Applications

Motivating, Evaluating, and Demonstrating CGrADS Research

Ian Foster
Department of Computer Science
University of Chicago
Mathematics and Computer Science Division
Argonne National Laboratory

http://hipersoft.rice.edu/stc_site_visit/talks/applications.pdf
Enabling Grid Computing

- The GrADS vision: federated computers and software
  - An “application” is constructed dynamically from services and components on the network—selected to meet requirements
  - A “computer” is a dynamically constructed collection of processors, data sources, sensors, networks—optimized for our application

- And thus
  - Reduced barriers to access mean that we do much more computing, and more interesting computing, than today => Many more components (& services); massive parallelism
  - Distributed resource ownership => Sharing (for fun or profit) is fundamental; so are trust, policy, negotiation, payment
  - Computing is performed, increasingly, on unfamiliar systems => Dynamic behaviors, discovery, adaptivity, failure

- Challenge: exploring such future scenarios today, in compelling yet realistic settings
  - Identify, address fundamental issues (beyond RPC syntax of the day)
CGrADS Application Strategy

- Select applications with challenging requirements and aggressive user communities
- Run each application through cycle multiple times, each time with increased sophistication
- Phase in new applications over course of project
- Initial targets
  - Computational astrophysics
  - Environmental modeling
- Subsequent targets
  - High energy physics
  - Earthquake engineering
  - Industrial applications
  - Genomics, microscopy
Numerical Relativity & Cactus

• Numerical simulation of extreme astrophysical events: colliding black holes, neutron stars, ...
  — Understand physics; predict gravitational wave forms
  — Relativistic effects => Einstein eqns
    - Computationally intensive (can be 1000s flops/grid point)
    - 3-D simulations only recently possible

• Cactus = modular, portable framework for parallel, multidimensional simulations
  — Construct codes by linking
    - Small core (flesh): mgmt services
    - Selected modules (thorns): Num. methods, grids & domain decomps, viz, steering, etc.
  — Custom linking/configuration tools

Colliding black holes

LIGO gravitational wave observatory
Dynamic Grid Computing and Cactus

• Application behaviors in a Grid environment:
  – Identify fastest/cheapest/biggest resources
  – Configure for efficient execution
  – Detect need for new resources or behaviors (e.g., new subtasks, resource slowdown, new appln regime, new resource available)
  – Adapt, and/or discover new resources; invoke subtasks on new resources and/or migrate

• Cactus thorns for management of appln behavior & resource use
  – Heterogeneous resources, e.g.:
    - Irregular decomp.; comms scheduling for comp/comm overlap
    - Variable halo for managing message size
    - Msg compression (comp/comm tradeoff)
  – Dynamic resource behaviors/demands, e.g.:
    - Perf monitoring, contract violation detection
    - Dynamic resource discovery, subtask spawning, migration
    - User notification and steering
Cactus Example: Terascale Computing

- Solved EEs for gravitational waves (real code)
  - Tightly coupled, communications required through derivatives
  - Must communicate 30MB/step between machines
  - Time step takes 1.6 sec

- Used 10 ghost zones along direction of machines: communicate every 10 steps

- Compression/decomp. on all data passed in this direction

- Achieved 70-80% scaling, ~200GF (only 14% scaling without tricks)
Model Problem: The Cactus Worm

- Migrate to “faster/ cheaper” system
  - When better system discovered
  - When requirements change
  - When characteristics change (e.g., competition)
  - On user request

- Tests most elements of Cactus & GrADS
- Evaluate on GrADS testbed

GrADS MacroGrid Testbed

- Architecture involves new Cactus thorn
- Resource selector detects available resources and determines when to migrate
- Application manager orchestrates migration
- Globus Toolkit substrate for resource discovery, allocation, management
Cactus Worm
Detailed Architecture & Operation

(0) Possible user input

Cactus "flesh" & other thorns

(1) Adapt. request

"Tequila" Thorn

(2) Resource request

Application Manager

(5) Cactus startup

GrADS Resource Selector

(4) Migration request

Storage resource

(7) Read checkpoint

Grid Information Service

(3) Write checkpoint

Query

Code repository

(6) Load code

Store models, etc.

(1') Resource notification
• Tequila thorn
  — Contract monitor driven by three user-controllable parameters
    - Time quantum for “time per iteration”
    - % degradation in time per iteration (relative to prior average) before noting violation
    - Number of violations before migration
  — Communicates with resource monitor via ClassAd-based protocol
    - Specify resource requirements & performance model
    - Can request synchronous or asynchronous notification
  — Generates checkpoint and initiates migration

• Resource selector
  — Uses Globus Toolkit MDS-2 mechanisms to discover and monitor resources
  — Implements “cluster matching” algorithm to detect suitable clusters
Migration in Action

Running At UC
Load applied
3 successive contract violations
Resource discovery & migration
Running At UIUC

(migration time not to scale)
Future Application Directions

• Next steps with Cactus (with EU GridLab project)
  — Integrate with GrADSoft, e.g.
    - Automated contract monitoring
    - Program Preparation System
    - Configurable Object Program and Application Launcher
  — New application scenarios
    - E.g., subtask creation, adaptive mesh refinement

• New applications to be introduced over time, with partners
  — GriPhyN: data-intensive high energy physics, astronomy applns
  — CAPS: environmental modeling, real-time data acquisition
  — IBM, Boeing, Lockheed: industrial, business intelligence, autonomic computing
  — Alliance for Cellular Signalling, PDB: Genomics and related topics
  — NCMIR: Real-time microscopy
  — NEES: Earthquake engineering, data analysis, simulation
Summary

• Application investigations are critical to CGrADS goals
  — Motivate, evaluate, demonstrate, and transfer R&D results

• We partner with application groups with challenging applications
  — Iteratively refine Grid-enabled application and CGrADS tools

• First such partnership involves the Cactus astrophysics code
  — Lessons learned
    - A real & demanding application can exploit adaptive techniques to execute efficiently in Grid environments
    - Even a relatively regular application can incorporate a range of useful mechanisms for adaptive behaviors & resource demands
  — Outcomes to date
    - Grid-enabled Cactus: wonderful experimental platform

• Future directions will involve increasingly aggressive (and ever more automated) application scenarios