Bridging Concepts and Practice in eScience via Simulation-driven Engineering

Rafael Ferreira da Silva[1], Henri Casanova[2], Ryan Tanaka[2], Frédéric Suter[3]

[1] USC Information Sciences Institute, Marina del Rey, CA, USA
[2] Information and Computer Sciences, University of Hawaii, Honolulu, HI, USA
[3] IN2P3 Computing Center, CNRS, Villeurbanne, France

WRENCH

https://wrench-project.org

Disconnect between theoretical and practical works

Theoreticians produce results that are never used by practitioners

Practitioners use approaches that may be vastly suboptimal because they are not informed by any theory

One of the reasons for this disconnect is that theoretical work must be done using formally defined models of computation

Ideally, these models are complete enough to be relevant to practice, but simple enough that obtaining theoretical results (e.g., optimality results, complexity bounds) is tractable

Real-world experiments are limited

One is limited to **particular platform configurations** (and sub-configurations)

- How can "what if?" scenarios be explored?
- How can generality be claimed?

One is limited by **specifics of the software infrastructure** that impose constraints on CI application executions

- Modifying complex software stacks (often written by others) just to test out ideas is not feasible

In the end, **the scope of real-world experiments is limited**, which impedes progress / discovery

Simulation

When one works in an experimental field in which **experiments are problematic**, one resorts to simulation

- Physicists have understood this decades ago :)

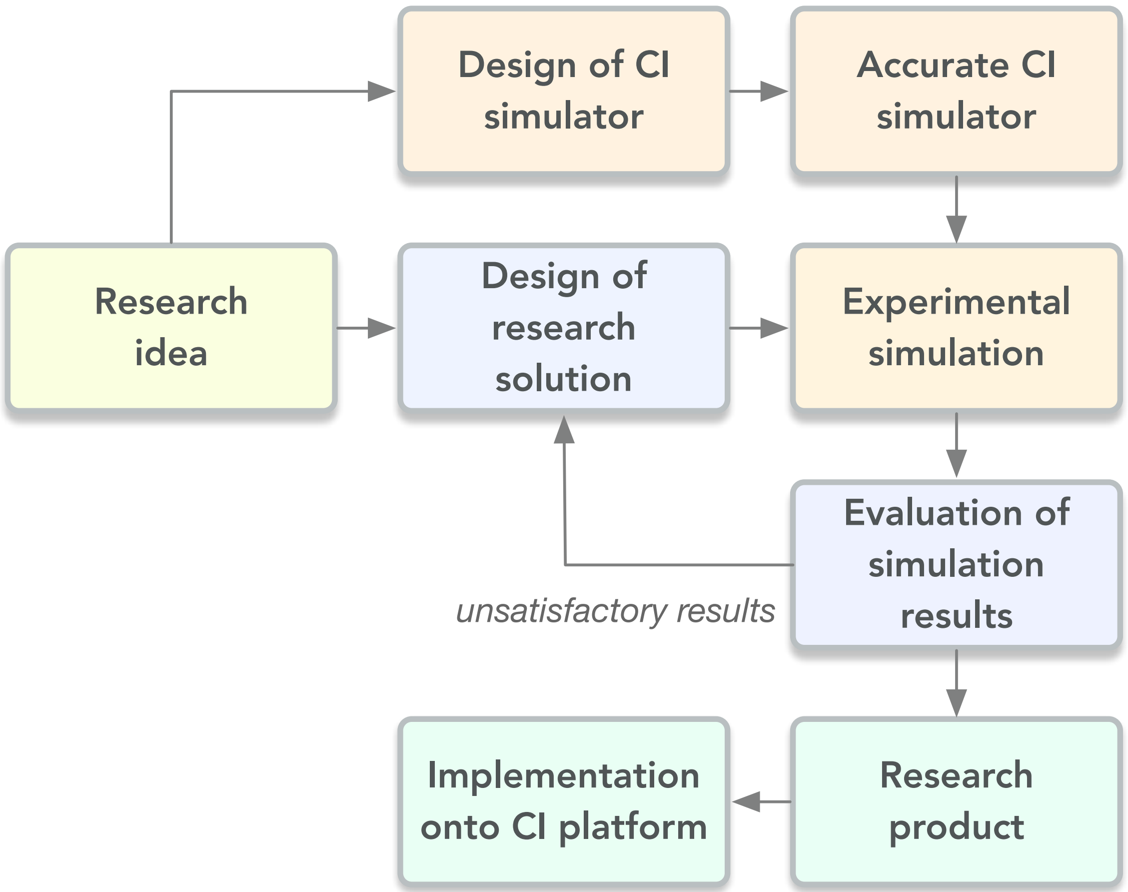
In some fields of Computer Science simulation is a **standard research and development methodology**

- e.g., Networking, Computer Architecture

Several simulators and simulation frameworks have been developed for **parallel and distributed computing**

- Some of them developed explicitly for workflows

Simulation-driven engineering life cycle



The ability to define **parameterizable services** is key for developing **accurate** CI simulators, from which **research products** evaluated via experimental simulation could be seamlessly integrated into actual CI platforms

The SimGrid framework

https://simgrid.org

SimGrid is a research project

- Development of **simulation models** of hardware/software stacks
- Models are **accurate** (validated/invalidated) and **scalable** (low computational complexity, low memory footprint)

SimGrid is open source usable software

- Provides different APIs for a range of simulation needs, e.g.:
  - **S4U**: General simulation of Concurrent Sequential Processes
  - SMPI: Fine-grained simulation of MPI applications

SimGrid is versatile scientific instrument

- Used for (combinations of) **Grid**, **HPC**, **Peer-to-Peer**, **Cloud**, **Fog** simulation projects

First developed in 2000, latest release: v3.23.2 (July 2019)

SimGrid's philosophy

https://simgrid.org

SimGrid's philosophy: provide **low-level abstractions**

Advantage: you can do anything with it

Drawback: implementing a simulation of a complex system is a lot of work

Critical analysis:

In [Kecskemeti et al.'14] pinpoints exactly the above trade-off:

"SimGrid is more scalable and validated than competing frameworks, but just too much work when wanting to simulate a WMS that interacts with CI components"

The WRENCH simulation framework

Objective #1: Make it **easy** to develop simulators of **complex CI application executions**

Done by providing high-level, reusable simulation abstractions

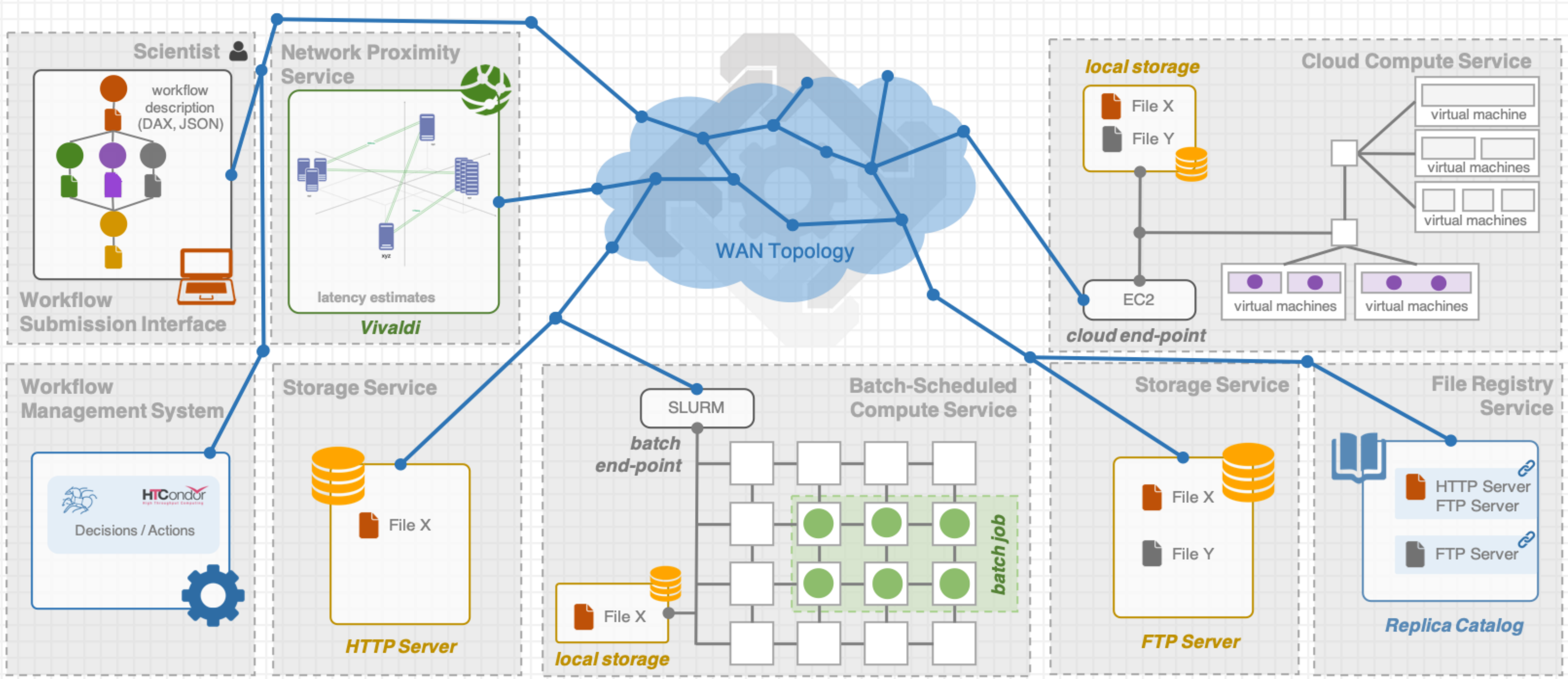
Objective #2: Produce **accurate** and **scalable** simulations

Done by building on SimGrid

WRENCH

Let's look at an example system one can simulate with WRENCH…

System to simulate



WRENCH core services

Simulation core

All necessary simulation models and base abstractions (computing, communicating, storing), provided by ***SimGrid***

Simulated core CI services

Abstractions for simulated CI components to execute computational workloads

Compute Services

Provide mechanisms for executing application tasks, which entail I/O and computation

bare-metal | virtualized cluster | cloud | batch-scheduled cluster

Storage Services

Store application files, which can then be accessed in reading/writing by the compute services when executing tasks that read/write files

File Registry Services

Databases of key-value pairs of storage services and files replicas

Network Proximity Services

monitor the network and maintain a database of host-to-host network distances

Workflow Management System

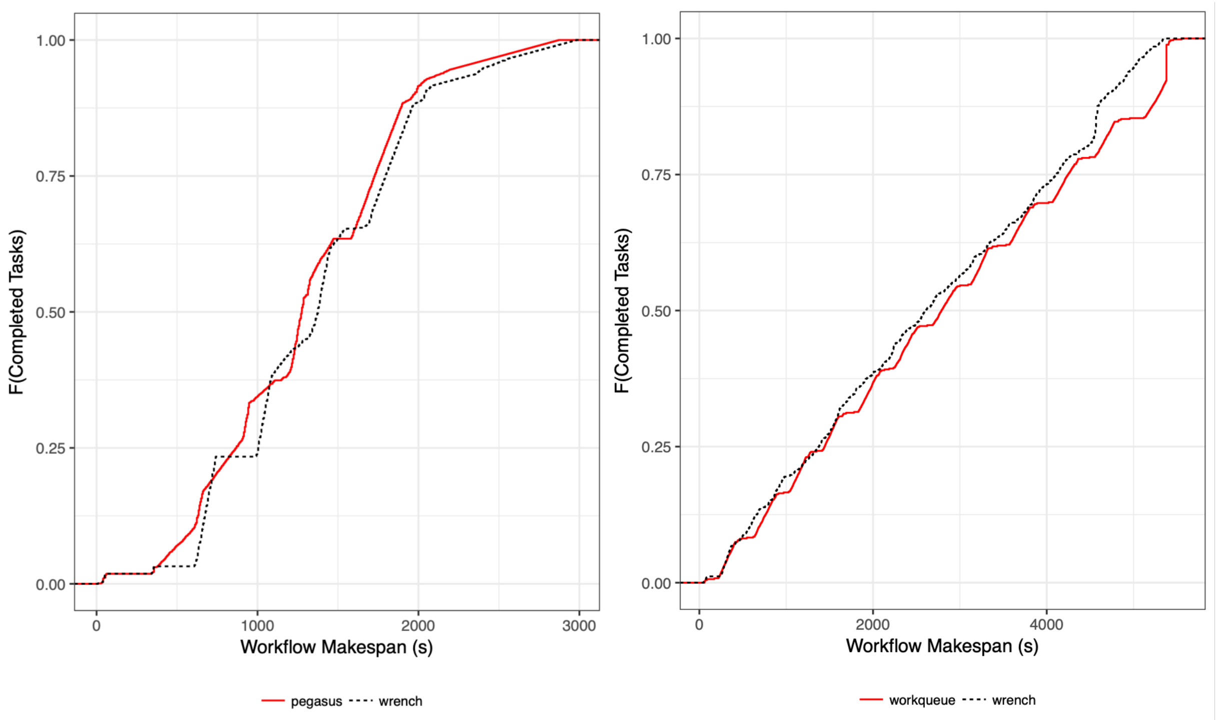
Provides the mechanisms for executing workflow applications, including decision-making for optimizing various objectives

WRENCH's impact on CI research

Accuracy: the ability to capture the behavior of a real-world system with as little bias as possible

Scalability: the ability to simulate large systems with as few CPU cycles and bytes of RAM as possible

Simulation Accuracy and Scalability

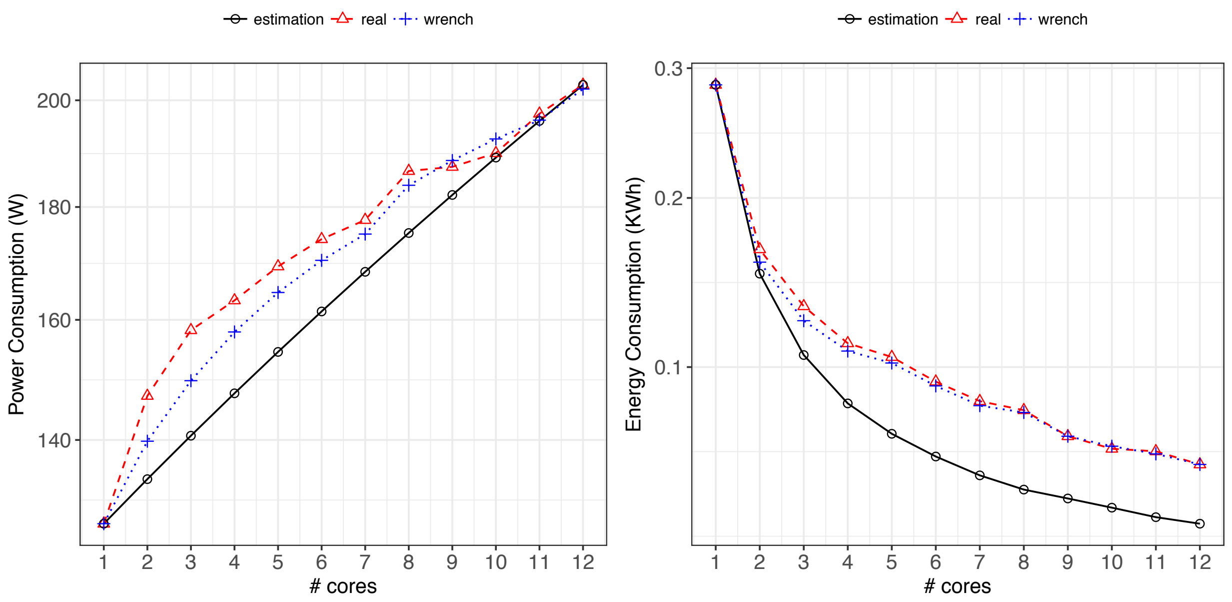


Empirical cumulative distribution function of task completion times for sample real-world ("pegasus" and "workqueue") and simulated ("wrench") executions.

WRENCH's impact on CI research

Energy-aware Computing

Investigated the impact of resource utilization and **I/O operations** on the **energy usage**, as well as the impact of executing multiple tasks concurrently on multi-socket, multi-core compute nodes



Comparison of power (*left*) and energy (*right*) consumption measurements for a real-world application ("`real`") using a well-known model from the literature ("`estimation`") and our WRENCH model ("`wrench`")

WRENCH
Pedagogic Modules

Simulation-driven self-contained pedagogic modules supported by WRENCH-based simulators

Activities entail running, through a **Web application**, a simulator with different input parameters

https://wrench-project.org/wrench-pedagogic-modules

WRENCH PEDAGOGIC MODULES Distributed Computing Courseware

A Primer on Networking

Networking | Latency & Bandwidth | Topologies | **Contention**

Learning objectives:

- Understand the concept of contention
- Be able to estimate data transfer times in the presence of contention

Networks are shared

Typically, several data transfers are occurring concurrently (at the same time) on a network topologies, and these transfers may be using the same network links. For instance, two concurrent transfers could be along two routes that share a single link. As a result, a data transfer's performance can be impacted by other data transfers. When a data transfer goes slower than it would go if alone in the network, it is because of *contention* (i.e., competition) for the bandwidth of one or more network link.

A Simple example

Consider the following topology with the two depicted data transfers, that each were started at exactly the same time and transfer 100 MB of data (symbolized by the red and the green arrow).

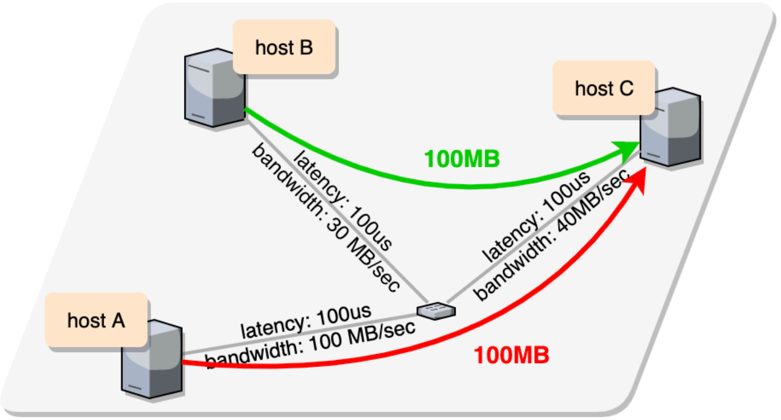
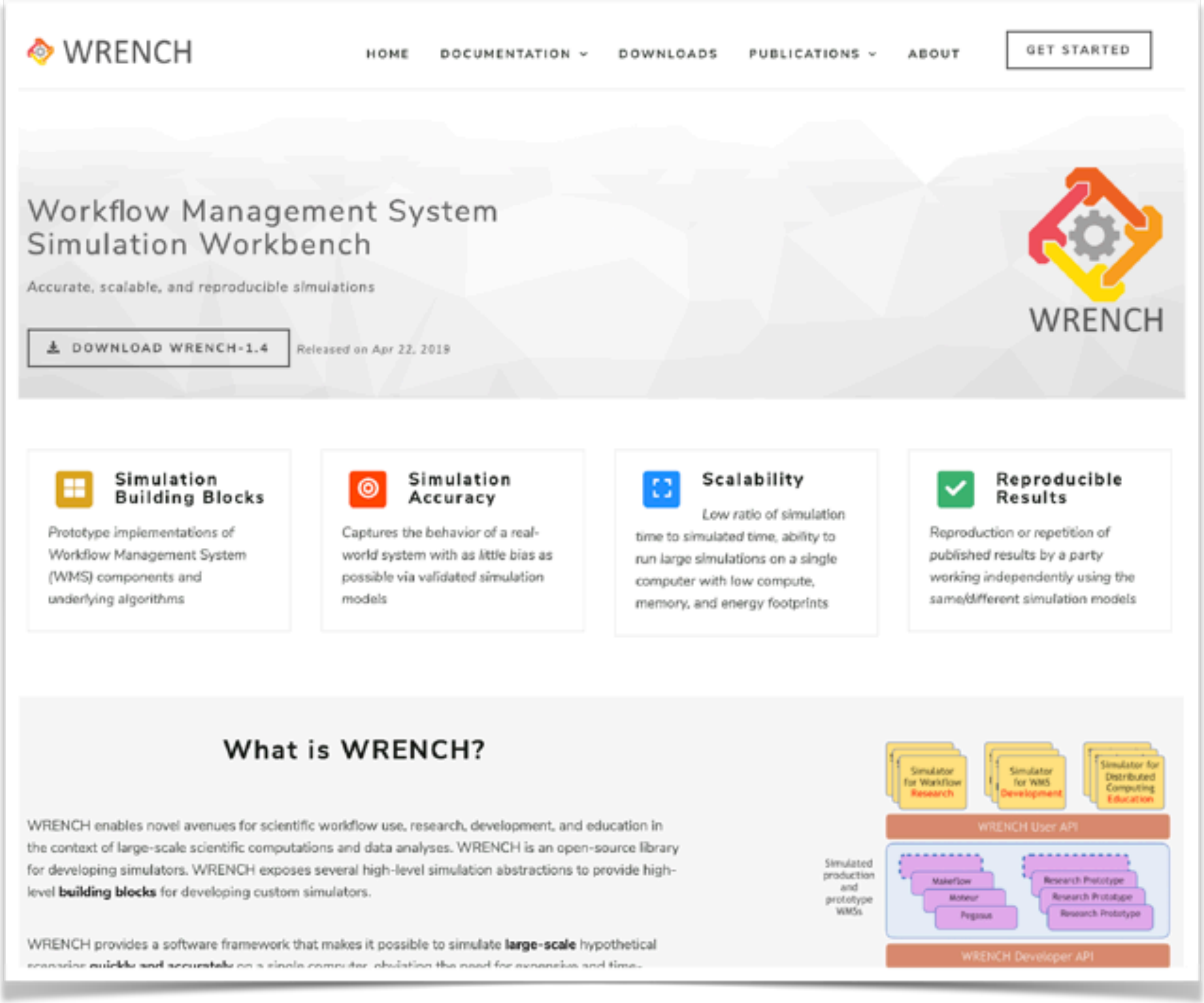


Figure 1: A simple example in which two data transfers contend for bandwidth.



Thank You

Questions?

https://wrench-project.org

This work is funded by NSF contracts #1642369 and #1642335; by CNRS under grant #PICS07239; and partly funded by NSF contracts #1923539 and #1923621.

UNIVERSITY of HAWAI'I MĀNOA
USC Viterbi School of Engineering Information Sciences Institute
CNRS IN2P3 Les deux infinis
NSF