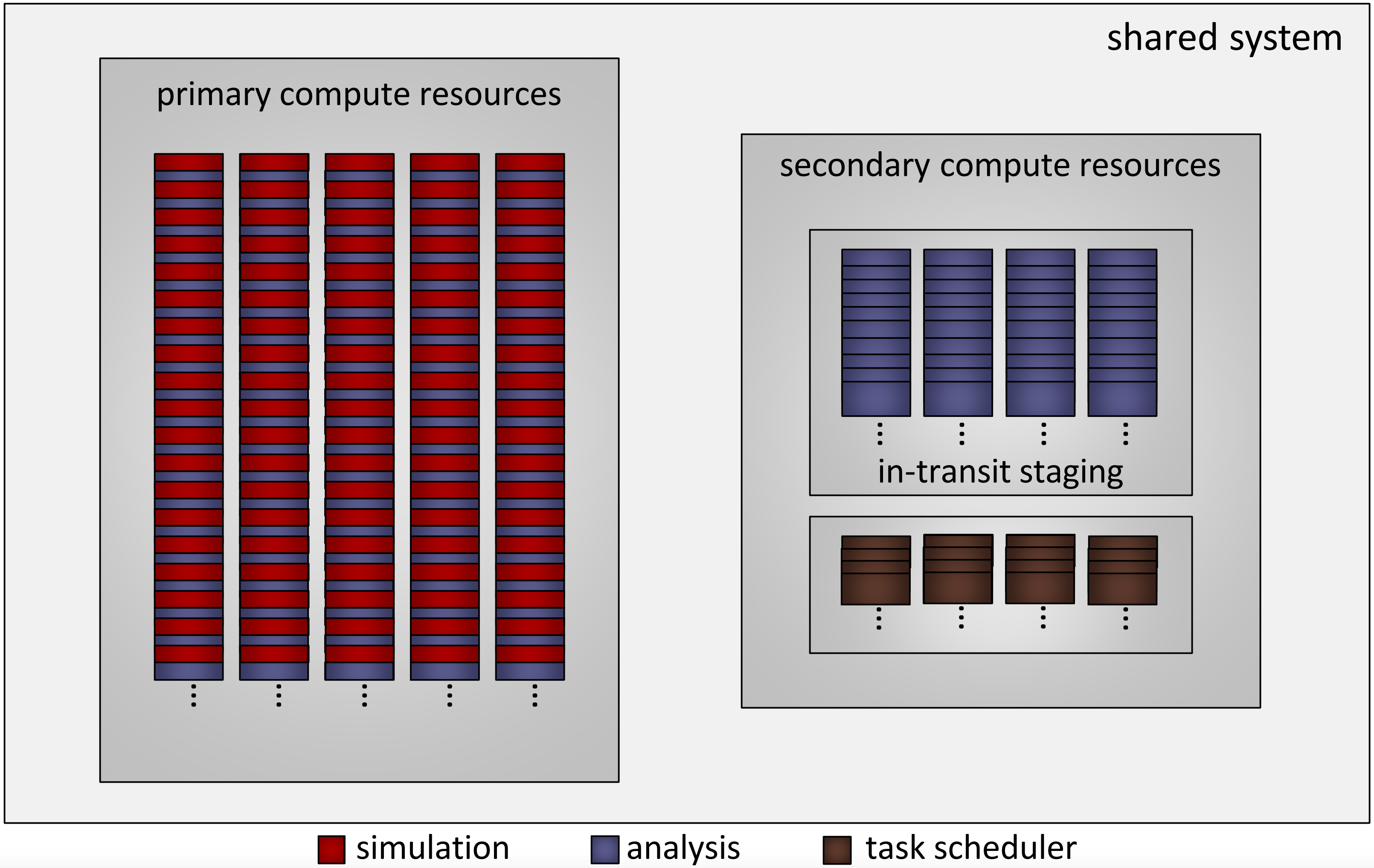


Exploring the design space of workflows: In-situ workflow performed synchronously with simulation

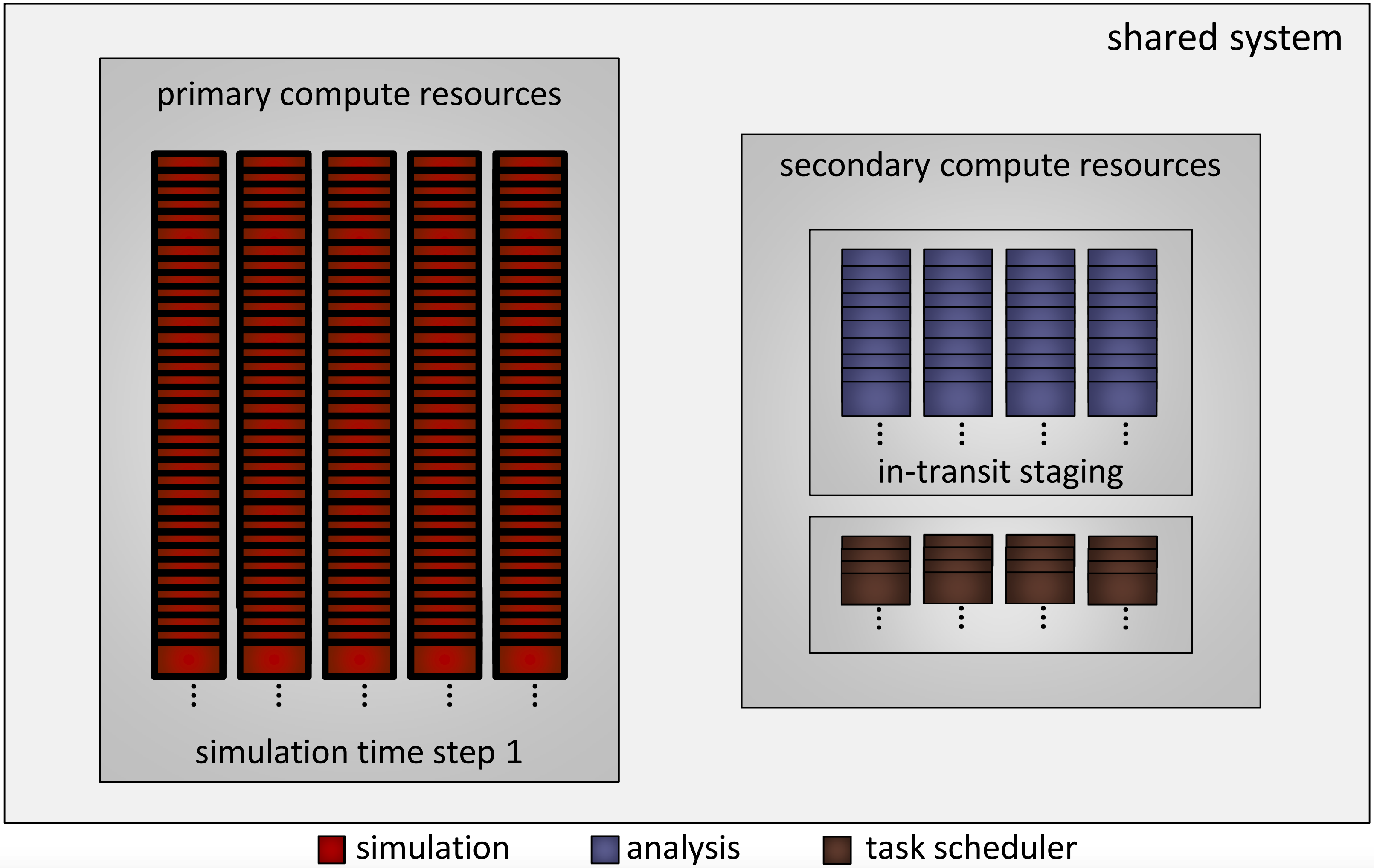


- Works well for data-parallel analyses with short run times
- For more complex analyses, impact to the simulation becomes too great

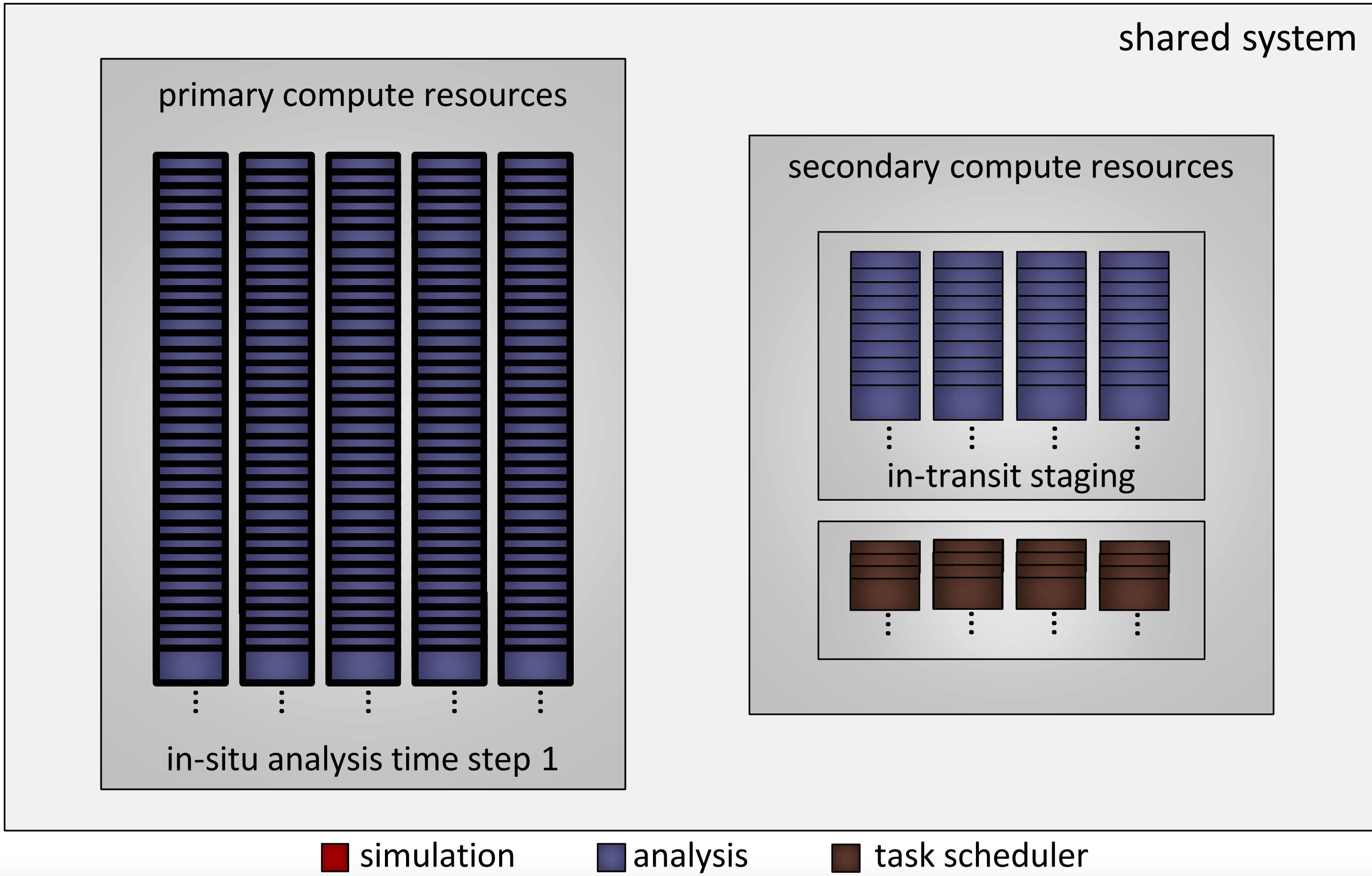
Exploring the design space of workflows: Temporally multiplexed hybrid in-situ + in-transit workflow



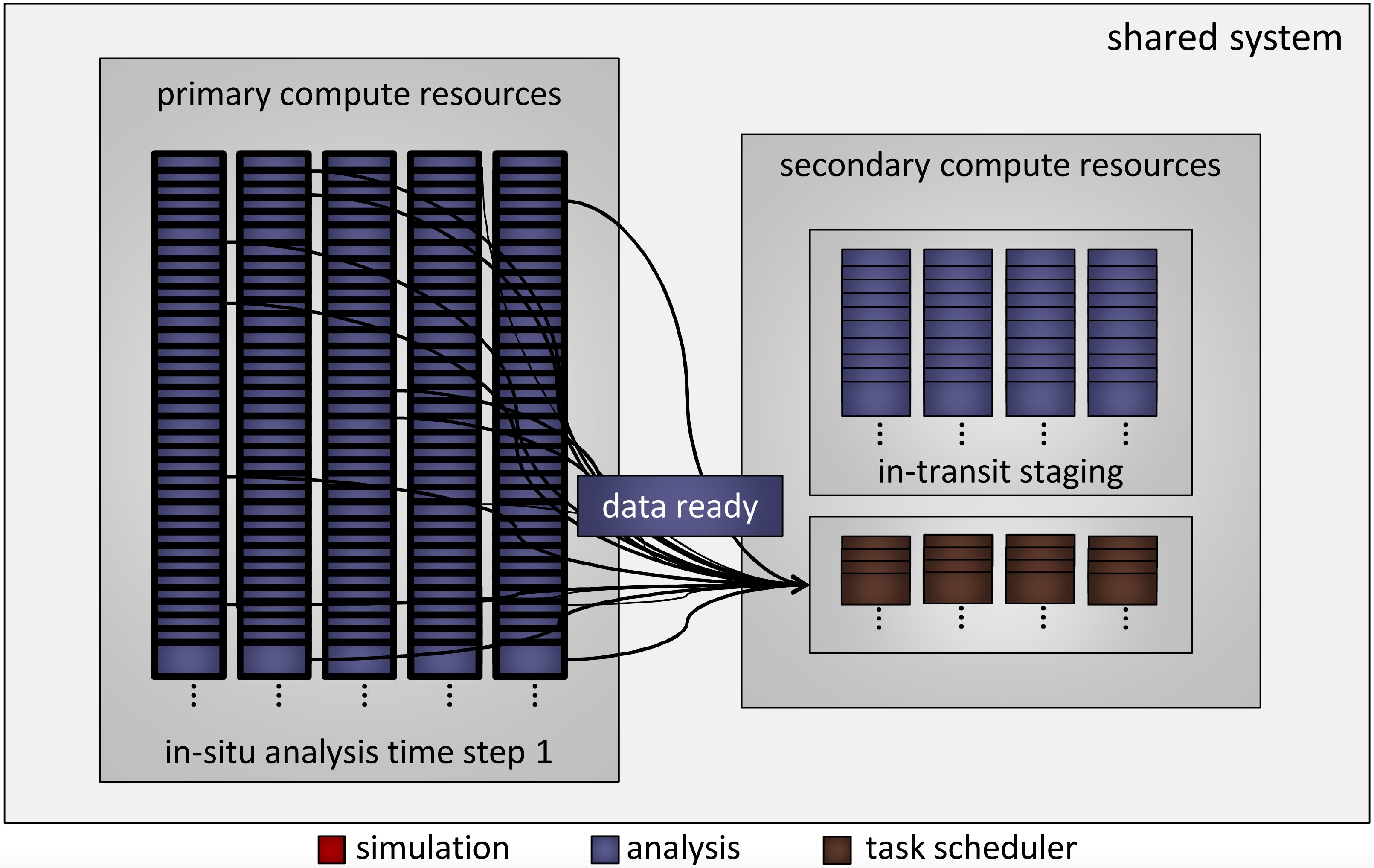
Exploring the design space of workflows: Temporally multiplexed hybrid in-situ + in-transit workflow



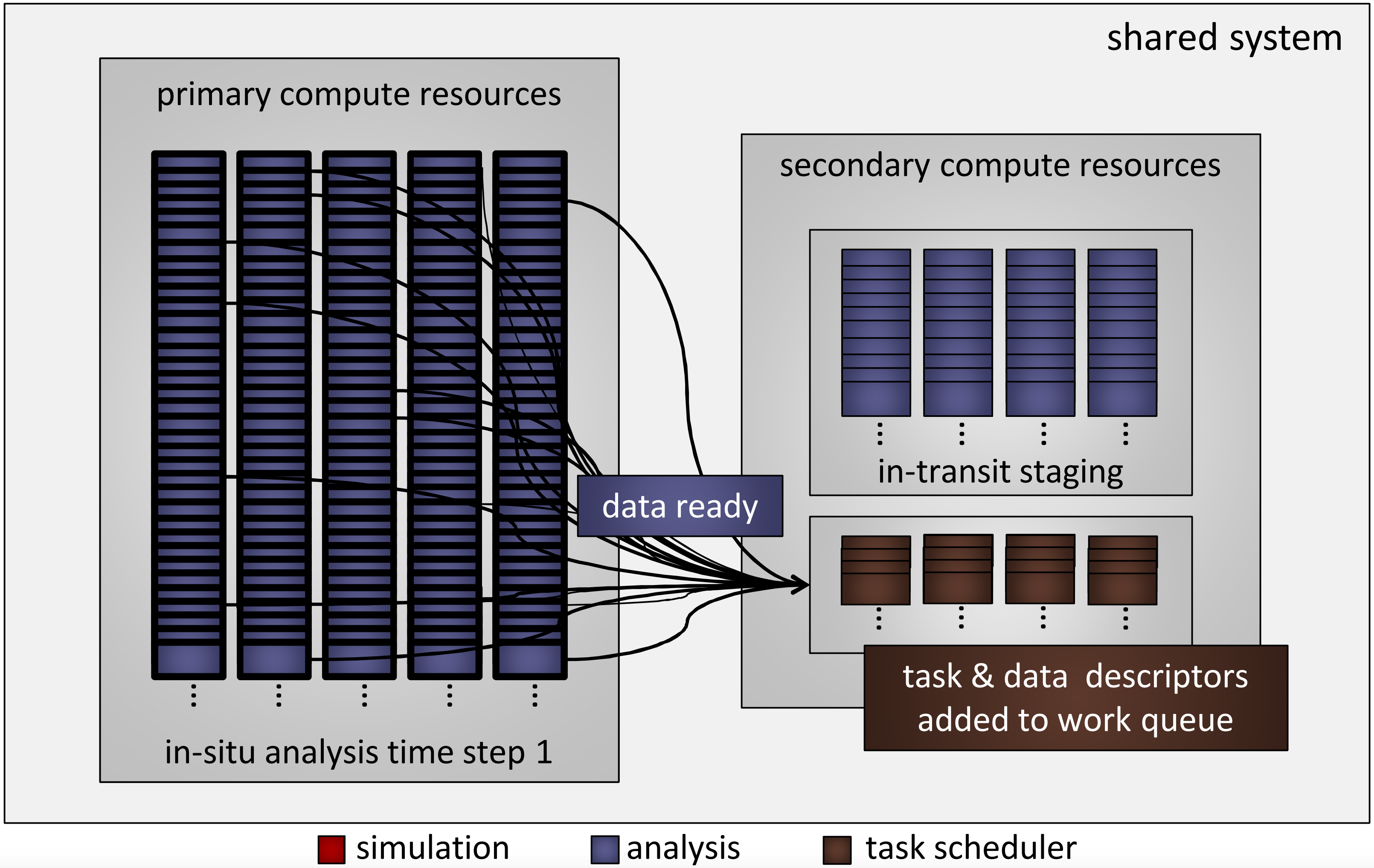
Exploring the design space of workflows: Temporally multiplexed hybrid in-situ + in-transit workflow



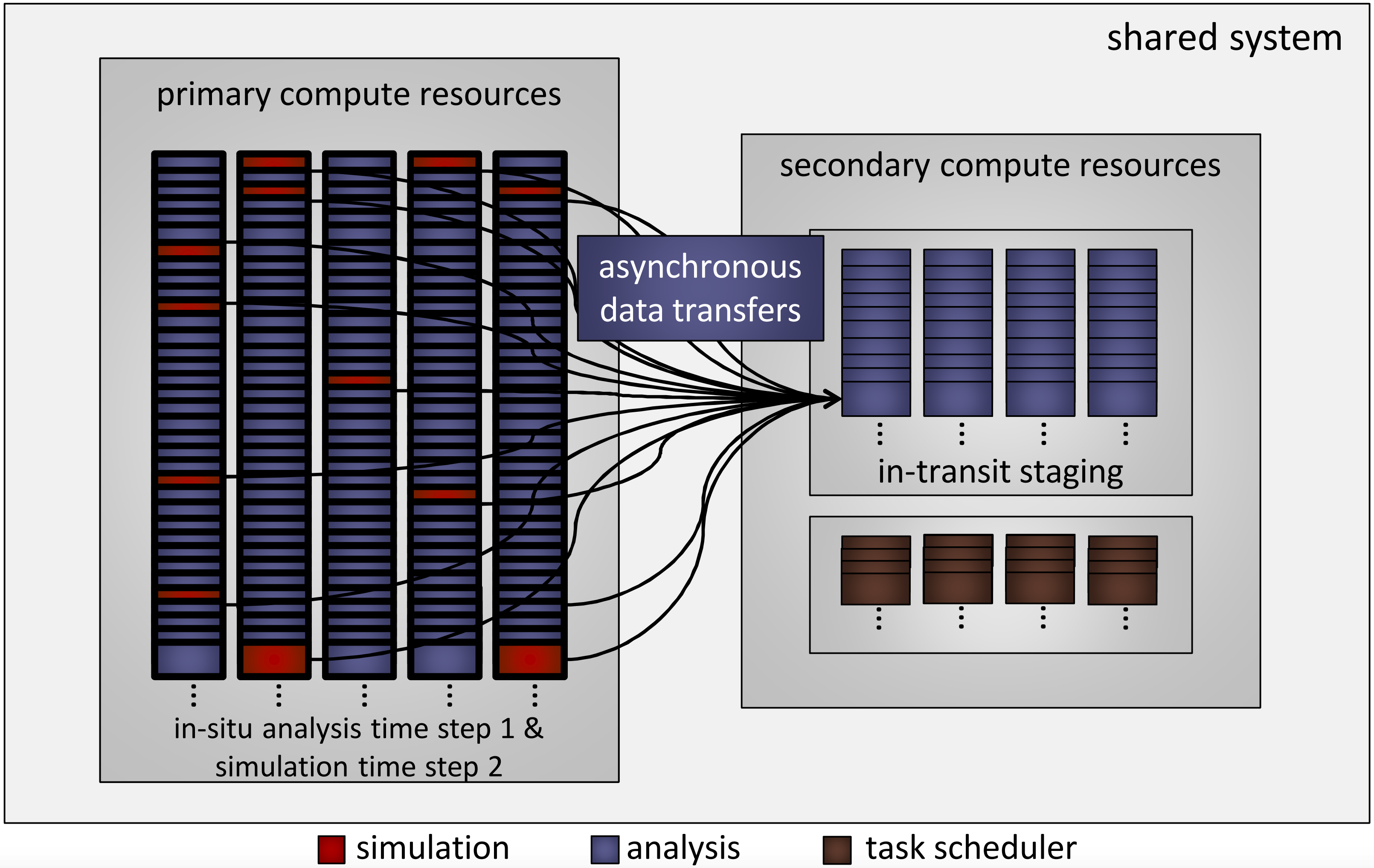
Exploring the design space of workflows: Temporally multiplexed hybrid in-situ + in-transit workflow



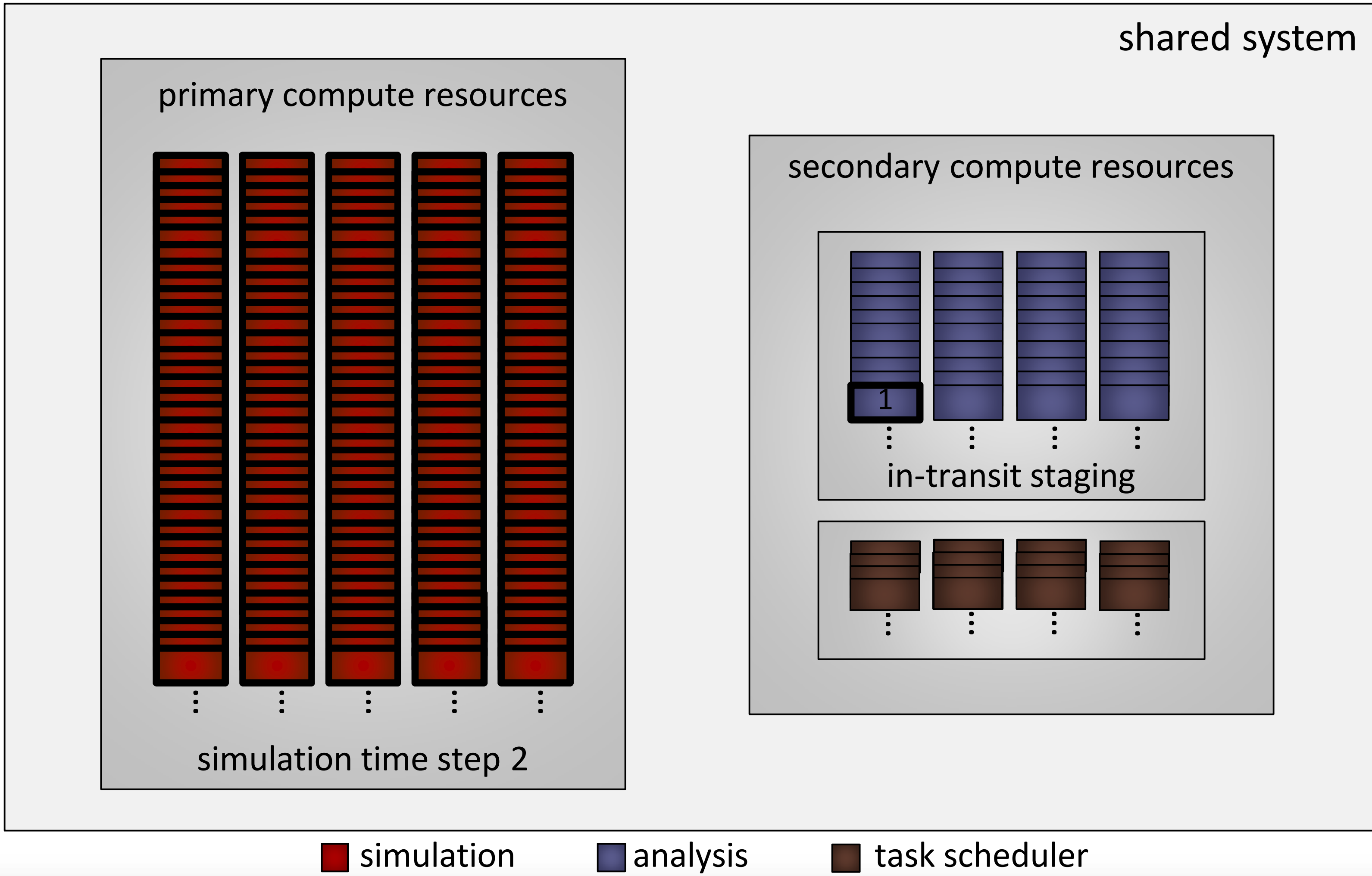
Exploring the design space of workflows: Temporally multiplexed hybrid in-situ + in-transit workflow



Exploring the design space of workflows: Temporally multiplexed hybrid in-situ + in-transit workflow



Exploring the design space of workflows: Temporally multiplexed hybrid in-situ + in-transit workflow

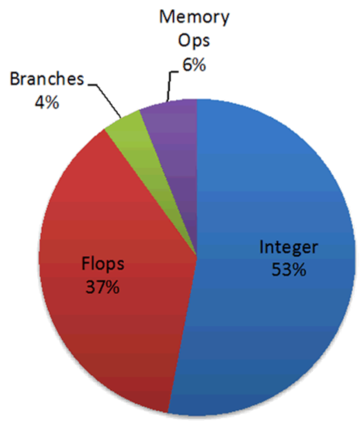


Hybrid workflow design impacts the manner in which analysis is performed

Hybrid analysis requires decomposition of algorithms into 2 stages

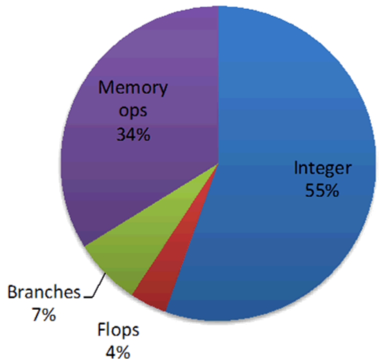
In-situ	In-transit
Data-parallel	More forgiving of complex communication needs
Short run time with respect to simulation	Can have longer run times while minimizing impact to simulation
Limited amount of memory; minimize cache impacts	Limited to memory and processing constraints of secondary resources
Should minimize the amount of data sent in-transit	Can only require data sent by in-situ stage

Investigating impact of workflow design on analyses: We focused on 3 algorithms with different characteristics



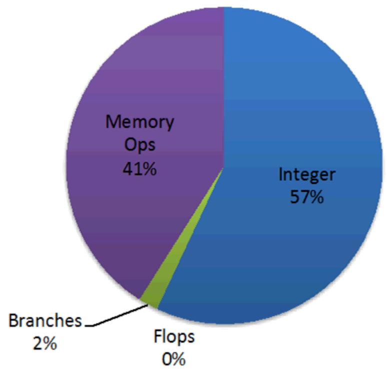
- Complexity: not data dependent
- Communication: small, fixed

Statistics



- Complexity: data dependent
- Communication: data dependent

Volume Rendering



- Complexity: very data dependent
- Communication: very data dependent

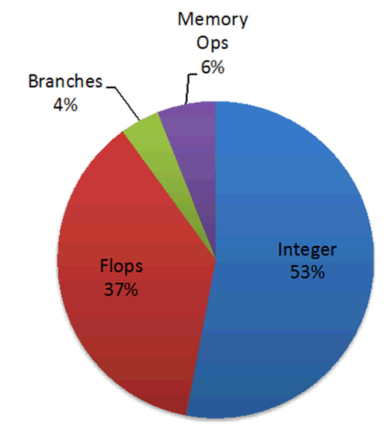
Topology

Instruction mixes obtained with Byfl
https://github.com/losalamos/Byfl

Investigating impact of workflow design on analyses: Streaming statistical analysis

- Purpose:
  - Quantitative summary of global trends in the data
  - Debugging & analysis
- Algorithmic details:
  - Compute 1st-4th order moments locally
  - Aggregate using pair-wise update formulas
- Variants Implemented:
  - In-situ local moments & aggregation
  - In-situ local moments + in-transit aggregation

$$
\begin{aligned}
M_{p,\mathscr{S}} &= M_{p,\mathscr{S}_1} + M_{p,\mathscr{S}_2} \\
&\quad + \sum_{k=1}^{p-2} \binom{k}{p} \left[\left(-\frac{n_2}{n} \right)^k M_{p-k,\mathscr{S}_1} + \left(\frac{n_1}{n} \right)^k M_{p-k,\mathscr{S}_2} \right] \delta_{2,1}^k \\
&\quad + \left(\frac{n_1 n_2}{n} \delta_{2,1} \right)^p \left[\frac{1}{n_2^{p-1}} - \left(\frac{-1}{n_1} \right)^{p-1} \right].
\end{aligned}
$$

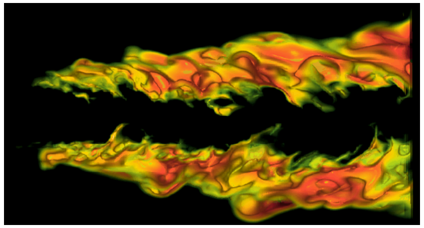


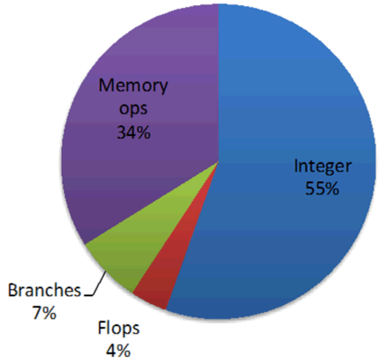
- Complexity: not data dependent
- Communication: small, fixed

Statistics

Investigating impact of workflow design on analyses: Parallel volume rendering

- Purpose:
  - Qualitative visual depiction of data
  - Debugging & analysis
- Algorithmic details:
  - Volume render local data generating partial images
  - Combine partial images via image compositing
- Variants Implemented:
  - In-situ volume rendering & compositing
  - In-situ down-sampling + in-transit rendering & compositing



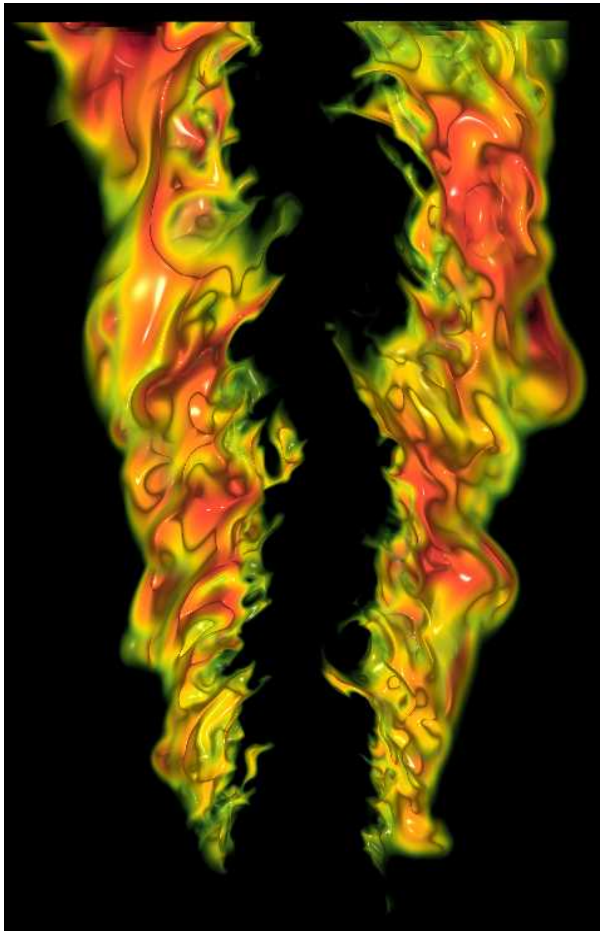


- Complexity: data dependent
- Communication: data dependent

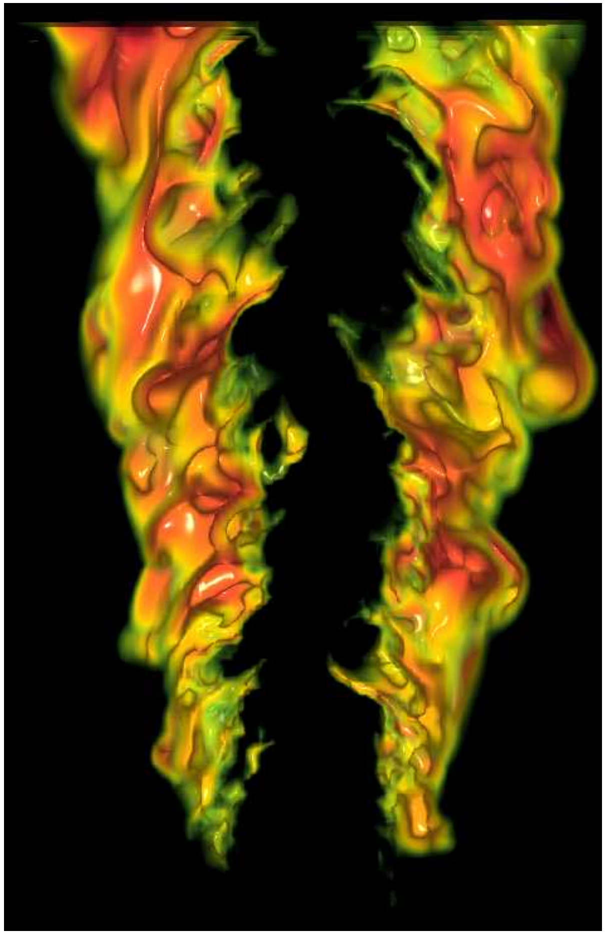
Volume Rendering

Investigating impact of workflow design on analyses: Parallel volume rendering

Down sampling can provide a sufficiently accurate depiction, particularly when debugging



full resolution



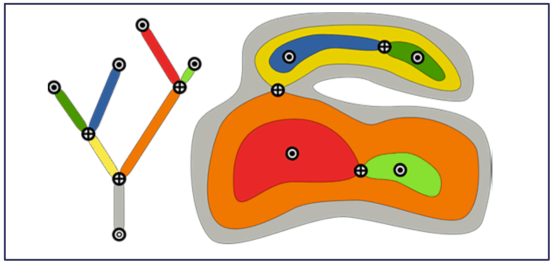
down sampled

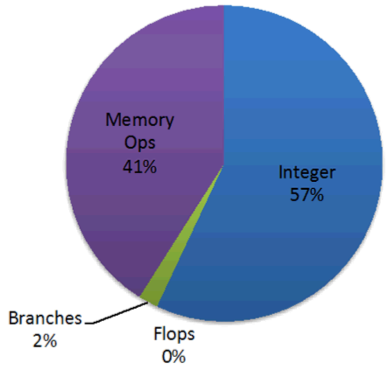
Investigating impact of workflow design on analyses: Reduced topology computation

- Purpose:
  - Complete characterization of level-set behavior of simulation variables
  - Used to define features of interest
  - Analysis

- Algorithmic details:
  - Compute local merge trees
  - Integrate to resolve features spanning multiple cores
  - Adjust local merge trees

- Variants Implemented:
  - In-situ local tree computation + in-transit integration





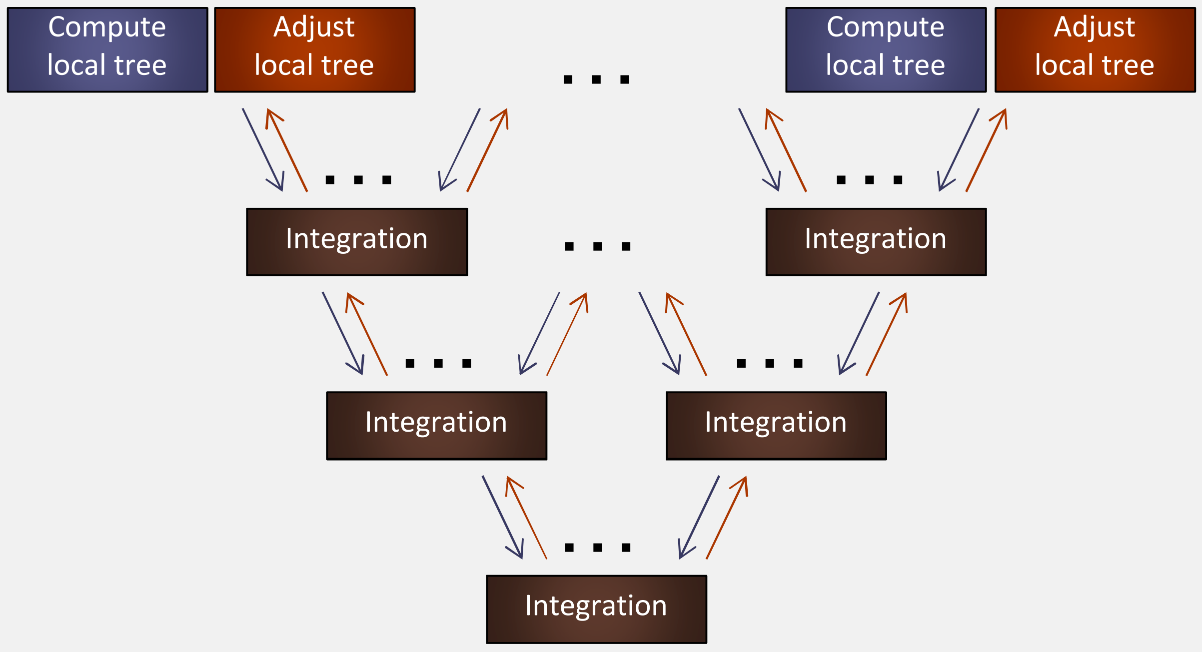
- Complexity: very data dependent
- Communication: very data dependent

Topology

Investigating impact of workflow design on analyses: Reduced topology computation

- Complex communication patterns & data dependencies make in-situ topological analysis a challenging research opportunity

- Algorithmic variants exist that tradeoff between amount of system resources used, simplicity, latency, and duplication of work



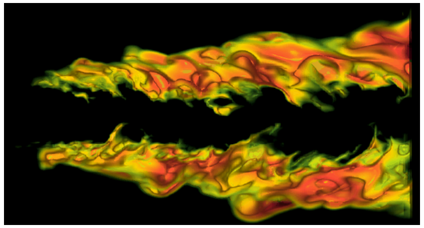
Local trees can be integrated in a N to 1 communication pattern

Simulation case study with S3D, a massively parallel turbulent combustion code

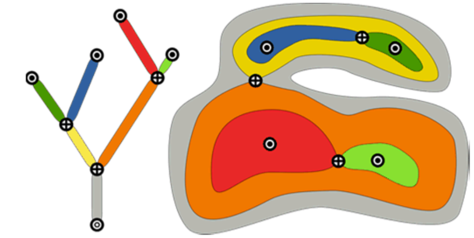
S3D

$$M_{p,\mathscr{S}} = M_{p,\mathscr{S}_1} + M_{p,\mathscr{S}_2} + \sum_{k=1}^{p-2} \binom{k}{p} \left[\left(-\frac{n_2}{n}\right)^k M_{p-k,\mathscr{S}_1} + \left(\frac{n_1}{n}\right)^k M_{p-k,\mathscr{S}_2} \right] \delta_{2,1}^k + \left(\frac{n_1 n_2}{n} \delta_{2,1}\right)^p \left[\frac{1}{n_2^{p-1}} - \left(\frac{-1}{n_1}\right)^{p-1} \right]. \qquad \text{(III.1)}$$

Statistics



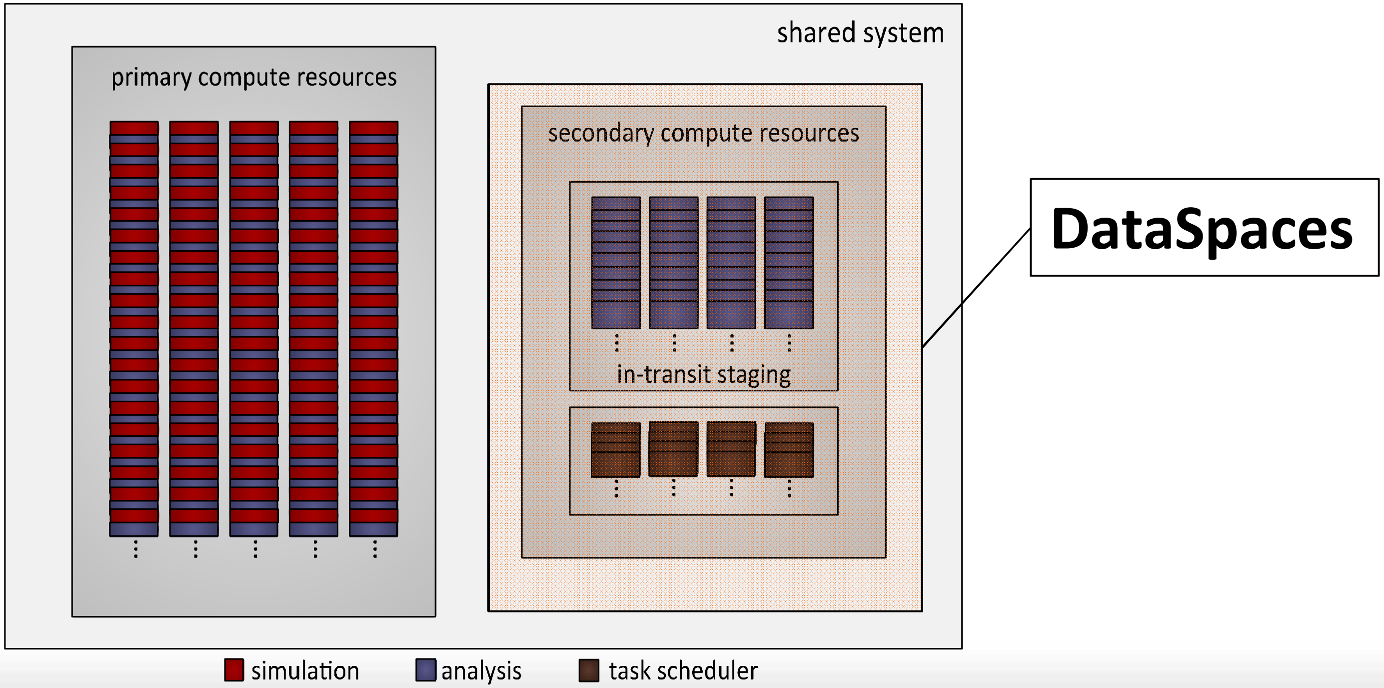
Volume Rendering



Topology

Simulation case study with S3D: Implementation details

- In-transit task scheduling is built off of DataSpaces
  - Distributed interaction and coordination service via shared space abstraction
  - Distributed design: Hashing used to balance RPC messages
  - Scalability obtained by mapping in-transit tasks onto separate compute nodes



Simulation case study with S3D: Implementation details

- Data movement enabled with DART
  - Asynchronous communication based on RDMA one-sided communication
  - Gemini interconnect (used by case study) provides user Generic Network Interface
  - Dynamically adapt between Fast Memory access (FMA) & Block Transfer Engine (BTE) based on message size
    - Small messages use SMSG that leverage FMA
      - Direct OS bypass to achieve low latency and high message rates
    - Large messages BTE memory operations are used
      - Achieve better communication/computation overlap
  - Transaction completion generates event notifications at both source and destination
- Both DART and DataSpaces are available in ADIOS

Additional information later today by Fan Zhang at the Doctoral Showcase from 3:30-5:00 in room 155-F

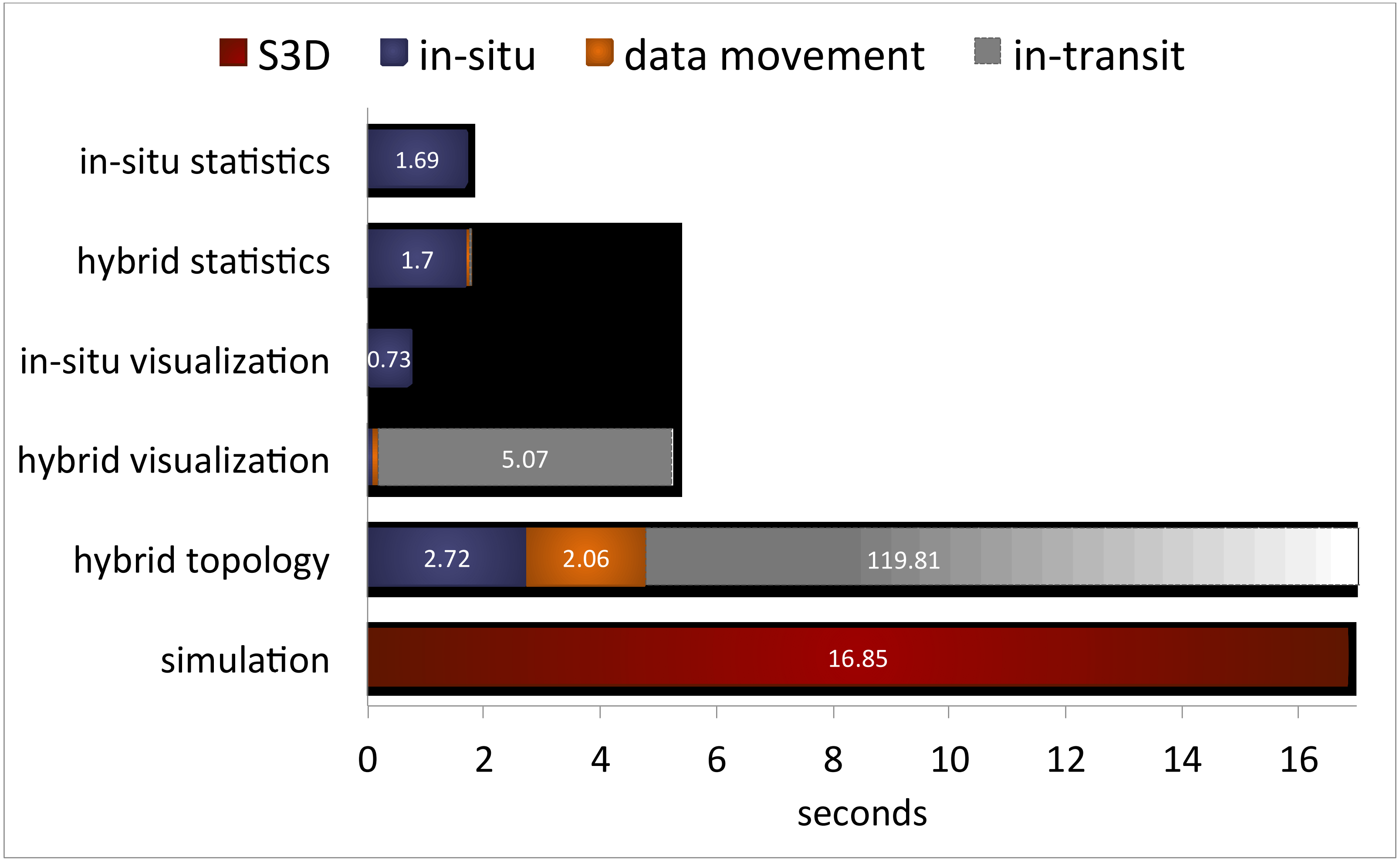
Simulation case study with S3D: Experimental set up

Jaguar, Cray XK6

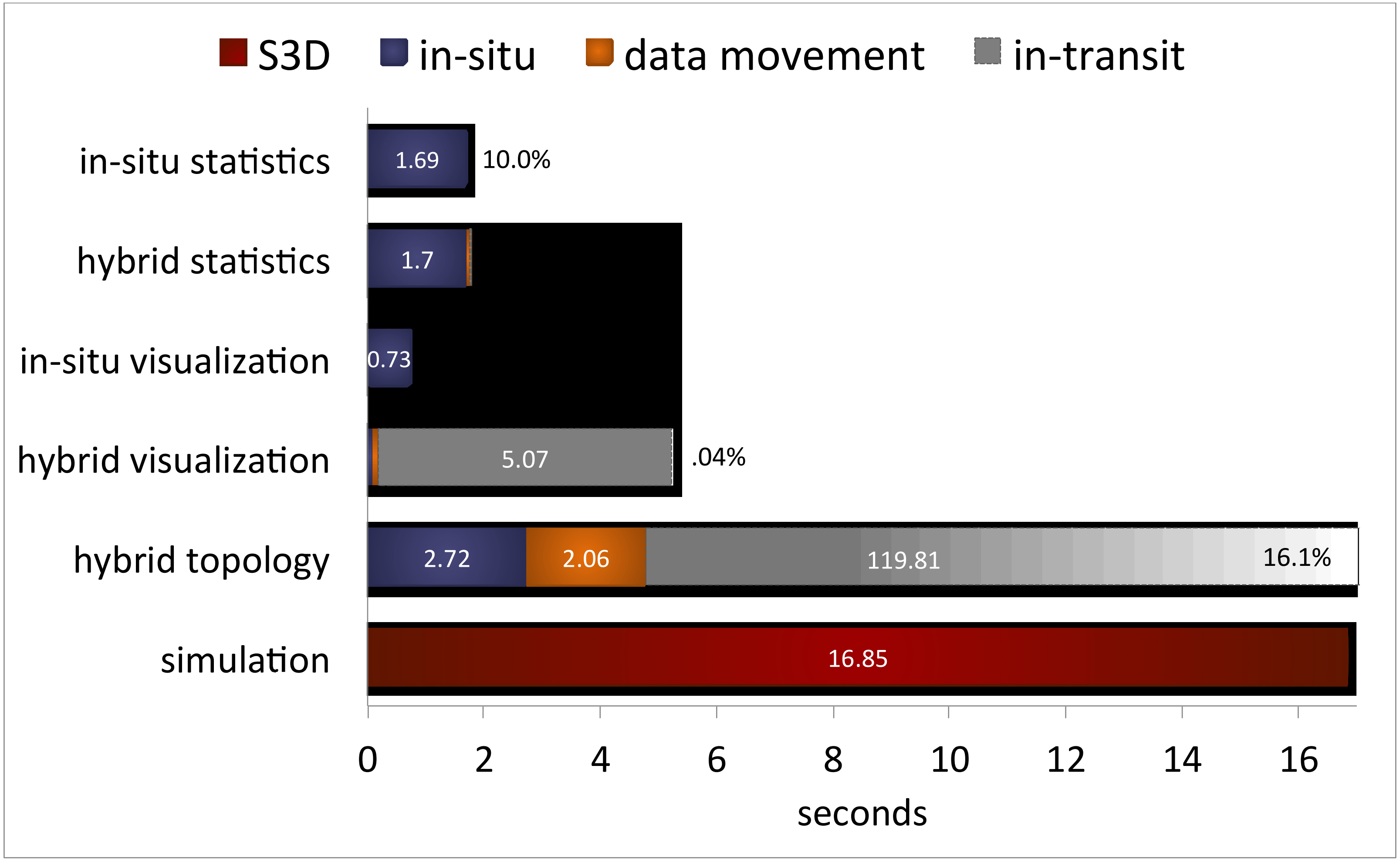
- 18,688 nodes
- Gemini interconnect
- 16-core AMD 6200 series Opteron processor
- 600 TB system memory

Volume size	1600x1372x430
# of variables	14
Variable size (bytes)	8
Data size (GB)	98.5
I/O read time (sec/time step)	6.56
I/O write time (sec./time step)	3.28
Simulation run time (sec./time step)	16.85
Analysis frequency	Set by scientist
Number of cores	4896
# simulation/in-situ cores	16x28x10=4480
# task scheduler cores	160
# in-transit cores	256

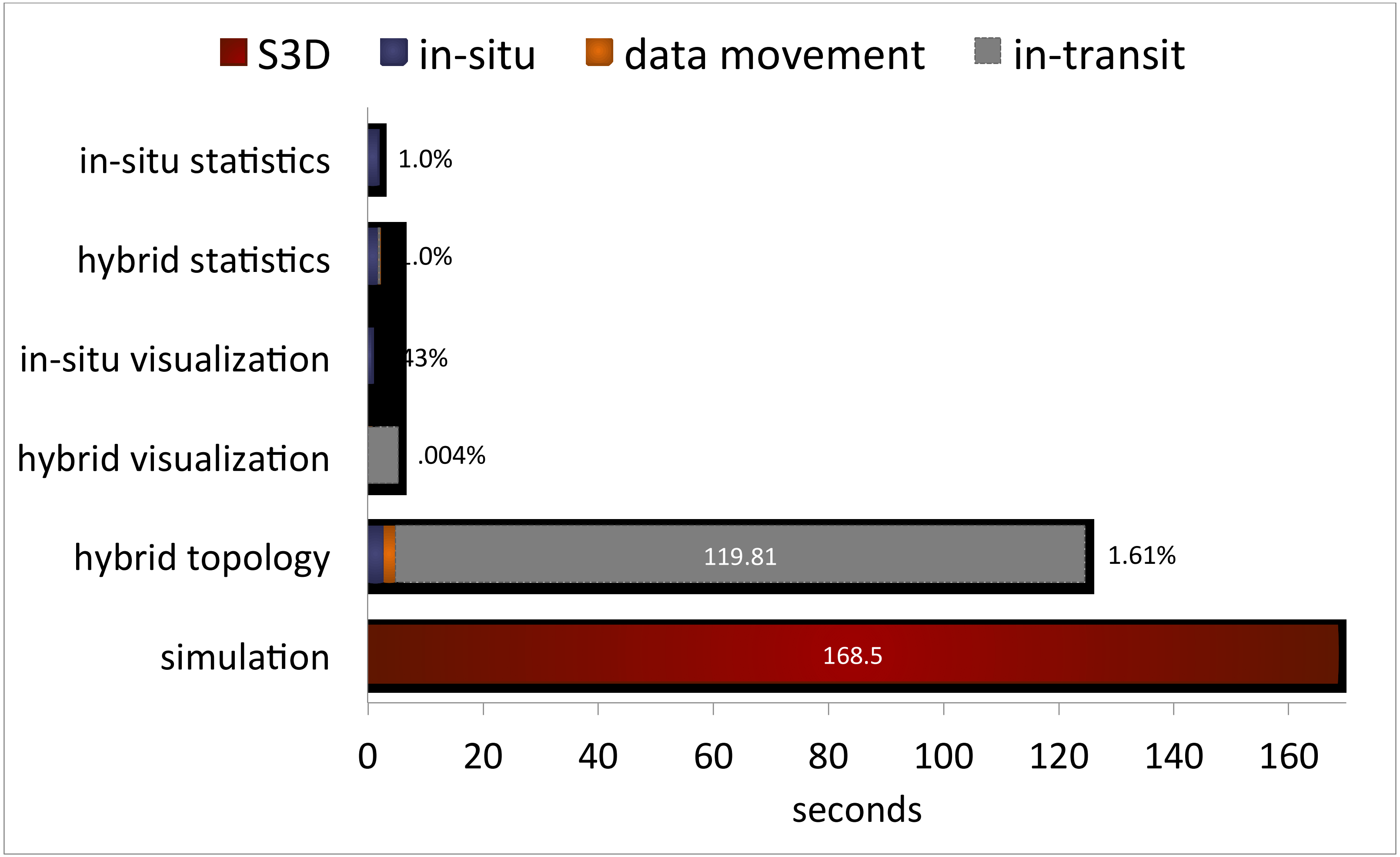
Simulation case study with S3D: Timing results for 4896 cores and analysis every simulation time step



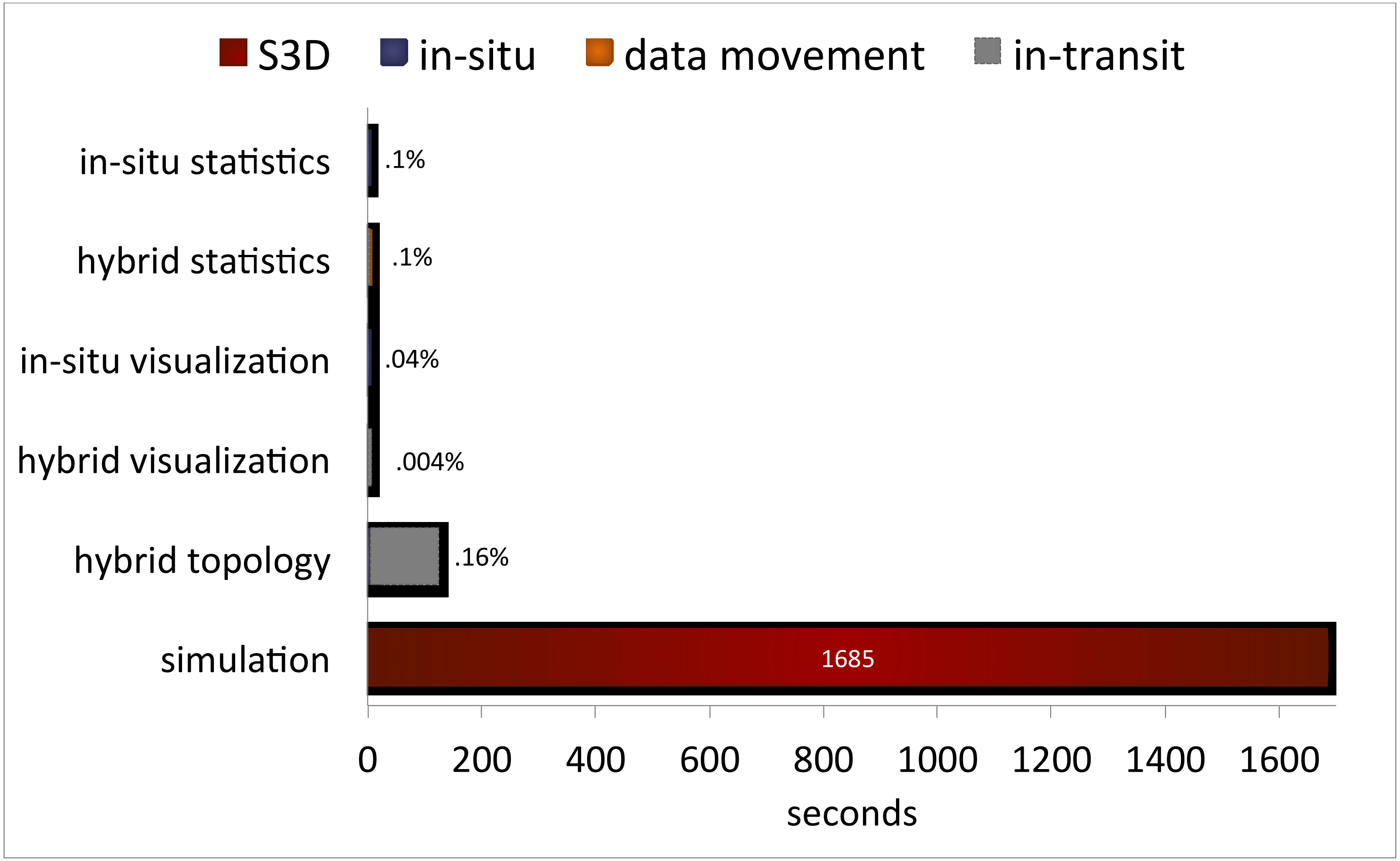
Simulation case study with S3D: Timing results for 4896 cores and analysis every simulation time step



Simulation case study with S3D: Timing results for 4896 cores and analysis every 10th simulation time step



Simulation case study with S3D: Timing results for 4896 cores and analysis every 100th simulation time step



Conclusion

- Exploring the design space of future workflows
  - We have developed a flexible data staging & coordination framework
    - Transparent transfer of data between primary & secondary compute resources
  - We have developed a temporally multiplexed approach
    - Decouple performance of analyses from that of the simulation by pipelining computations

- Investigating the impact of workflow design on analyses
  - We have developed new formulation of three common analyses
    - Massively parallel in-situ + serial in-transit stage

- We have performed a case study demonstrating analyses on large-scale turbulent combustion at high temporal frequencies

Future work

- Exploring the design space of future workflows
  - Thorough characterization of expected workflows
  - Projections on future architectures using simulators and modeling capabilities (SST)
- Investigating the impact of workflow design on analyses
  - Identify metrics required to characterize classes of analyses
  - Identify which classes of algorithms perform best under which workflow designs
  - Algorithmic shifts – subsampling with quantification of error
- Development of software stack to support workflows at extreme-scale
  - Solvers, data movement, data analysis, programming models, resilience, scheduling and run time systems all must work together

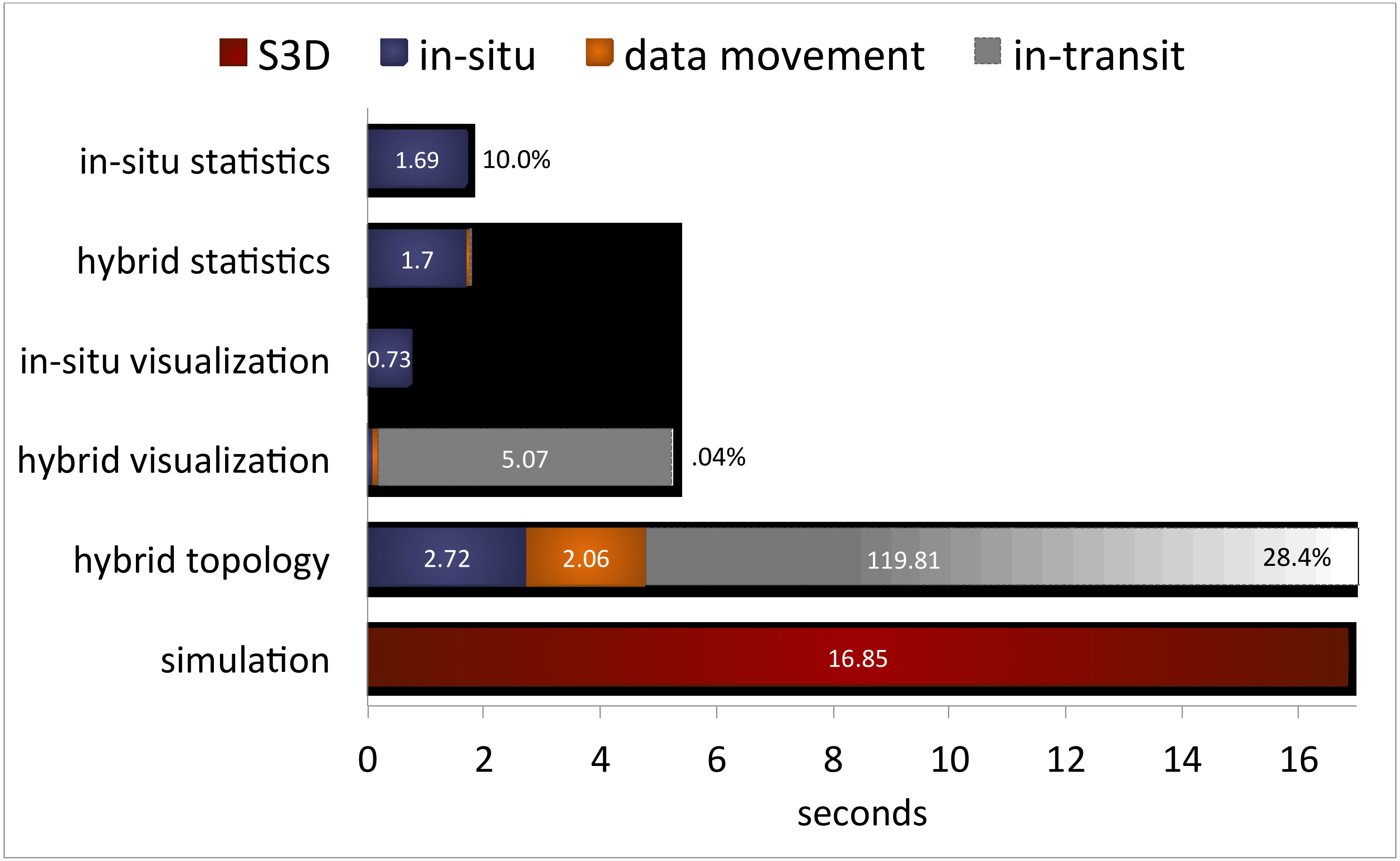
Questions?

Acknowledgements: DOE Office of Science, Advanced Scientific Computing Research

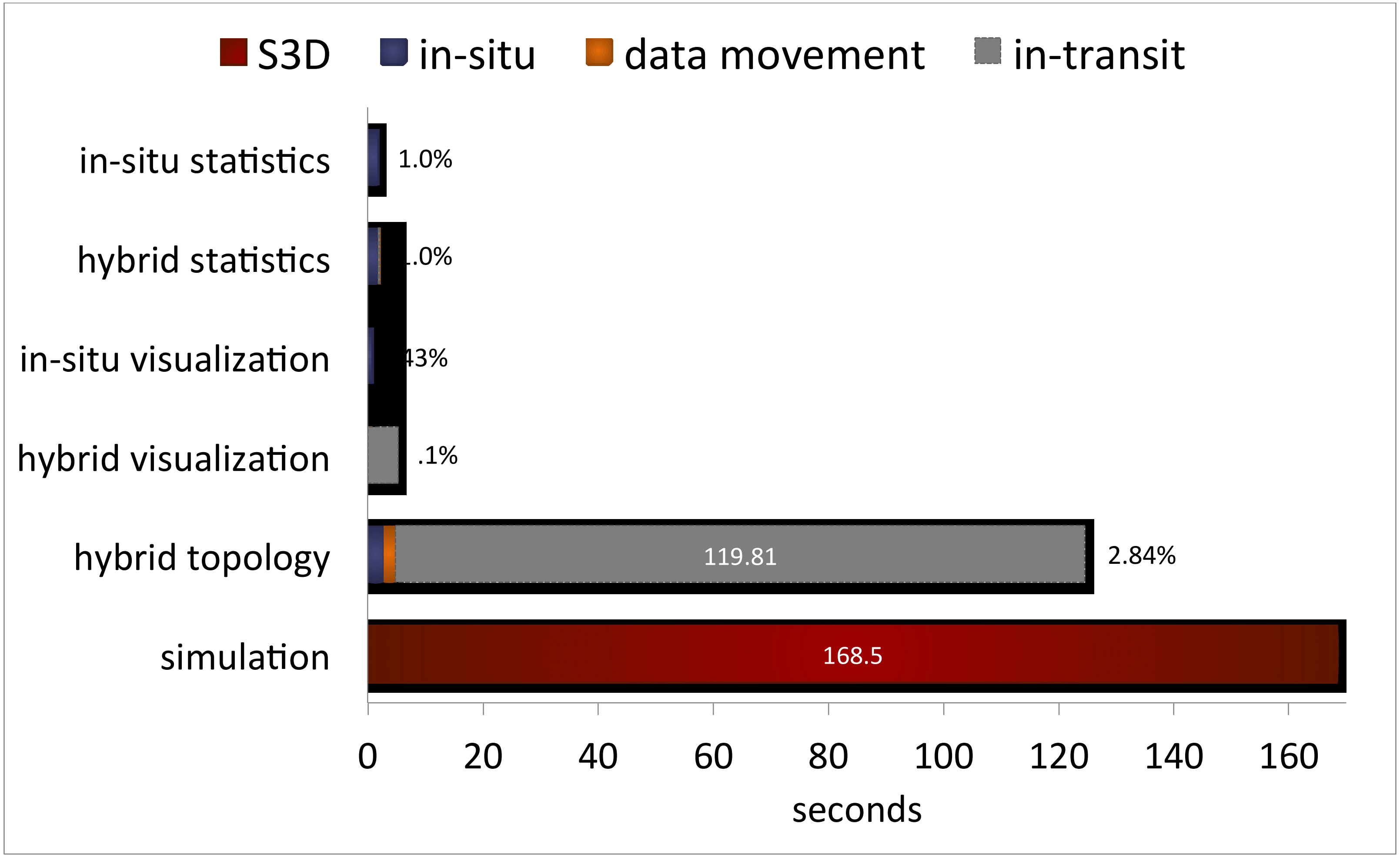
Contact: Janine Bennett, jcbenne@sandia.gov

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

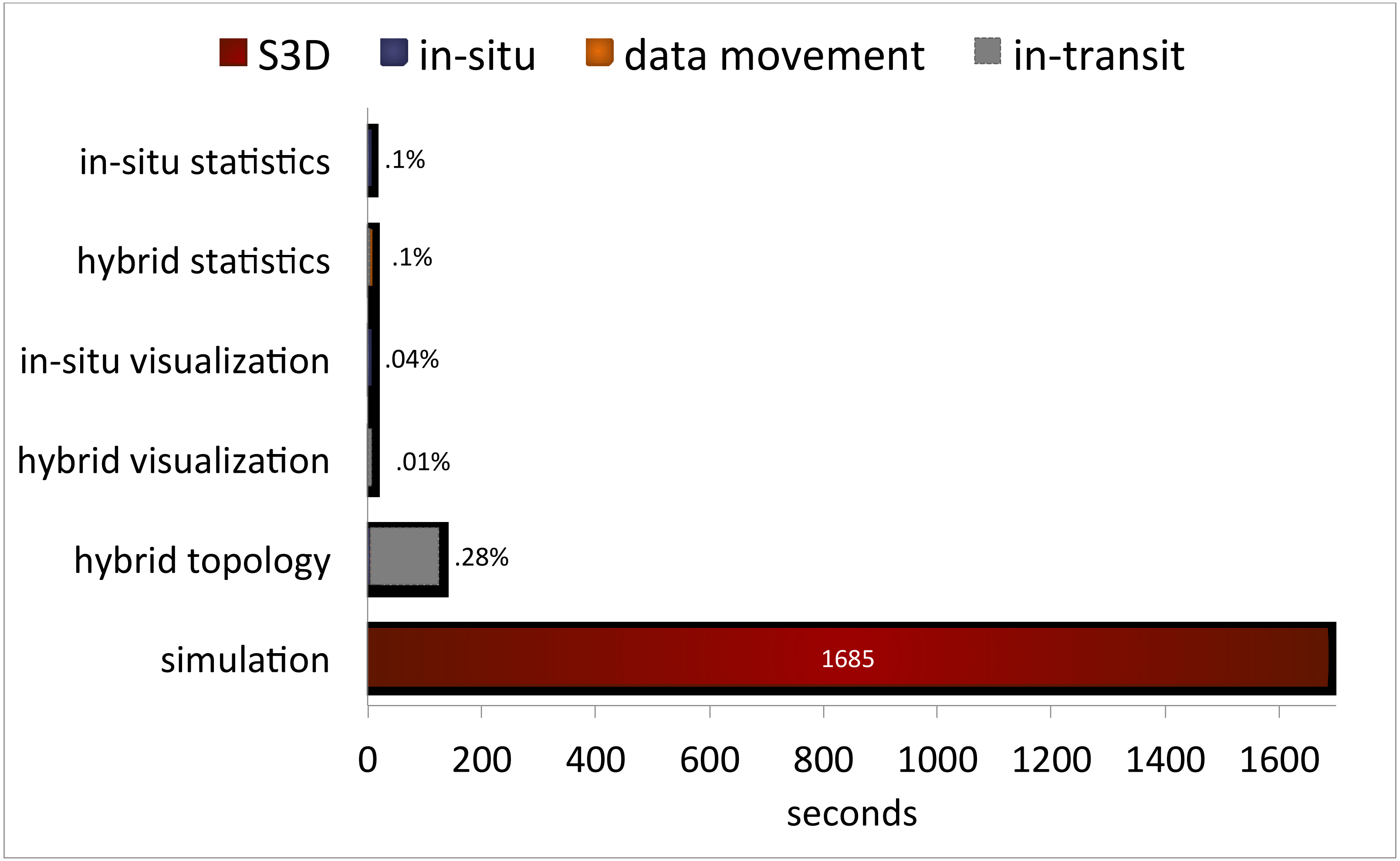
Simulation case study with S3D: Timing results for 4896 cores and analysis every simulation time step



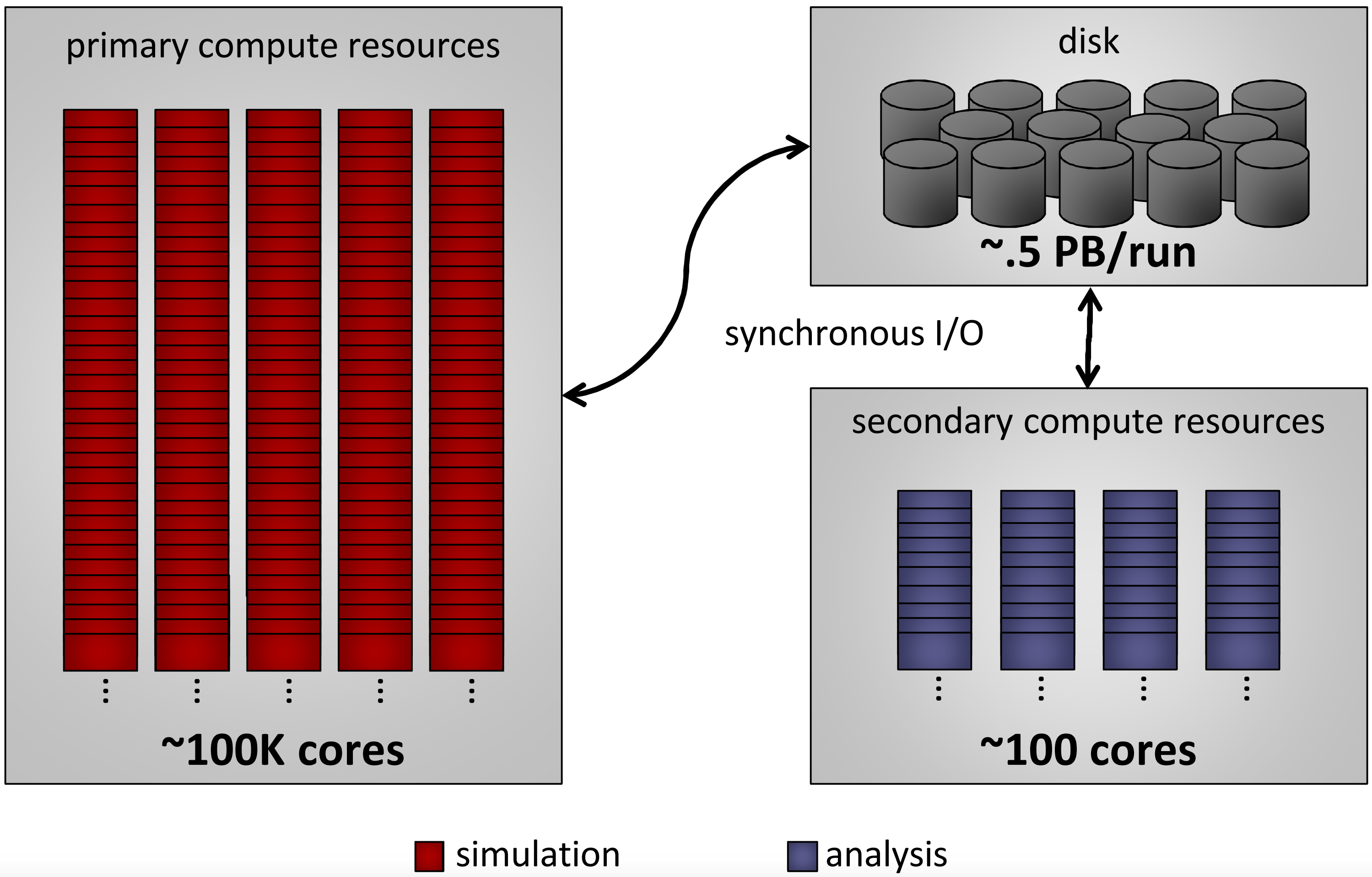
Simulation case study with S3D: Timing results for 4896 cores and analysis every 10th simulation time step



Simulation case study with S3D: Timing results for 4896 cores and analysis every 100th simulation time step



The current workflow of compute first, analyze later does not scale on projected high performance computing architectures



The current workflow of compute first, analyze later does not scale on projected high performance computing architectures