Sim-Situ

A Framework for the Faithful Simulation of in situ Processing

Valentin Honoré, Tu Mai Anh Do, Loïc Pottier, Rafael Ferreira da Silva, Ewa Deelman, Frédéric Suter

In situ Processing – Historical meaning

- Post-hoc

In situ Processing – Historical meaning

- Post-hoc

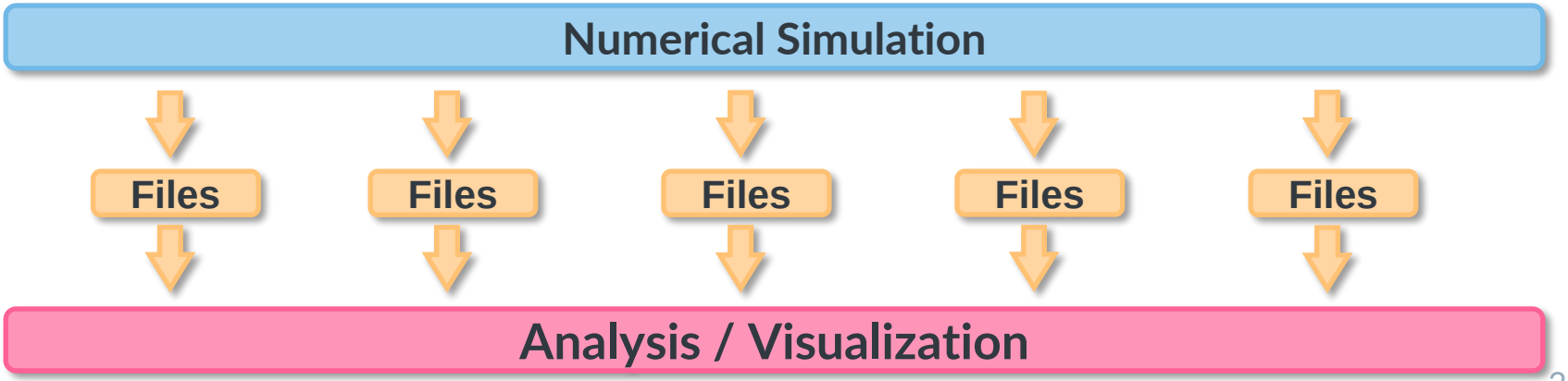
☹ Files becoming too big + Gap increase between CPU and I/O ☹

In situ Processing – Historical meaning

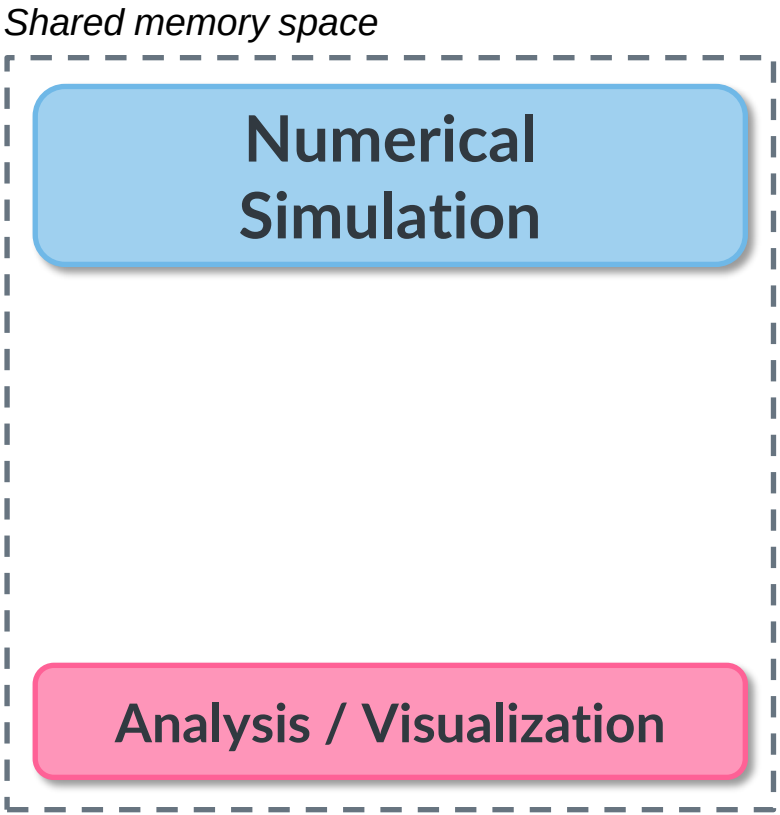
- Post-hoc

☹ Files becoming too big + Gap increase between CPU and I/O ☹

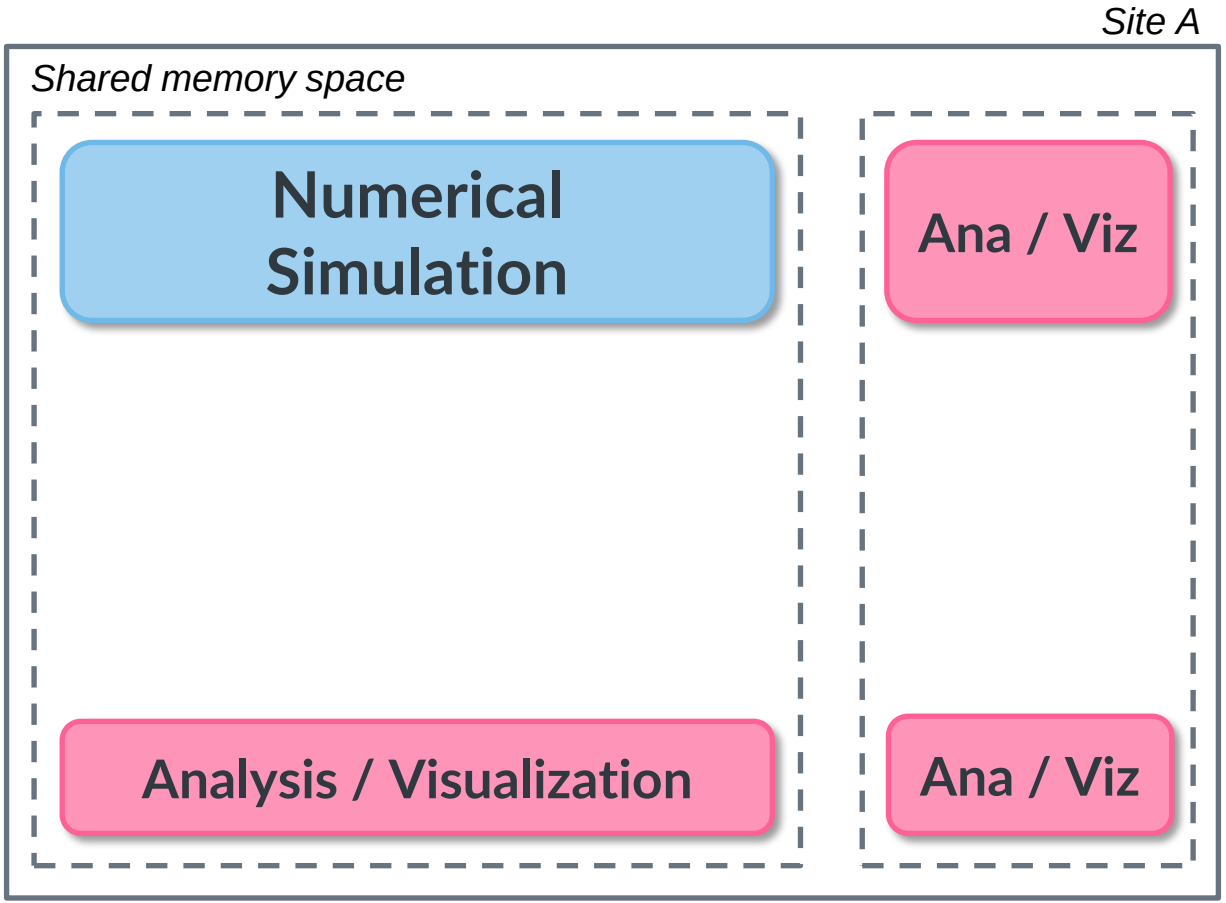
- In situ



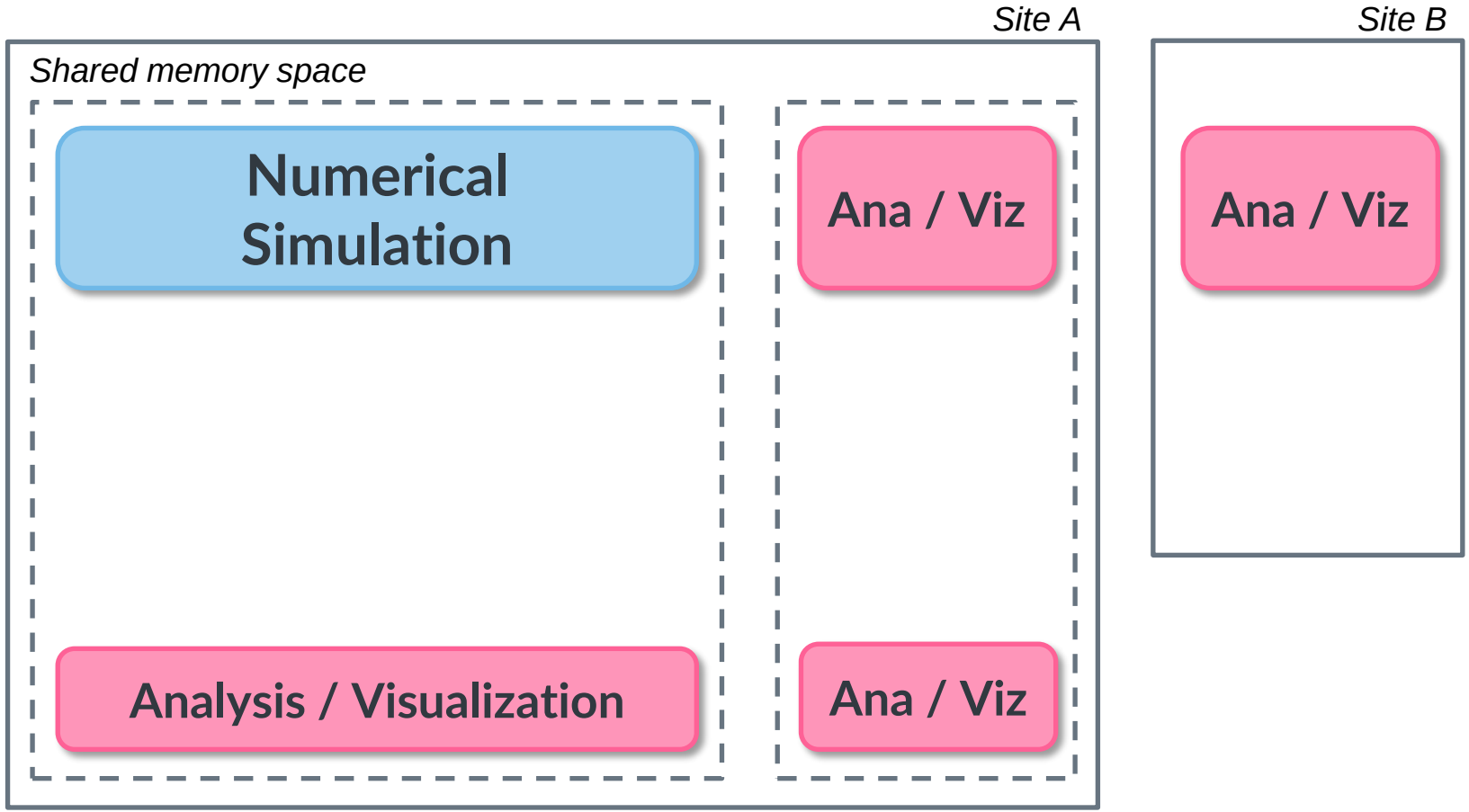
In situ Processing – Modern meaning



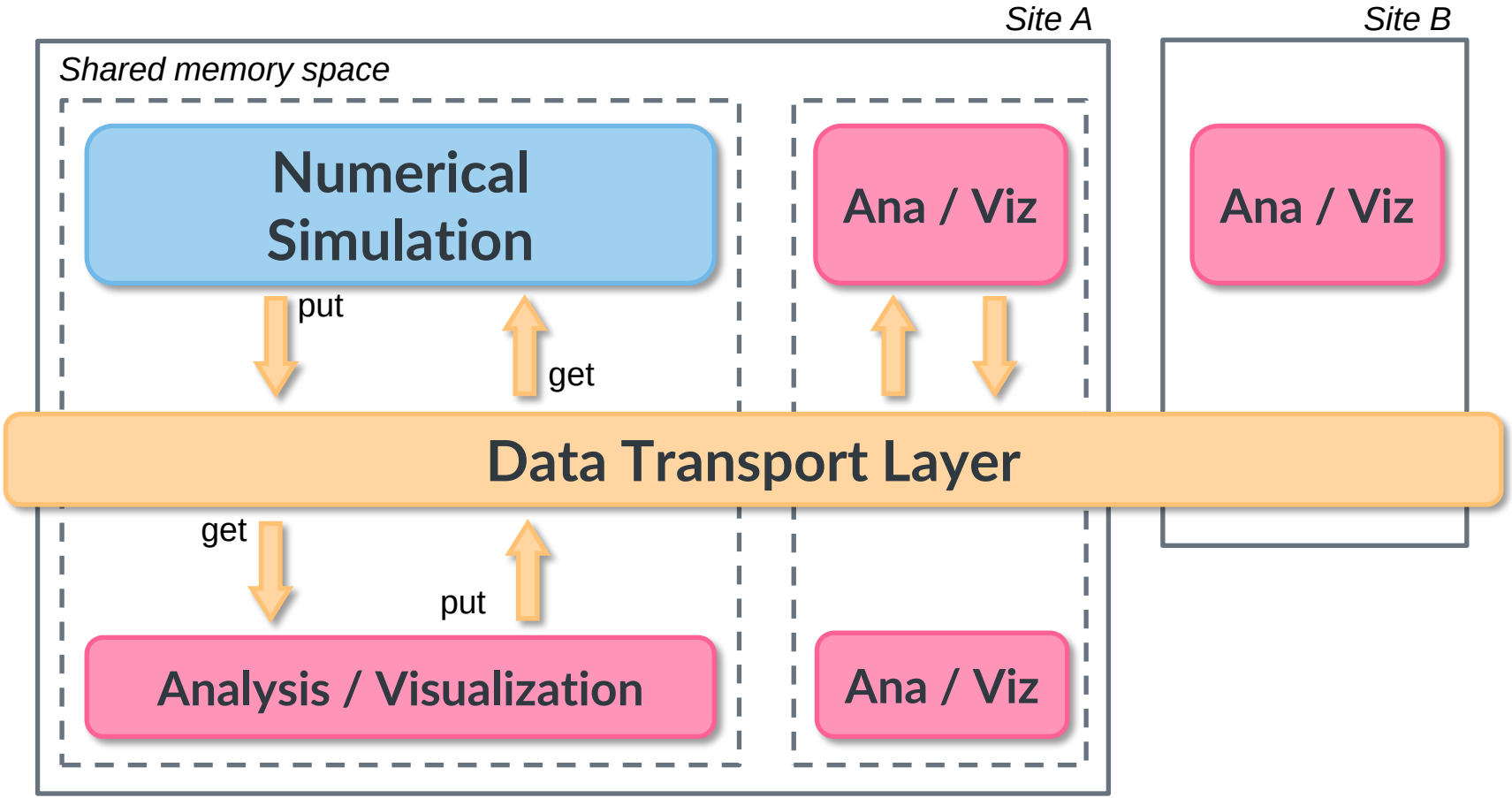
In situ Processing – Modern meaning



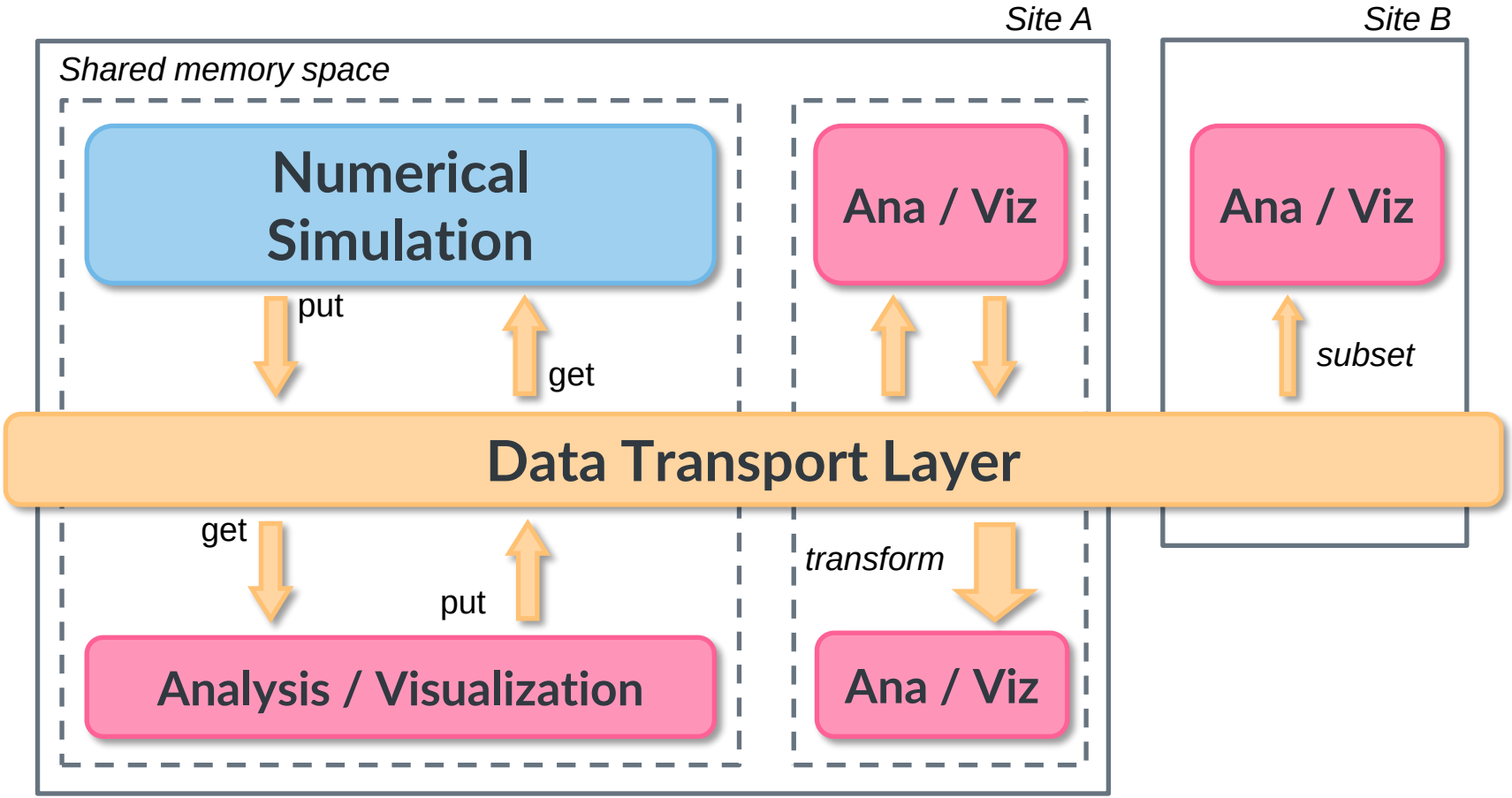
In situ Processing – Modern meaning



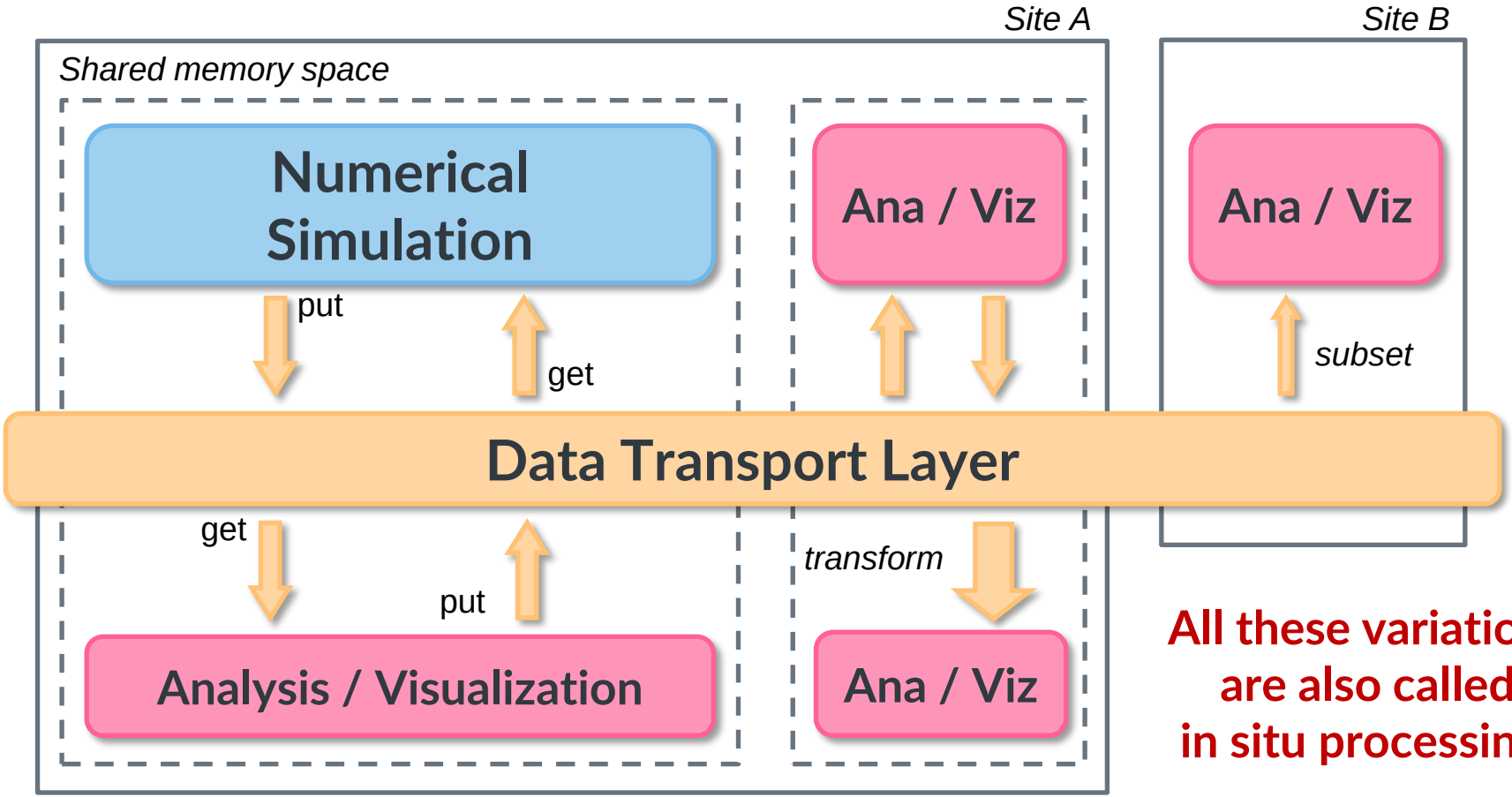
In situ Processing – Modern meaning



In situ Processing – Modern meaning



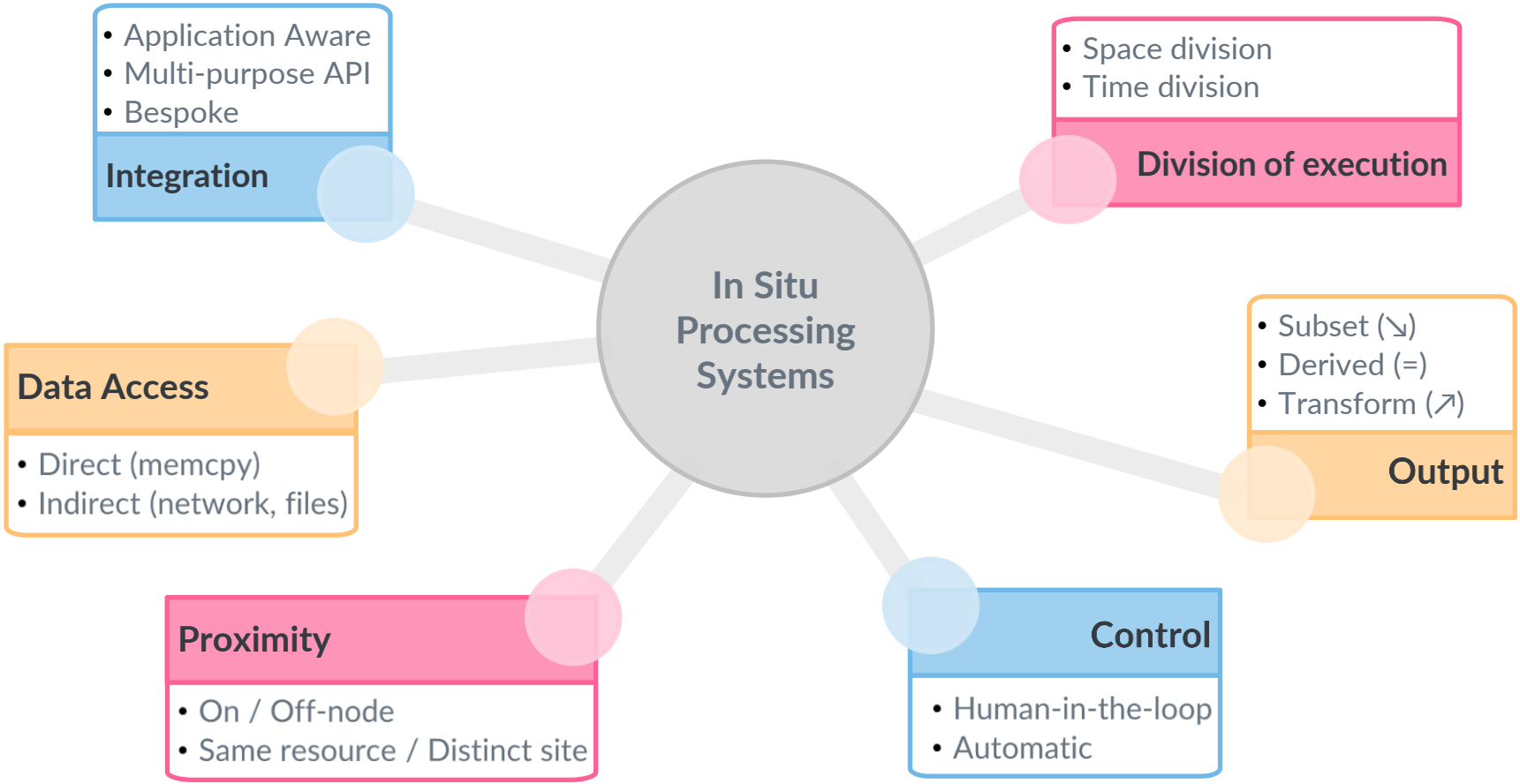
In situ Processing – Modern meaning



All these variations are also called in situ processing!

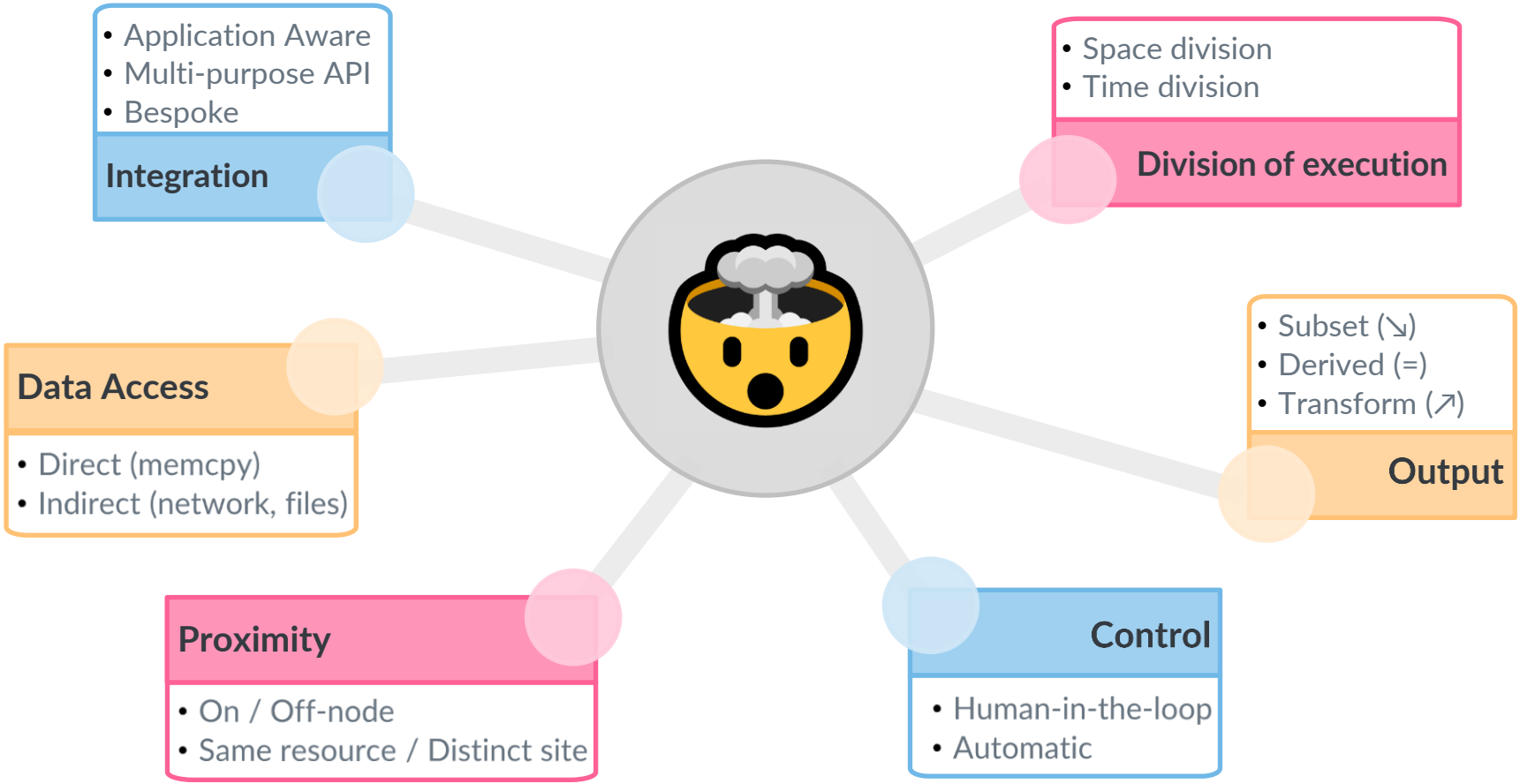
In situ Processing – Terminology

H. Childs, et al. (2020)
IJHPCA 34:6, 676–691



In situ Processing – Terminology

H. Childs, et al. (2020)
IJHPCA 34:6, 676–691



In situ Processing – Research Questions

RQ1 **Allocation**

How much resources give to simulation, analysis, viz?

RQ2 **Mapping**

Where and when run the analysis/visualization?

RQ3 **Data Transport**

Are files out of question? In-memory or over network?

RQ4 **Scalability**

Will choices remain the best one at scale?

RQ5 **Optimization**

Multiple options, not always completion time

Why Sim-Situ?

- Answer RQs → **Performance evaluation**
  - Take good decisions → **Objective performance indicators**
- **Go beyond the traditional empiric guess** → **Speed and Flexibility**
  - Explore many unconventional scenarios
  - Consider unconventional performance metrics

Experiments

→ Time- and resource-consuming
→ Complex to set up
→ Limited in scope
→ Sensitive to exogenous factors

Simulation

→ Run on a laptop
→ Highly flexible
→ What-if scenarios
→ Reproducibility and control

Sim-Situ Architecture

Fundation – SimGrid

https://simgrid.org

A scientific instrument on your laptop

- Open Project since 1998
- Key strengths
  - **Usability:** Fast, Reliable, User-oriented APIs
  - **Validated performance models:** Open Science → Predictive Power
  - **Versatility:** HPC, Cloud, Fog, Grid, P2P, …

Fundation – SimGrid

https://simgrid.org

A scientific instrument on your laptop

- Open Project since 1998
- Key strengths
  - **Usability:** Fast, Reliable, User-oriented APIs
  - **Validated performance models:** Open Science → Predictive Power
  - **Versatility:** HPC, Cloud, Fog, Grid, P2P, …
- SimGrid's fundamental concepts (the **S4U** API)

Actors

Execute user-provided functions

Program anything you want/need

Fundation – SimGrid

https://simgrid.org

A scientific instrument on your laptop

- Open Project since 1998
- Key strengths
  - **Usability:** Fast, Reliable, User-oriented APIs
  - **Validated performance models:** Open Science → Predictive Power
  - **Versatility:** HPC, Cloud, Fog, Grid, P2P, …
- SimGrid's fundamental concepts (the **S4U** API)

Actors	Activities
Execute user-provided functions	Computation, communication, I/O
Program anything you want/need	Synchro mechanisms

Fundation – SimGrid

https://simgrid.org

A scientific instrument on your laptop

- Open Project since 1998
- Key strengths
  - **Usability:** Fast, Reliable, User-oriented APIs
  - **Validated performance models:** Open Science → Predictive Power
  - **Versatility:** HPC, Cloud, Fog, Grid, P2P, …
- SimGrid's fundamental concepts (the **S4U** API)

Actors	Activities	Resources
Execute user-provided functions	Computation, communication, I/O	CPUs, Links, Disks
Program anything you want/need	Synchro mechanisms	Hosts, VMs, Netzones, …

Fundation – SimGrid

https://simgrid.org

A scientific instrument on your laptop

- Open Project since 1998
- Key strengths
  - **Usability:** Fast, Reliable, User-oriented APIs
  - **Validated performance models:** Open Science → Predictive Power
  - **Versatility:** HPC, Cloud, Fog, Grid, P2P, …
- SimGrid's fundamental concepts (the **S4U** API)

Actors	Activities	Resources
Execute user-provided functions	Computation, communication, I/O	CPUs, Links, Disks
Program anything you want/need	Synchro mechanisms	Hosts, VMs, Netzones, …
Mailboxes	Rendez-vous points between actors	

Faithful Simulation – SMPI

Numerical Simulation

- **Have faith: do not modify anything, run your application!**
- (Sim/em)ulation of (unmodified) MPI applications
  - Support of (the essential of) MPI-3.1
  - **Collective operations:** Borrowed selection logic of popular runtimes
  - **Coupling:** With S4U codes or multiple MPI codes

```
$ smpicc source.c -o application  # The code is now compiled
$ smpirun -platform cluster.xml -hostfile hostfile.txt  ./application # It starts
[...]  # Some debug information about your data provenance
Got 42 from rank 0
```

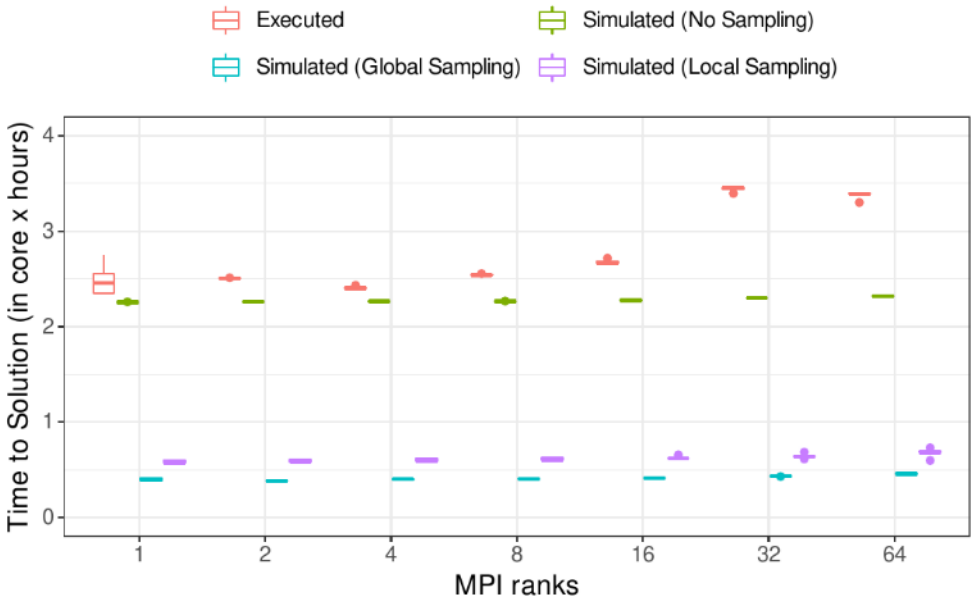
- **Integration testing**
  - Compile and run 100+ proxy-apps every night
  - C, C++, F77, F90, kokkos, and some OpenMP

https://framagit.org/simgrid/SMPI-proxy-apps/

Faster to run or simulate?

Numerical Simulation

- **Running** the application
  - Acquire cores /nodes
  - Run in parallel
  - Less predictable
- **Simulating** the application
  - Use your laptop
  - Fold execution on one core
  - Reproducible



Use **sampling** to speed up the simulation (**x5**)

- By just adding a SMPI macro to one function call

Flexible data management

Data Transport Layer

- Inspired by popular data management frameworks (ADIOS, DataSpaces)
  - Client/Server library
  - Put/Get mechanism
  - Self-descriptive variables
  - Configurable data transport method

Flexible data management

Data Transport Layer

- Inspired by popular data management frameworks (ADIOS, DataSpaces)
  - Client/Server library
  - Put/Get mechanism
  - Self-descriptive variables
  - Configurable data transport method

- Current implementation status (**v0.1**)
  - **Dynamic** client connection/deconnection
  - Multiple **data channels**
  - **Multi-dimensional** arrays
  - **Synchronous/asynchronous** transfers
  - Two transport methods (**memcpy** and **network**)

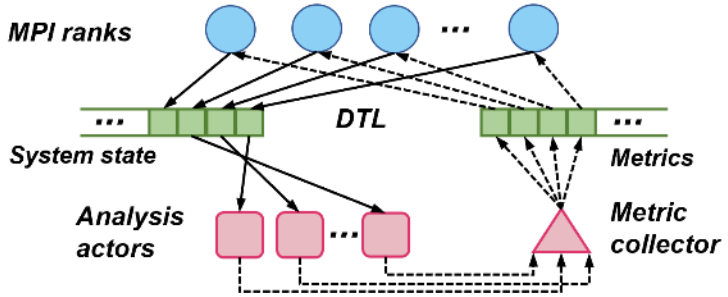
Customizable Analysis Actors

Analysis Visualization

- **Be flexible: use simple abstractions for analysis**
  - Simple amount of work, complex cost models, small program, …
  - Connect them at will in an **analysis workflow**

- Basic structure of an analysis actor
  1. Connect to the DTL on a specific channel
  2. Get some variables
  3. Perform the analysis
  4. Put some new variable in the DTL (optional)

- **Sim-Situ** provides **stock implementations**
  - **Distributed** analysis
  - **Parallel** analysis
  - Data **aggregator**



Sim-Situ in Practice

- Add customizable in-situ processing to an MPI application
  - **Slightly modify** the application
    - Handling extra configuration parameters — ~10 lines
    - Interacting with the DTL — ~20 lines
    - Compile against SMPI and Sim-Situ — ~5 lines
  - Define your analysis workflow(s) — Configurable
    - Define **actor behaviors** and use **data channels** to compose them
  - Add 3 extra files — Configurable
    - XML: Target **simulated infrastructure**
    - Text: Analysis actors **mapping** and MPI **hostfile**
- Define your experimental scenarios
- Run them!

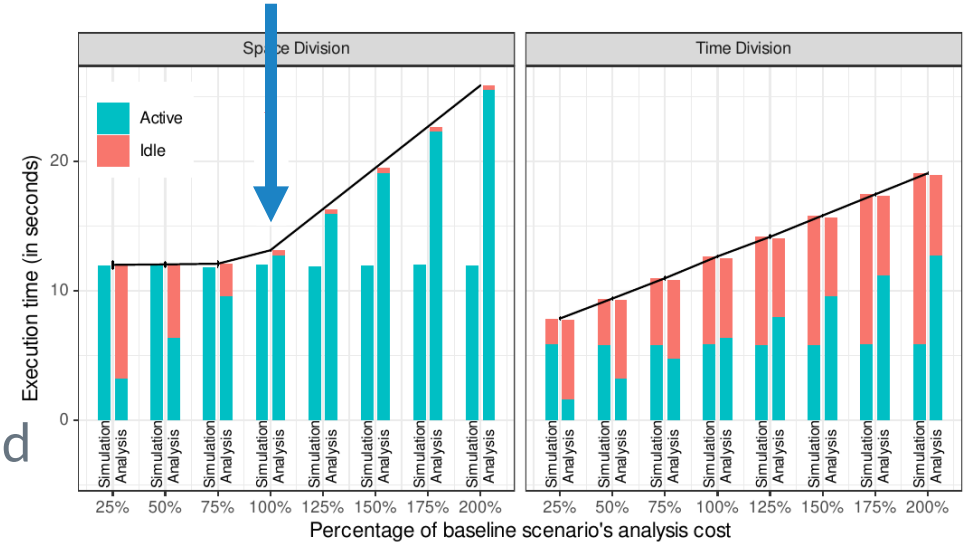
Check our experimental artifact *https://doi.org/10.6084/m9.figshare.20416008.v1*

Use Cases

Time or Space Division of Execution?

RQ1
RQ2
RQ5

- **Hypothesis:** Existing balanced scenario with space division
- **What-if** analysis cost changes?

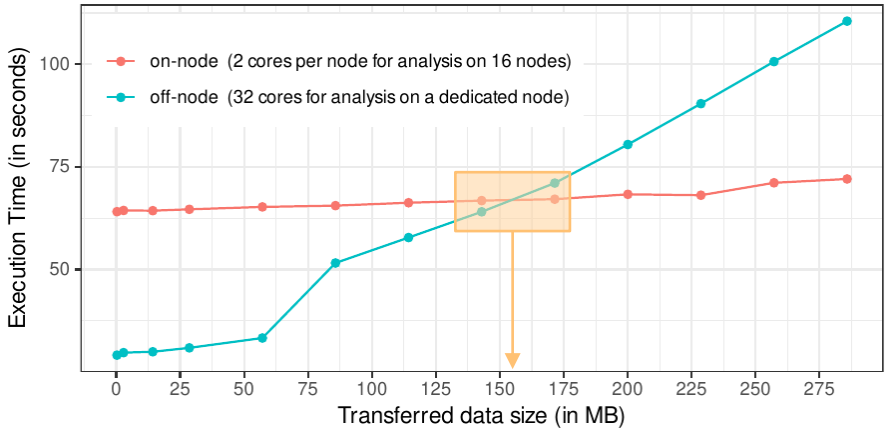


- Space Division
  - No impact on duration of numerical simulation
  - No idle time
  - Analysis results may be delayed
- Time Division
  - More resource for numerical simulation
  - … but has to wait for the analysis to proceed
  - Completion of numerical simulation may be delayed

On-node or Off-node?

RQ2
RQ3
RQ4

- **Hypothesis:** analysis time increases when distributed across more nodes
- **On-node**
  - Simulation and analyses share nodes (30/2 ratio on 16 nodes)
  - Direct access (memcopy)
  - Distributed analysis
- **Off-node**
  - 15 nodes for simulation + 1 node for analysis
  - No distribution of analysis
  - Indirect access (network)



- Measure the impact of size of staged data on execution time
  - With only 2 host files and a variable configuration parameter

Conclusion and Future work

- **Sim-Situ:** a simulation-based framework to study **in-situ processing**

Numerical Simulation

Faithful (Sim/Em)ulation of the unmodified application
Keep the exact execution pattern

Data Transport Layer

Flexible data management with a simple API (Put/Get) and a Pub/Sub model

Analysis/Visualization

Customizable and composable abstracted actors to enable complex workflows

A configurable tool to go beyond the traditional empiric guesses!

- **Future work**

More applications
From proxy-apps to full scale

More stock implementations
Models of visualization routines
Complex analysis workflows

More transport methods
Capture behavior of advanced algorithms

Thanks!

Any questions?

Sim-Situ

A Framework for the Faithful Simulation of in situ Processing

Valentin Honoré, Tu Mai Anh Do, Loïc Pottier, Rafael Ferreira da Silva, Ewa Deelman, Frédéric Suter

This work was partially sponsored by the Laboratory Directed Research and Development Program of the Oak Ridge National Laboratory, managed by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy.