CENTER FOR COMPUTATION & TECHNOLOGY

TeraGrid-DEISA: Application-Level Interoperability

A Science-Driven Project Using Advanced CyberInfrastructure funded by NSF via a HPCOPS award to LONI

Morris Riedel

Project Lead: Shantenu Jha

http://saga.cct.lsu.edu

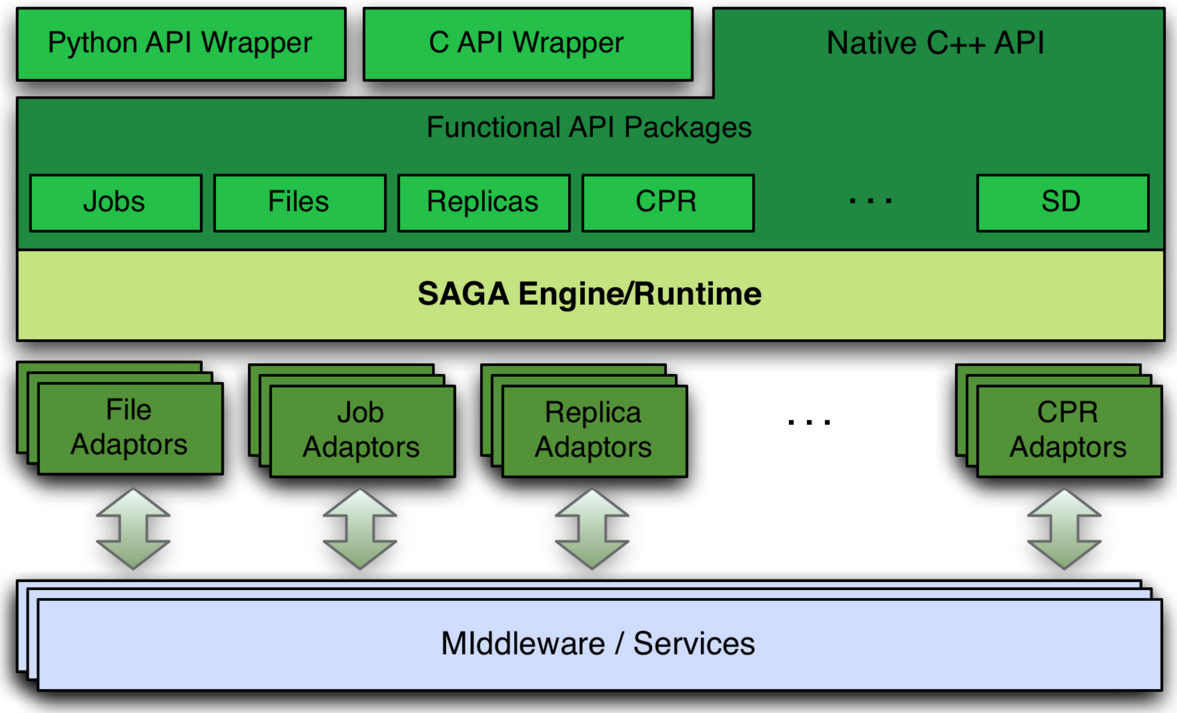
http://www.teragridforum.org/mediawiki/index.php?title=LONI

CENTER FOR COMPUTATION & TECHNOLOGY

In a Nutshell

- SAGA: A novel way to develop applications
  - Facilitates Applications using frameworks respecting characteristics and requirements
  - Demonstrated effectiveness in *scaling-out*
  - Extend & generalize to achieve science (contributing to the VPH) using resources on LONI, TeraGrid & DEISA concurrently without prior reservations.
- No “paired Grid” concept. Not System-level but Application-level Interoperabilty:
  - Enabling Dynamic Execution of HPC Applications

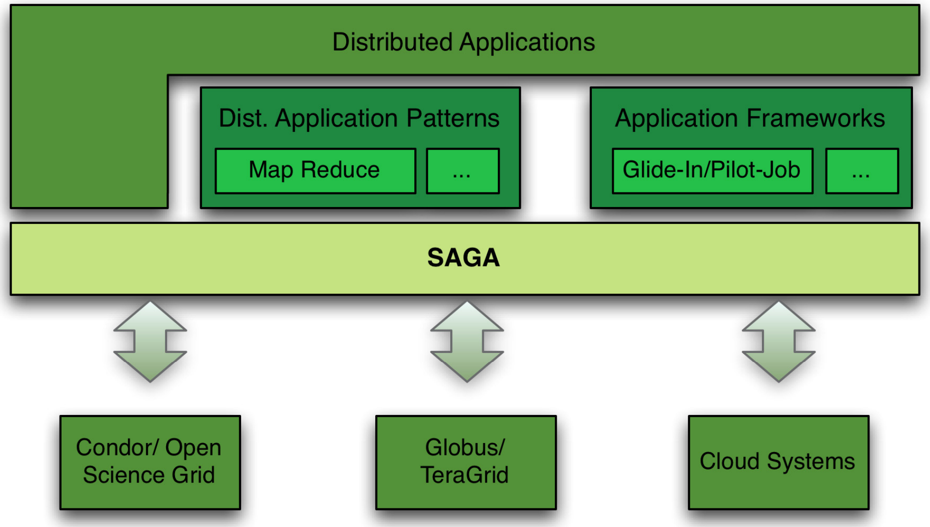
SAGA: In a Nutshell



Hybrid Applications that are both compute and data-intensive

SAGA and Applications

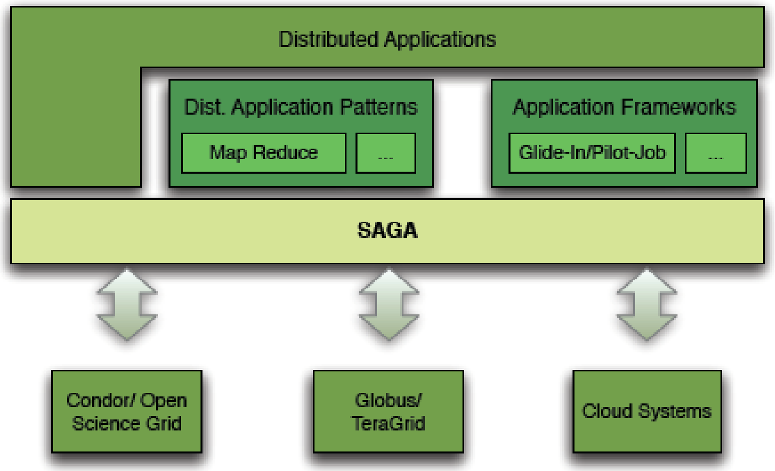
- *Legacy* application -- distributed execution modes
  - Replica-Exchange MD
  - *Novel* first-principles applications
- Distributed application using patterns
  - MapReduce
- Applications using Frameworks
  - Frameworks can use patterns

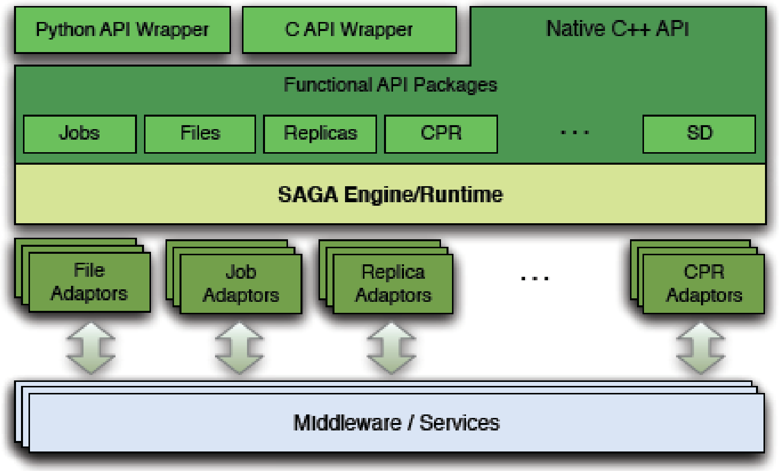


The theoretical underpinnings of developing applications using SAGA are strongly influenced by the DPA theme

CENTER FOR COMPUTATION & TECHNOLOGY

SAGA: Unified View





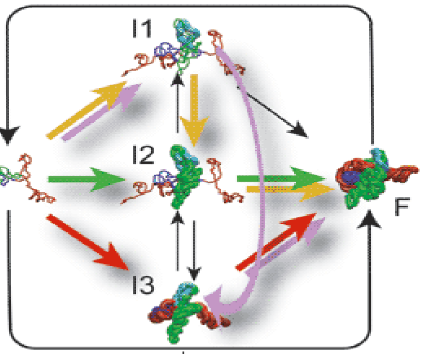
Focus on Application Development and Characteristics, not infrastructure details

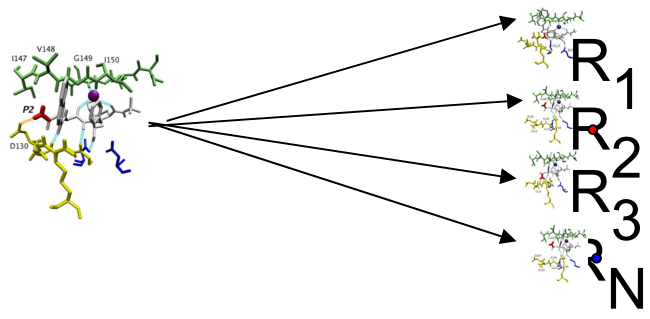
CENTER FOR COMPUTATION & TECHNOLOGY

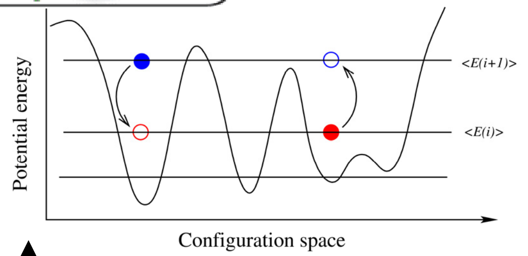
omii-uk Software Solutions for e-Research

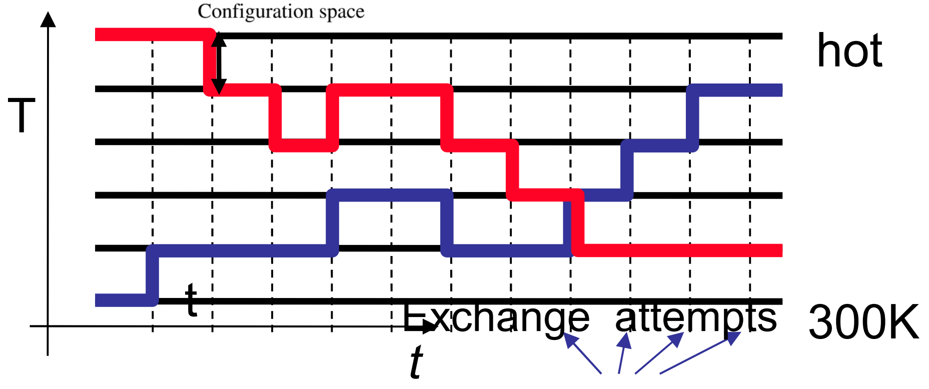
Distributed Execution Modes Replica Exchange: Hello Distributed World

- Task Level Parallelism
  - Embarrassingly distributable!
  - Loosely coupled
- Create replicas of initial configuration
- Spawn 'N' replicas over different machine
- Run for time *t ;* Attempt configuration swap
- Run for further time t; Repeat till finish

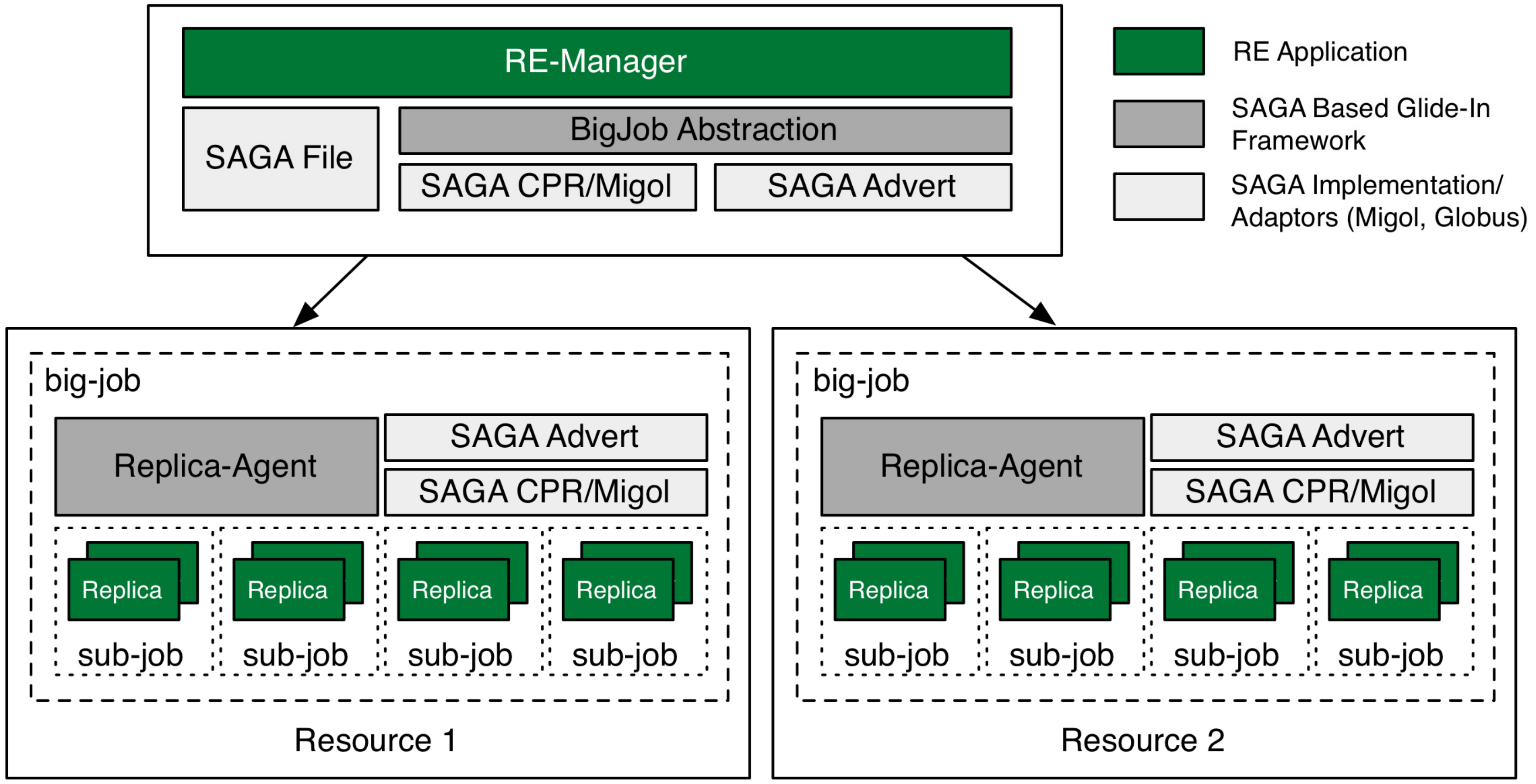






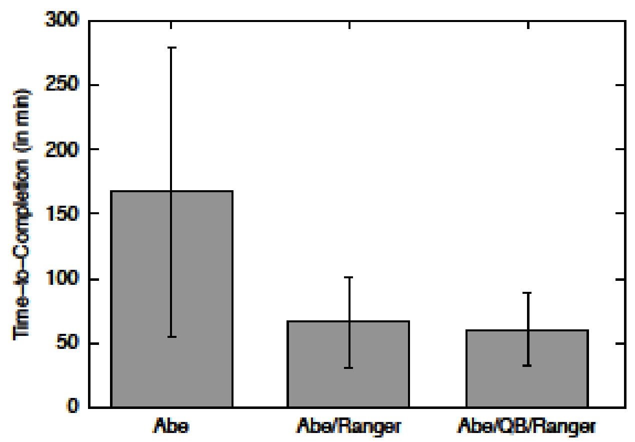


FAUST: Framework to Support Deployment & Scheduling of Pilot/Multiple Jobs



FAUST production-level implementation coming:
http://macpro01.cct.lsu.edu/~oweidner/faust/

Using Multiple Resources Performance Enhancements



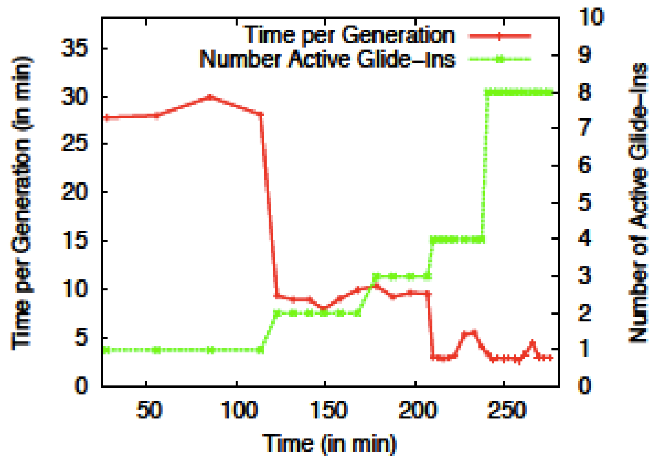
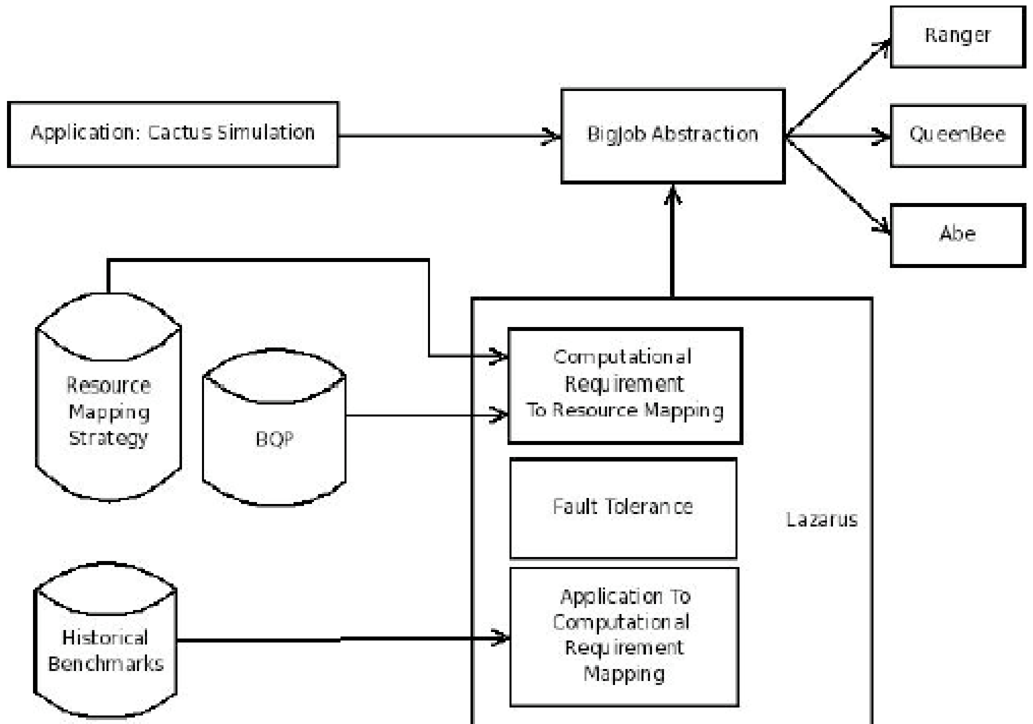
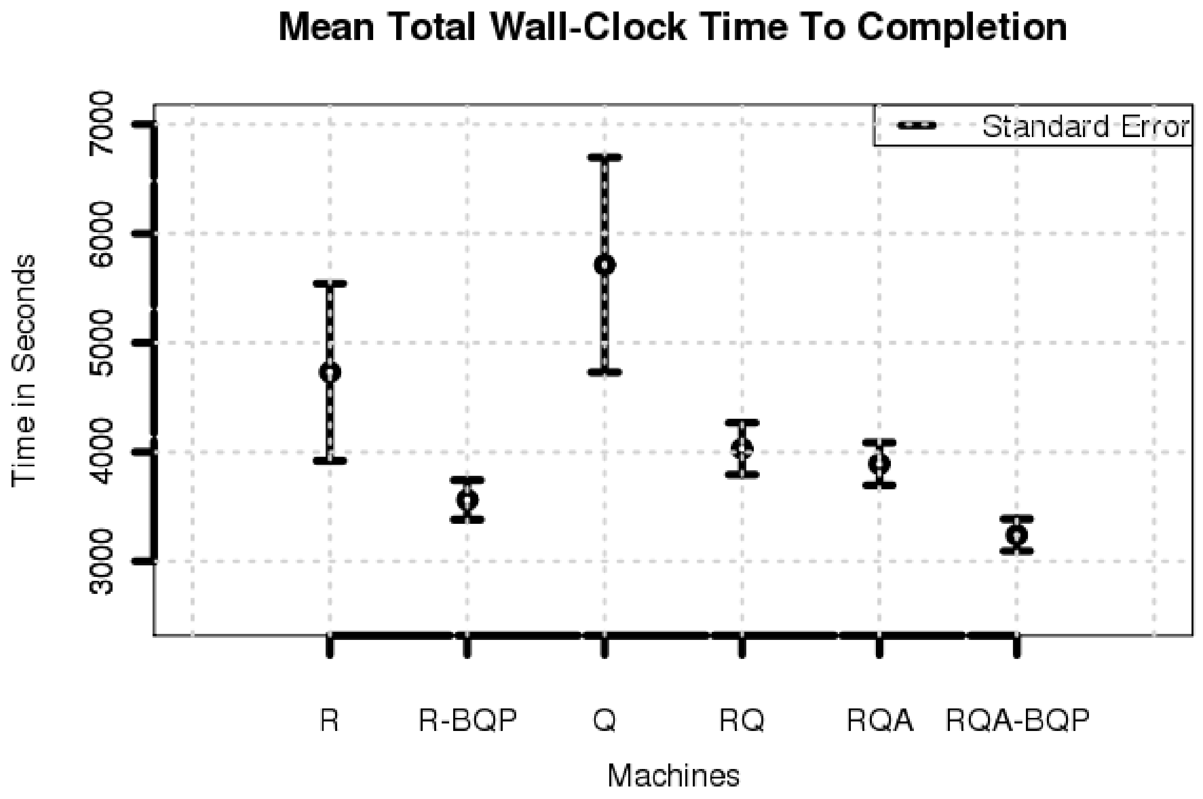


Figure 3: Performance data providing conclusive evidence that SAGA can be used to lower the time-to-solution, as the number of resources that can be used increases. SAGA provides the ability to use multiple resources in a simple and scalable fashion. The total number of computer-cycles used do not necessarily increase, i.e., time-to-completeness is not at the expense of efficiency. The lower figure contains plots which show the time-series of the average times between exchange attempts (upper line using the left-hand y axis) and the number of active Glide-Ins over a 6hr run.

Lazarus: A SAGA-Based Framework for Autonomic Applications


Application: Cactus Simulation
BigJob Abstraction
Ranger
QueenBee
Abe
Resource Mapping Strategy
BQP
Historical Benchmarks
Computational Requirement To Resource Mapping
Fault Tolerance
Application To Computational Requirement Mapping
Lazarus

CENTER FOR COMPUTATION & TECHNOLOGY



Plots showing Performance Advantages arising from abstractions for Autonomic applications. SAGA-based LAZARUS Framework lowers time to solution by facilitating *scaling-out.*

The aim is to utilize and extend capabilities, provided by higher-level abstractions (e.g. FAUST, LAZARUS) to utilize distinct, multiple, Heterogenous Grids simultaneously.