Simulation Factory: Simplified Simulation Management

Michael Thomas (presenting author)[1,3], Dr. Erik Schnetter[2,3], Dr. Gabrielle Allen[2,3]

Department of Computer Science, Louisiana State University[1]
Department of Physics and Astronomy, Louisiana State University[2]
Center for Computation and Technology, Louisiana State University[3]

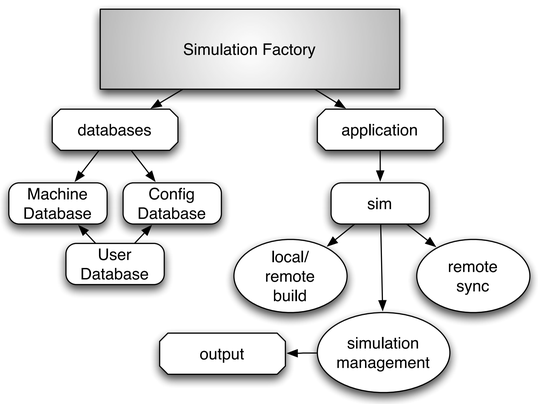
Abstract

Computational Science on large high performance computing resources is hampered by the complexity of these resources. Much of this complexity is due to low-level details on these resources that are exposed to the application and the end user. This includes *mechanisms for remote access*, *configuring and building applications from source code*, and *managing simulations and their output files via batch queue systems*.

The **Simulation Factory** addresses these challenges by simplifying remote access, building executables, and managing simulations. Furthermore, **Simulation Factory**'s automation avoids many possible user errors that can in the worst case render month-long simulations worthless.

Introduction and Motivation

Figure: The architecture of **Simulation Factory**



Simulation Factory addresses **four** main challenges introduced by the complexity of **High Performance Computing**:

- **Configuration**: Neutralize the uniqueness of each individual HPC resource through the use of the **Machine Database**, which describes the resource.
- **Synchronization**: Synchronize an authoritative source tree using various file transfer methods.
- **Remote Access**: Authenticate communication between a host resource and a remote HPC resource.
- **Manage Simulations**: Deploy and manage simulations, and provide a consistent means to collect and manage output.

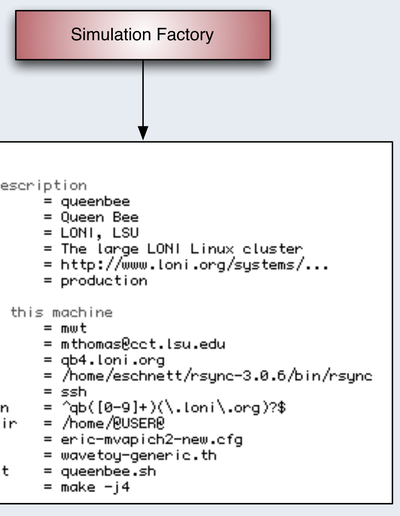
Configuration

The **Machine Database**: An **INI-Style** plain-text database that describes the following key elements of the host resource:

- **Host location, hostname, and other details**
- **Authentication, file access and synchronization tools**
- **Default directory structures**
- **Job queueing systems and submission commands**

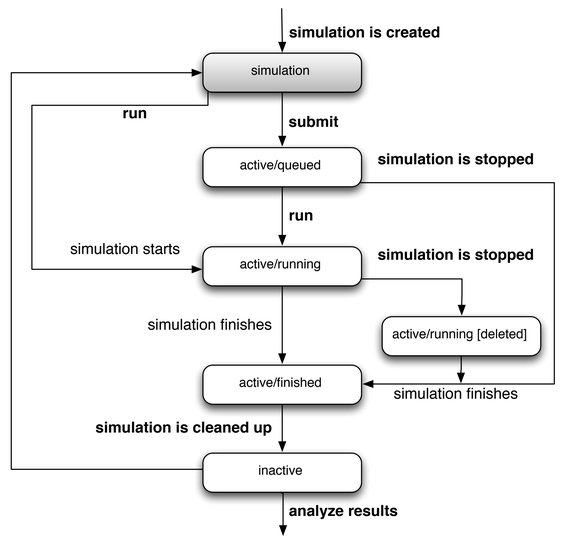
Machine Database

Figure: An excerpt from the **Machine Database** entry for the HPC Resource QueenBee



Simulations

Figure: The lifecycle of a simulation within **Simulation Factory**



A **Simulation Factory** simulation has a well defined lifecycle. The simulation has **three** major stages:

- *create:* **Initializes** the simulation, creating any necessary files and directories for direct execution or submission into the host queuing system.
- *submit:* **Submits** the simulation to the queueing system of the host resource.
- *run:* **Executes** a simulation in the host queuing system, or directly executes a simulation without the use of the host queueing system.

Future Work

Simulation Factory continues to evolve. New work will focus on:

- Creation of a graphical user interface for easier adoption.
- Adapt Simulation Factory to support many simulation toolkits.
- Archiving simulations with metadata that describes each unique simulation.
- Support for research group collaboration.

Acknowledgments

This work is supported by NSF awards #0932251 (TeraGrid Extension: Bridging to XD) and #0721915 (Alpaca), #0701566 (XiRel), #1005165 (NSF REU), and #0904015 (CIGR). We used TeraGrid resources at LONI, NCAR, NCSA, Purdue, and TACC under allocation TG-MCA02N014.