Hosted Science:
Managing Computational Workflows in the Cloud

Ewa Deelman
USC Information Sciences Institute

Acknowledgements: Gideon Juve (USC), Maciej Malawski (AGH), Jarek Nabrzyski (ND), Applications: Bruce Berriman et al (Caltech), Tom Jordan et al (USC),

http://pegasus.isi.edu
deeelman@isi.edu
The Problem

• Scientific data is being collected at an ever increasing rate
  • The “old days” -- big, focused experiments – LHC
  • Today “cheap” DNA sequencers – and an increasing number of them

• The complexity of the computational problems is ever increasing

• Local compute resources are often not enough (too small, limited availability)

• The computing infrastructure keeps changing
  • Hardware, software, but also computational models
Computational workflows
--managing application complexity

Help express multi-step computations in a declarative way
Can support automation, minimize human involvement
  Makes analyses easier to run
Can be high-level and portable across execution platforms
Keep track of provenance to support reproducibility
Foster collaboration—code and data sharing
So far applications have been running on local/campus clusters or grids

MPI codes ~ 12,000 CPU hours,

Post Processing 2,000 CPU hours

Data footprint ~ 800GB
DNA sequencing, a new breed of data-intensive applications

Data collected at a sequencers
  Needs to be filtered for noisy data
  Needs to be aligned
  Needs to be collected into a single map

Vendors provide some basic tools
  you may want to try the latest alignment algorithm
  you may want to use a remote cluster

Challenges:
  automation of analysis, reproducibility
  Portability
  provenance

USERS!
Hosted Science—making it easier for the user

- Today applications are using the cloud as a resource provider (storage, computing, social networking)
- In the future more services will be migrating to the cloud (more integration)
  - Hosted end-to-end analysis
  - Data and method publication
  - Instruments
The Future is Now
Illumina’s BaseSpace

Data Analysis

BaseSpace now performs one alignment and variant detection for free on all Illumina data! To learn more about what’s included, click here.

BaseSpace makes data analysis easy. Push-button tools let researchers easily leverage all types of analysis applications and seamlessly view their results. Our flexible “app store” environment is being developed to bring the industry’s best tools to your fingertips, with new tools added constantly.

Currently, BaseSpace can perform the following analyses on your data:

- **Resequencing Alignment**
  - Sequencing of an enriched portion of the human genome, or of a small genome (such as a cow). Reads are aligned against the reference, and variants are noted.

- **AmpliSeq Sequencing**
  - Sequencing of PCR amplicons from probes targeting particular genome positions (up to ~100 loc, from up to ~96 samples).

- **De Novo Assembly**
  - Assembly of small (< 2MB) genome from 16S ribosome RNA reads without the use of a genomic reference.

- **Small RNA Analysis**
  - Resequencing workflow applied to microRNAs.

- **Library QC**
  - Fast resequencing of a reference genome to QC the DNA library.

- **Metagenomics**
  - The 16S metagenomics workflow is used to classify organisms from a metagenomic sample by amplifying specific regions in the 16S ribosomal RNA. The raw output of this workflow is a classification of reads at several taxonomic levels (kingdom, phylum, class, order, family, genus).

Prep
15 minutes hands-on
1.5 HOURS

Sequence
20 minutes hands-on
4 HOURS

Analyze
fully automated
3 HOURS

Share
secure and store

Workflow times include dual surface scanning and v2 kits.
Large-Scale, Data-Intensive Workflows

Montage Galactic Plane Workflow

- 18 million input images (~2.5 TB)
- 900 output images (2.5 GB each, 2.4 TB total)
- 10.5 million tasks (34,000 CPU hours)

An analysis is composed of a number of related workflows—

*an ensemble*
Workflow Ensembles

Set of workflows

Workflows have different parameters, inputs, etc.

Prioritized

Priority represents user’s utility

Montage 2MASS galactic plane (John Good, Caltech)

2009 CyberShake sites (SCEC)

USC

San Onofre Nuclear Power Plant
Issues when running on commercial clouds—part of hosted science

• Assume you are not running on academic clouds like FutureGrid

• It is difficult to manage cost
  • How much would it cost to analyze one sample?
  • How much would it cost to analyze a set of samples?

• It is difficult to manage deadlines
  • “I would like all the results to be done in a week”
  • “I would like the most important analyses done in a week”
  • “I have a week to get the most important results and $500 to do it”
Problem Description

How do you manage workflow ensembles in hosted environments?

Typical research question:

*How much computation can we complete given the limited time and budget of our research project?*

Constraints: **Budget** and **Deadline**

**Goal:** given budget and deadline, maximize the number of prioritized workflows in an ensemble
Explore provisioning and task scheduling decisions

Inputs:
- Budget, deadline, prioritized ensemble, and task runtime estimates

Outputs:
- **Provisioning**: Determines # of VMs to use over time
- **Scheduling**: Maps tasks to VMs

Algorithms:
- **SPSS**: Static Provisioning, Static Scheduling
- **DPDS**: Dynamic Provisioning, Dynamic Scheduling
- **WA-DPDS**: Workflow-Aware DPDS
SPSS

Plans out all provisioning and scheduling decisions ahead of execution (offline algorithm)

Algorithm:

For each workflow in priority order
Assign sub-deadlines to each task
Find a minimum cost schedule for the workflow such that each task finishes by its deadline
If the schedule cost $\leq$ the remaining budget: accept the workflow
Otherwise: reject the workflow

Static plan may be disrupted at runtime
DPDS

Provisioning and scheduling decisions are made at runtime (online algorithm)

Algorithm:

- Task priority = workflow priority
- Tasks are executed in priority order
- Tasks are mapped to available VMs arbitrarily
- Resource utilization determines provisioning

May execute low-priority tasks even when the workflow they belong to will never finish

We assume no pre-emption of tasks
WA-DPDS

DPDS with additional workflow admission test:

- Each time a workflow starts
- Add up the cost of all the tasks in the workflow
- If there is enough budget: accept workflow
- Otherwise: reject workflow

Other admissions tests are possible
- e.g. Critical path <= time remaining
Dynamic vs. Static
Task execution over time

Dynamic

Static

Time

VM

DPDS and WA-DPDS

SPSS

Dynamic

Static
Evaluation

Simulation
   Enables us to explore a large parameter space
   Simulator extends CloudSim framework

Ensembles
   Use synthetic workflows generated using parameters from real applications
   Randomized using different distributions, priorities

Experiments
   Determine relative performance
   Measure effect of low quality estimates and delays
Ensemble Types

5 different applications (based on real apps)

Ensemble size Number of workflows (50)

Workflow size

\{100, 200, 300, 400, 500, 600, 700, 800, 900, and 1000\}

Constant size

Uniform distribution

Pareto distribution

Priorities Priority not correlated with size
Performance Metric

Exponential score:

\[ \text{Score}(e) = \sum_{w \in \text{Completed}(e)} 2^{-\text{Priority}(w)} \]

**Key:** High-priority workflows are more valuable than all lower-priority workflows combined:

\[ 2^{-p} > \sum_{i = p+1} \ldots 2^{-i} \]

Consistent with problem definition
### Budget and Deadline Parameters

**Goal:** cover space of interesting parameters

\[
\begin{align*}
\min_{w \in e} & \ Cost(w), \ \sum_{w \in e} \ Cost(w) \\
\min_{w \in e} & \ CriticalPath(w), \ \sum_{w \in e} \ CriticalPath(w)
\end{align*}
\]

In all the experiments we assumed that the VMs have a price of $z_1$ per VM-hour. This price was chosen to simplify interpretation of results and should not affect the relative performance of the different algorithms. In this study we do not take into account the heterogeneity of the infrastructure since we assume that it is always possible to select a VM type that has the best price to performance ratio for a given application.

All the experiments were run with maximum autoscaling factor $v_{max}$ set to 1. After experimenting with DPDS and WA-DPDS, we found that, due to the high parallelism of workflows used, the resource utilization remains high enough without adjusting the autoscaling rate. Based on experiments with the target applications, we set the SPSS...
Relative Performance

How do the algorithms perform on different applications and ensemble types?

Experiment:

- Compare relative performance of all 3 algorithms on 5 applications
- 5 applications, 3 ensemble types, 10 random seeds, 10 budgets, 10 deadlines

Goal: Compare % of ensembles for which each algorithm gets the highest score
% of ensembles for which each algorithm gets the highest score
Inaccurate Runtime Estimates

What happens if the runtime estimates are inaccurate?

Experiment:

- Introduce uniform error of $\pm p\%$ for $p$ from 0 to 50
- Compare ratios of actual cost/budget and actual makespan/deadline
- All applications, all distributions, and 10 ensembles, budgets and deadlines each

**Goal:** See how often each algorithm exceeds budget and deadline
Inaccurate Runtime Estimate Results

Cost / Budget

Makespan / Deadline

Runtime estimate error

Cost / Budget

Makespan / Deadline

Runtime estimate error
Task Failures

Large workflows on distributed systems often have failures

Experiment:
- Introduce a uniform task failure rate between 0% and 50%, if a task fails, it is retried
- All applications, all distributions, and 10 ensembles, budgets and deadlines

**Goal:** Determine if high failure rates lead to significant constraint overruns
Task Failure Results

Cost / Budget

Makespan / Deadline
Summary—looking into the future

There is a move to hosting more services in the cloud

Hosting science will require (incomplete list)

• a number of integrated services
• seamless support for managing resource usage and thus cost and performance
• tools to help evaluate costs for entire computational problems (ensembles), not just one workflows
• ease of use---can you do science as an app?

References: http://pegasus.isi.edu, Paper at SC’12