

SMPI Courseware: Teaching Distributed-Memory Computing With MPI in Simulation

HENRI CASANOVA¹ ARNAUD LEGRAND ²
MARTIN QUINSON ³ FRÉDÉRIC SUTER ⁴

¹ University of Hawai`i at Manoa, USA

² Université Grenoble Alpes, CNRS, France

³ ENS Rennes, France

⁴ IN2P3 Computing Center, France

SCOPE

- Teaching distributed-memory programming with MPI is often an important part of an HPC course
- Well-known and important learning objectives
 - SPMD, data distribution, basic MPI proficiency
 - Point-to-point and collective communications
 - Performance bottleneck analysis
 - Communication/communication overlap
 - ...
- Typical approach: provide students access to some HPC platform
 - Often a commodity cluster
- This approach comes with challenges...

REAL PLATFORMS: LOGISTICAL CHALLENGES (1)

- Logistical Challenge: An institution may not have an HPC platform
 - Or none readily available for teaching purposes
- There are several solutions:
 - Build a low-cost platform (e.g., raspberry pies, clusters of SoCs)
 - Use virtualization/container technology (e.g., on some cloud)
 - Use some public testbed if available
- These solutions can have limitations that impede some SLOs
 - Limited scale, performance behavior not representative of production HPC clusters
- But let's say access to a representative HPC platform is secured...

REAL PLATFORMS: LOGISTICAL CHALLENGES (2)

- Logistical Challenge: Production platforms are shared among users that compete for resources
- This can create problems when novice students are let loose
 - Many platforms can be brought down surprisingly easily by buggy programs
- Even assuming a platform that is dedicated to students in a course, competition among students can cause problems
 - e.g., hog the batch queue before an assignment deadline
- Our own experience teaching HPC provides anecdotal evidence of the above
 - Were we just unlucky?
- But even if the above is not an issue, there are pedagogic challenges when using a real platform

REAL PLATFORMS: PEDAGOGIC CHALLENGES (1)

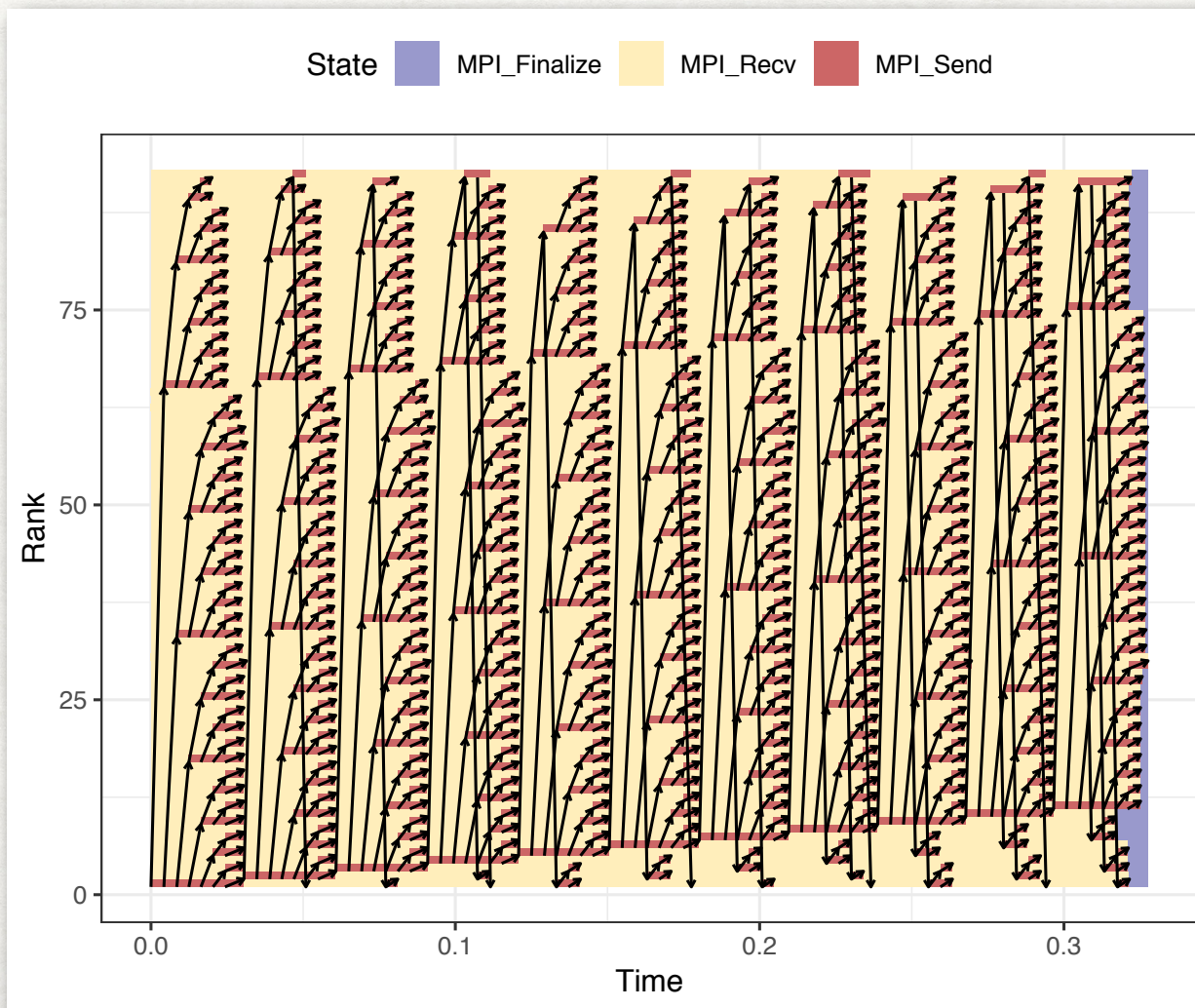
- Challenge: students' hands-on learning opportunities are limited to platform configurations at hand
 - Perhaps just one
- This is a problem for performance-oriented SLOs
- Instructors can find themselves saying things like:
 - “If we had more compute nodes, then this would happen...”
 - “If the network topology were different, then this wouldn't work as well...”
 - “If the network latency were lower, then this algorithm would be great...”
 - “If nodes were heterogeneous then this algorithm would be useful”
- This is not the most effective way for students to acquire knowledge

REAL PLATFORMS: PEDAGOGIC CHALLENGES (2)

- Challenge: real platforms come with “noise”
- Program executions are subject to idiosyncratic deterministic and non-deterministic software and hardware effects
 - Reproducing MPI benchmarking is hard [Hunold et al. 2016]
 - Neighbor jobs impact network [Bathele et al. 2013]
 - Manufacturing variability [Inadomi et al. 2015]
 - Machine learning techniques proposed to deal with “noisy” performance measurements [Tuncer et al. 2018]
- This is very easy to see with simple MPI programs...

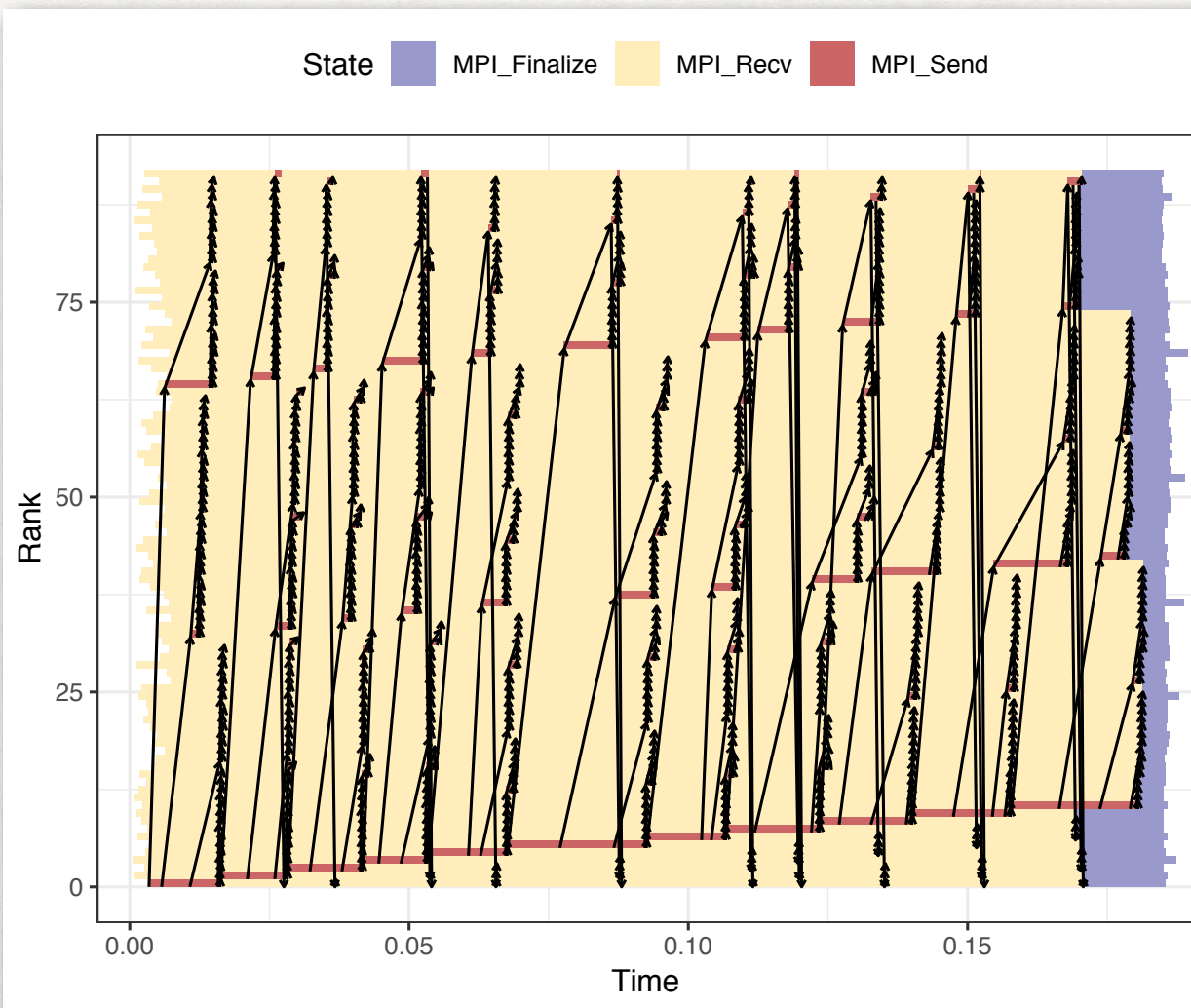
REAL PLATFORMS: PEDAGOGIC CHALLENGES (3)

- What 11 MPI_Bcast in a row should look like



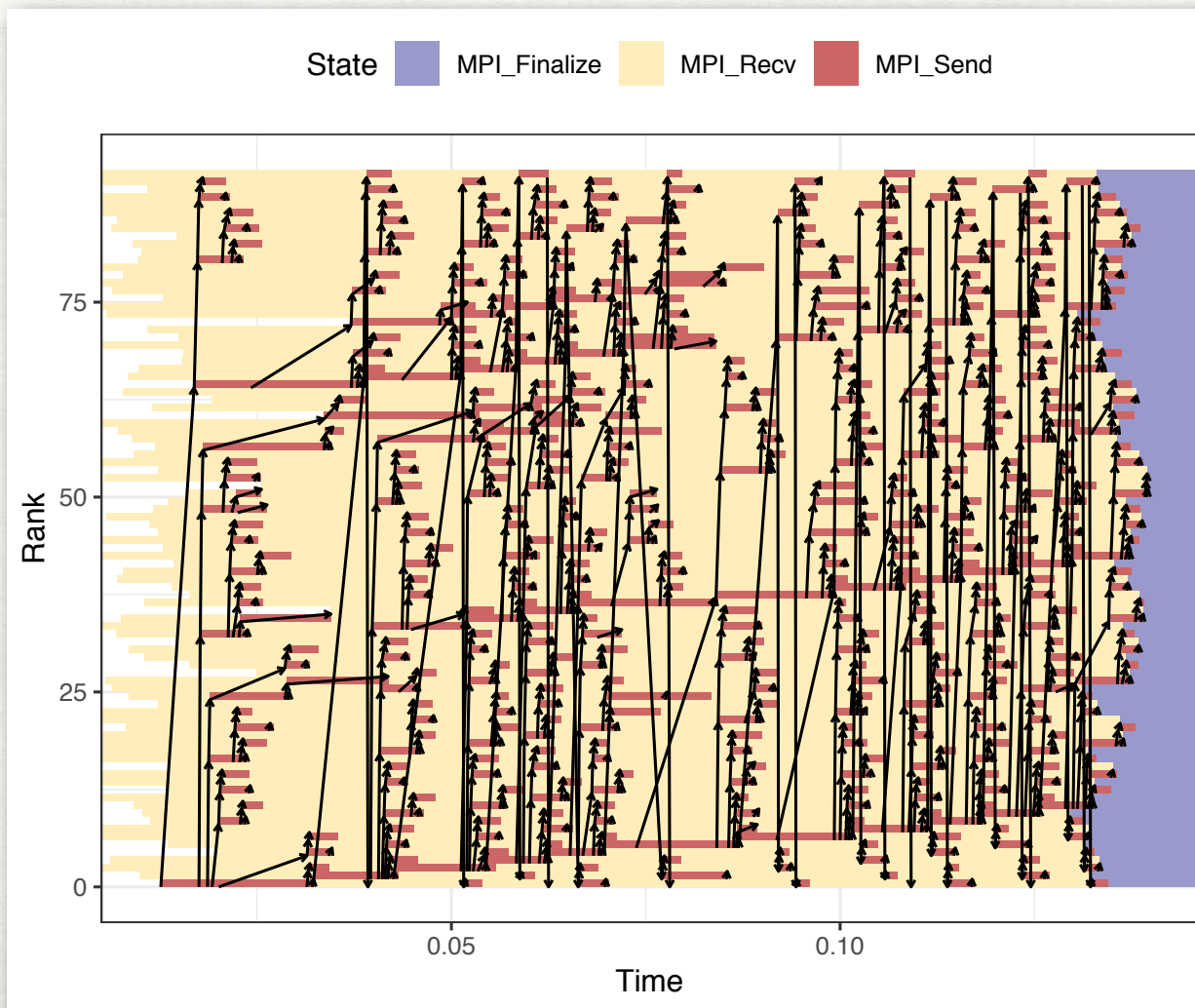
REAL PLATFORMS: PEDAGOGIC CHALLENGES (4)

- What 11 MPI_Bcast in a row look like on a real cluster (it will be different next time)



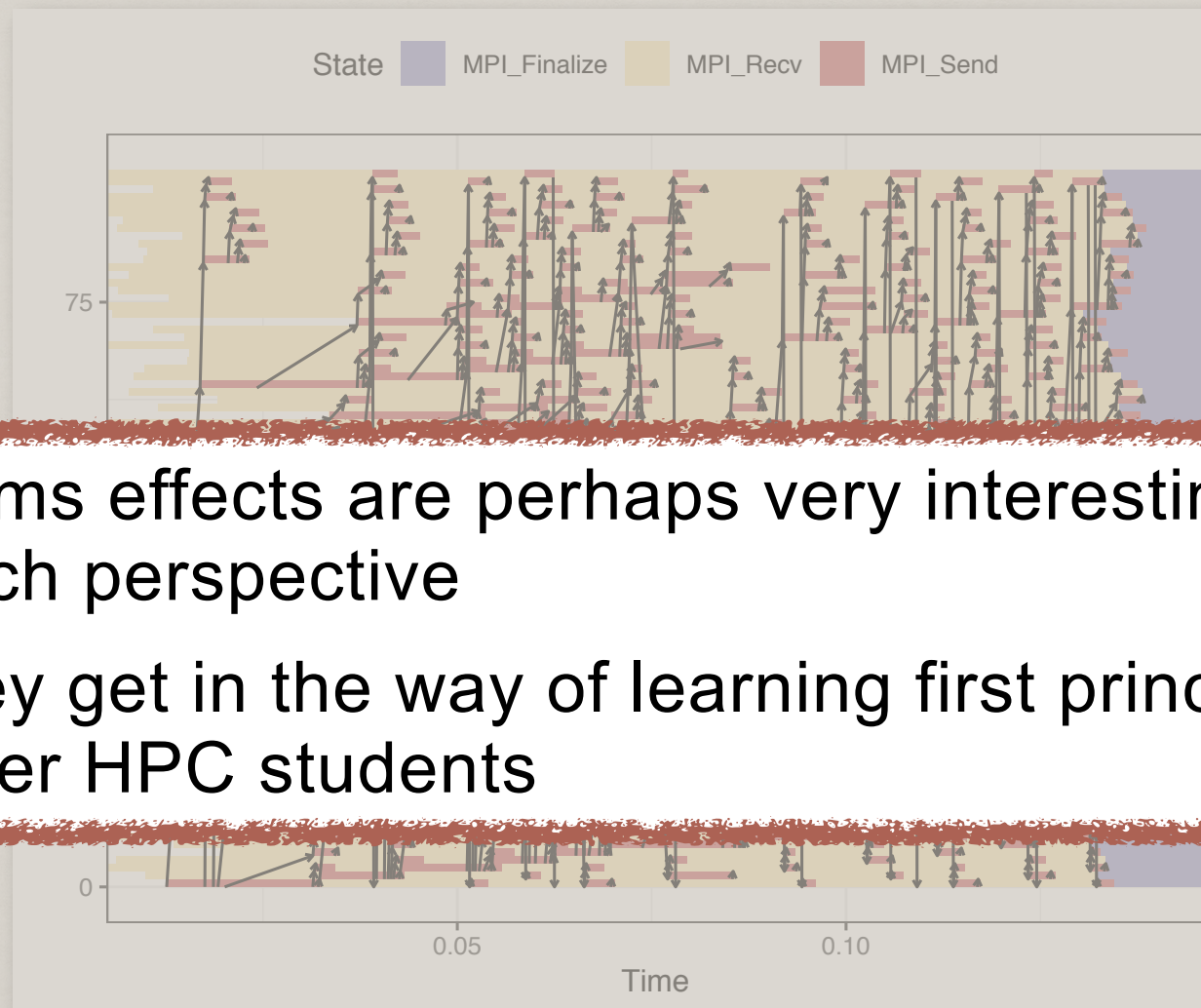
REAL PLATFORMS: PEDAGOGIC CHALLENGES (5)

- What 11 MPI_Bcast in a row look like on a single multicore-laptop



REAL PLATFORMS: PEDAGOGIC CHALLENGES (5)

- What 11 MPI_Bcast in a row look like on a single multicore-laptop



- Platforms effects are perhaps very interesting from a research perspective
- But they get in the way of learning first principles for beginner HPC students

SIMULATION FOR TEACHING

- An approach to solve all these (and other) challenges is simulation
 - No need for an actual platform, arbitrary platform configurations, perfect repeatability, quick executions
 - Used routinely for teaching in some areas of Computer Science (architecture, networks)
- This is not a new idea:
 - Started in the early 1990s
 - More recent works as well:
 - Simulation-based curriculum for HPC M.S. degree [Zarza et al. 2012]
 - ParaLab [Kozinov et al. 2015]

OUR PROPOSED SIMULATION APPROACH

- We propose to teach distributed-memory computing and MPI programming in simulation
- Compelling advantages over previous work:
 - Everything can be done on a student's laptop, even large-scale simulations
 - Students write standard MPI programs and can debug them in simulation mode
 - Arbitrary platform configurations (cores, hosts, clusters, switches, backbones, WANs, ...)
- All this is possible thanks to SMPI

SMPI



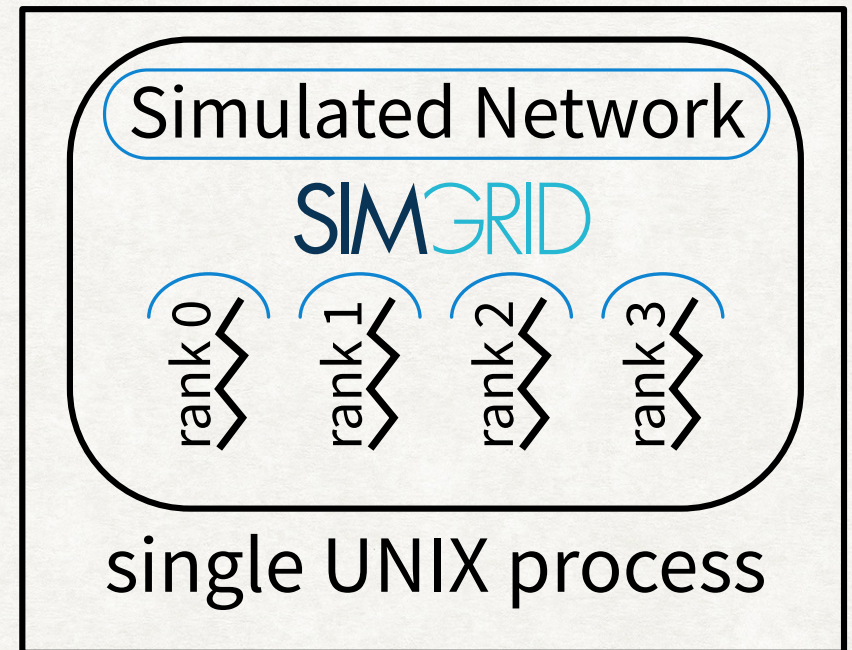
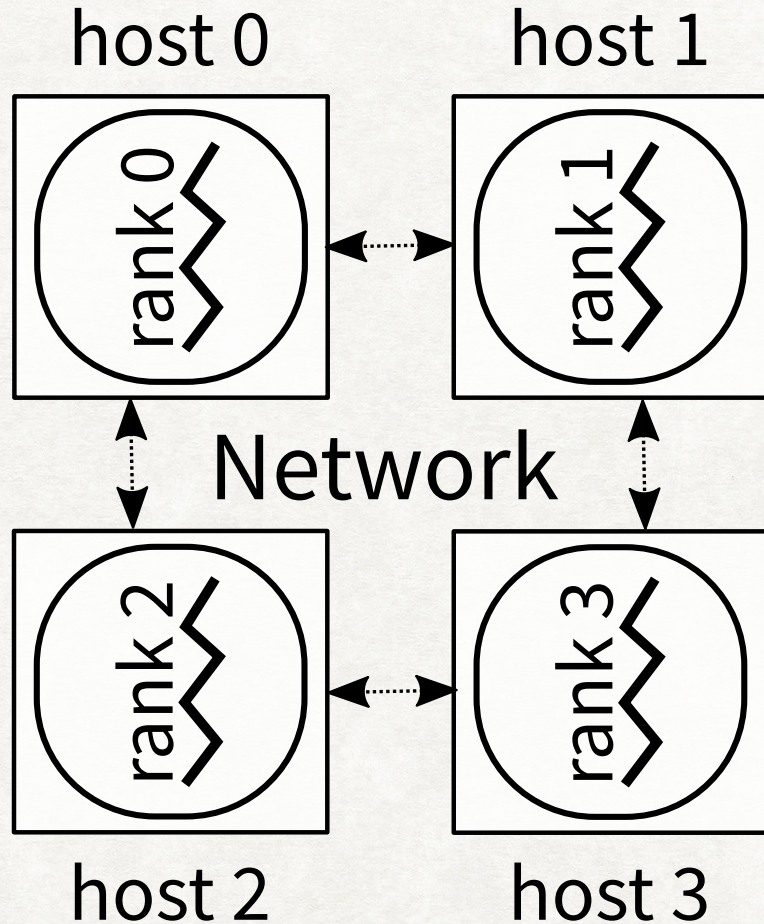
- SMPI (Simulated MPI) comes with the SimGrid distribution: <http://simgrid.org>
- Easy to use:

```
smpicc myprogram.c -o myprogram
```

```
smpirun -np 12 -platform platform.xml  
-hostfile hosts.txt ./myprogram
```

- Full MPI-2 standard and some MPI-3
- Implements collective communications using algorithms in several MPI implementations
- Scalable and validated [Degomme et al., TPDS 2017]

MPI vs. SMPI

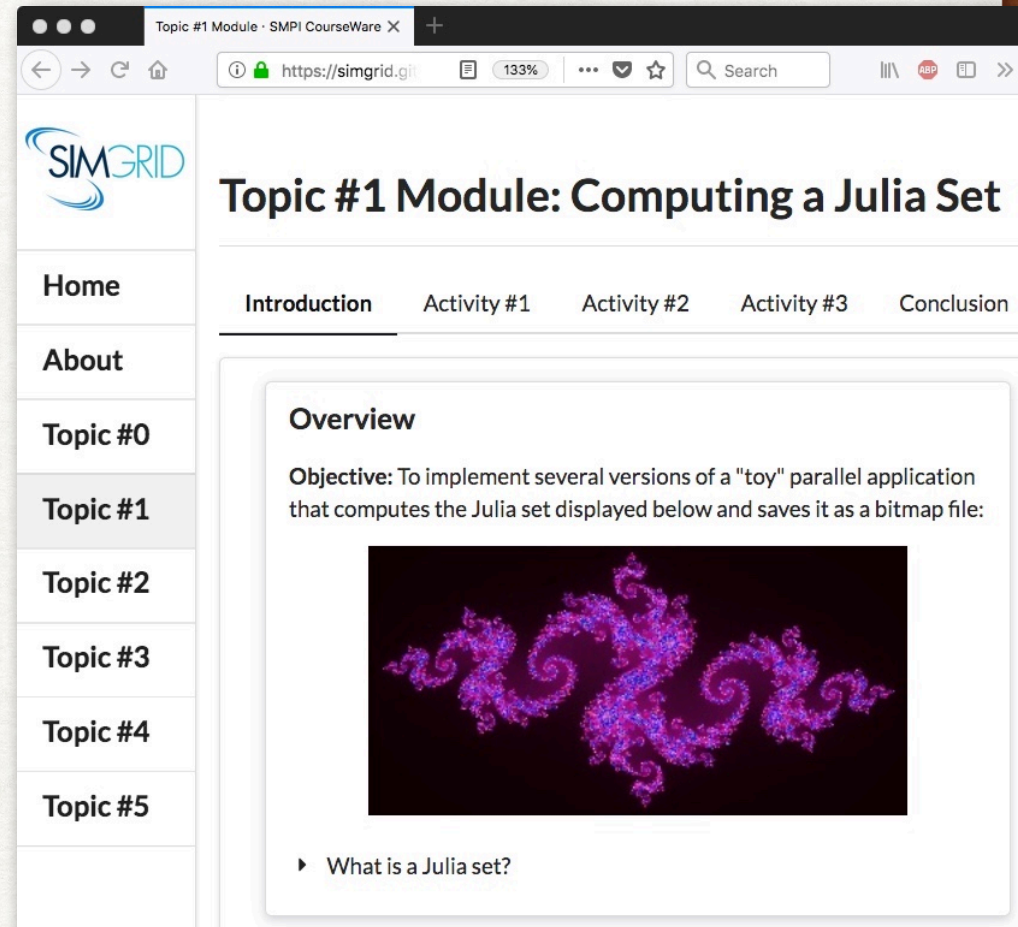


SMPI AND SCALABILITY

- By default, the entire application code is executed
- This is problematic at large scale (time and RAM)
- SMPI allows two tricks for scalability:
 - Trick #1: When re-running a basic block of code, just simulate a delay based on previous executions
 - Trick #2: When allocating memory, reuse an already allocated zone of memory
- These tricks break application correctness
- But they preserve simulated performance behavior and decrease simulation time by orders of magnitude

SMPI COURSEWARE

- SMPI CourseWare is not a curriculum
- It is a set of simulation-driven assignments that cover well-known SLOs
- These assignments can be integrated into any HPC course
- SMPI CourseWare has been used in several instances or two university courses at the graduate level
- Let's highlight a few of the things we have experienced/observed in the classroom...



LOW BARRIER OF ENTRY

- The first “getting your feet wet” assignment is about computing a Julia set
- One key advantage of simulation is that students can hit the deck running:
 - No need to Ssh to a remote machine
 - Fully interactive execution (no need to learn how to submit batch jobs)
 - No need to copy back the output for visualization
 - No “the cluster/network/whatever is down right now” problem
- These seem like details, but they get in the way of a good first experience
- This first assignment can thus be done in a live-coding session in the classroom with the help of the instructor in 1h15min

ARBITRARY PLATFORMS

- An assignment targets collective communications
 - Students implement several broadcast algorithms and compare to production MPI_Bcast implementations
- In this assignment students use many physical topologies
 - Rings, k-ary n-cubes, trees, crossbar switches, hierarchical, ..
- This really drives home that “communication pattern and physical topology impedance” matters
- Would be impossible to achieve with real platforms

STUDENT-DRIVEN EXPERIMENTS/DISCOVERY

- We have observed that students begin experimenting on their own
 - “I set the latency to zero and my performance is radically better... my implementation must be latency-bound... wait... why do I send so many tiny messages???”
 - “If I set my processors to be twice as fast, then I don’t go twice as fast... I have a communication bottleneck!”
 - ...
- Student-driven learning seems to work really well once they are given a simulation to play with
 - With short time-to-results instead of “queue waiting time + execution time” that get in the way
 - And, yes, there is something strange about waiting 10s for an output that says “this has taken 2hours” (but when asked, students don’t complain about the “fakeness” of it all)

CONCLUSION

- SMPI is available to anyone and welcomes contributors
 - It is easy to adapt and extend “real platform” assignments to become “in simulation” assignment (and we can help!)
- We are using this “teaching with simulation” in other contexts, still with SimGrid as the base technology
 - Cloud computing (ENS Rennes)
 - Scientific workflows (UHM, ISI/USC)
- Ultimate goal: a large corpus of reusable assignments for the HPC curriculum at large, for high-school, undergraduate, and graduate students

https://simgrid.github.io/SMPI_CourseWare/