Scientific Grid Computing via Community-Controlled Autobuilding of Software Packages Across Architectures

Steffen Möller\textsuperscript{1} Daniel Bayer\textsuperscript{1} David Vernazobres\textsuperscript{2}
Albrecht Gebhardt\textsuperscript{3} Dirk Eddelbüttel\textsuperscript{4}

\textsuperscript{1}University of Lübeck, Institute for Neuro- and Bioinformatics, \textsuperscript{2}Westphalian Wilhelms University of Münster, Institute for Evolution and Biodiversity, Division of Bioinformatics, \textsuperscript{3}University of Klagenfurt, Institute for Statistics, \textsuperscript{4}Debian Project, Chicago

NETTAB, Pisa
June 2007
Outline

Motivation
   Grid computing
   Challenge
   R packages
   Debian

Methods
   Automated Packaging
   Grid Runtime Environments
   Selection of Packages

Results
   RDF Catalog of Runtime Environments
   RDF Representation

Discussion
Motivation for Grid Computing in Bioinformatics Research

Large number of data parallel problems:
  ▶ Image analysis
  ▶ Sequence analysis
  ▶ Statistical genetics

Long-lasting jobs
  ▶ Ligand screening, Protein docking
  ▶ Monte-Carlo Simulation

All data from biological high-throughput efforts
Principles of Grid Computing

“Integration of local batch systems”

- Users specifies a job
  - required software packages
  - cpu time
  - ...
- Site is selected that matches demands
- Job is executed on worker node of that site

half a working day to set up as server, 5 min as client... once certificates are available.

site\_n

until accepted

site\_2

site\_1

inspects

Job
NorduGrid and ARC

- Compute and data sharing grid
- Launched in 2001
- >7500 active hosts

Special features:
- Integrates regular batch systems
- Distributed data handling
- Minimally-invasive – single machine config

Details on www.nordugrid.org
Bringing huge software repositories to the Grid

Software installations are traditionally performed by site administrators:

- Restricted availability of resources
- Serious validation of error-prone installation

Heterogeneous communities do not know each other’s software and research aims

- Limited motivation
- Homogenize descriptions of packages

Independence of human factors sought
Provisioning of a Homogeneous Grid Environment

Bringing Science, Linux and Grid Communities Together
**R Statistics Environments**

Repositories: CRAN, BioConductor, Omegahat

---

**R**

- free software environment for *statistical computing* and graphics.
- available on UNIX, Windows and MacOS

---

**CRAN**

- > 1000 R packages provided by the community
- central repository
R Statistics Environments
Repositories: CRAN, BioConductor, Omegahat

**BioConductor**
- > 1000 R packages provided by the community
- central repository
- Methods supporting research in Bioinformatics (Microarrays, Proteomics, ...)
- Access to biological data and its visualisation

**Omegahat**
- Additions for using Java, Perl, SOAP, ...
Applications in Biological Research
biocView controlled vocabulary of BioConductor
The Debian Linux Distribution

Debian package maintainers

- Automate compilation of software packages
- Completion (man pages, description)
- Dissemination to 11 architectures by autobuilders

Community

- Package maintainers come directly from the users community
- Authentication as decentralised *chain of trust*
- QA by *homogeneity of platforms* and reporting system
Classification by Debtags

Facets based:
- accessibility
- admin
- devel
- field
- game
- hardware
- interface
- junior
- legacy
- mail
- network
Automated Builds of Debian Packages

Problems
- Not all packages installable (not yet existent, disk space)
- Order of packaging (inter-dependencies)

Solution: Debian’s pbuilder
-_deps resolved dynamically
- Planning build order

Public Debian Package Repository

Local Debian Package Repository

Automated Builds

R Source Repository
Debian as *Lingua Franca*

- Automated provisioning of packages for 11 Platforms
- Automated installations
- Detailed descriptions
  - **Formal**: Debtags
  - **Verbose**: Package descriptions

---

**Package**: `bash`
**Priority**: required
**Section**: shells
**Installed-Size**: 1848
**Architecture**: i386
**Version**: 3.1dfsg-8
**Depends**: base-files (>= 2.1.12),
**Suggests**: bash-doc
**Size**: 872884

**Description**: The GNU Bourne Again Bash is an sh-compatible command language interpreter that executes commands read from the standard input or from a file. Bash also incorporates useful features from the Korn and C shells (ksh and csh).

**Tag**: implemented-in::c, interface::shell, role::program, scope::utility, uitoolkit::ncurses

---

Möller, Bayer, Vernazobres, Gebhardt, Eddelbüttel

Scientific Grid Computing via Community-Controlled Autobuilding of Software Packages Across Architectures
Conversion of biocView vocabulary to Debtags

1. Selected nodes in biocViews tree are annotated with Debtags
2. Packages receive all Debtags associated with referenced biocView entries
Traditional Grid *Runtime Environments*

- A much respected special feature of the ARC grid middleware:
  1. Site-maintainers install a particular software for all worker nodes
  2. Software installation is promoted via Grid Information System
- Job descriptions explicitly mention required runtime environments
- Runtime environments are organised via a web site: http://gridrer.csc.fi/
List of Runtime Environments

The first entry for each RE is the reserved name of the RE, and the version number of the latest release. Other available versions are documented on the RE's Homepage.

<table>
<thead>
<tr>
<th>RE Name</th>
<th>Description</th>
<th>RE Homepage</th>
<th>Status</th>
<th>Last update</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPS/BIO/JASPAR-CORE-1.0</td>
<td>JASPAR-CORE</td>
<td><a href="http://www.grid.tsl.uu.se/RTEs/JASPAR-CORE/">http://www.grid.tsl.uu.se/RTEs/JASPAR-CORE/</a></td>
<td>Available</td>
<td>2006-09-02</td>
</tr>
<tr>
<td>APPS/BIO/LAGAN-1.2</td>
<td>LAGAN</td>
<td><a href="http://www.grid.tsl.uu.se/RTEs/LAGAN/">http://www.grid.tsl.uu.se/RTEs/LAGAN/</a></td>
<td>Available</td>
<td>2006-09-21</td>
</tr>
<tr>
<td>APPS/BIO/TFBS-0.5.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Automated Grid Runtime Environments

Additional development seeded in Lübeck

1. Software packages are organised in Catalogs
   ▶ RDF description (architecture, debtags, dependencies)
   ▶ automated location-independent installation
   ▶ Reference to binary for download

2. Service at sites
   ▶ install requested runtime environments on demand
   ▶ purge legacy installations
Conversion from Debian to Runtime Environments

**Current Implementation: Debian → Tar files**

- Script retrieves files to be repacked as tar file
- Automatically prepared install script to set environment variables for R
- No support for dependencies to *non-R* Debian packages

**Alternative: Virtualisation**

- Preparation of image for virtualisation
- Directly functional for all Debian packages
- ETA: 6 months
Deciding on the Eligibility of Packages

- Users positively select single packages for their computation
  - the selection of an R package is perceived as an integral part of the scientific application and
  - not specific to Grids
  - the selection is always a positive selection

- Site administrators
  - select classes of applications/libraries
  - both positively (ok to install) and negatively (not of interest) using
    - regular expressions or
    - SPARQL queries on Catalogs
Catalog of R packages for the Grid

- 1700 Packages are made available as Grid Runtime Environments

- Complete automation of software updates
  1. from Community to Debian
  2. from Debian to Grid

- Presentation
  HTML to users
  RDF to machines
RDF Catalog of Runtime Environments

RDF Representation

RDF triplets in the Catalog

kb#1
- kb#name
- ns#type
- kb#instance
- kb#description

kb#2
- kb#depends
- ns#type
- kb#url

kb#3
- kb#depends
- ns#type
- kb#url

kb#4
- kb#depends
- ns#type

kb#MetaPackage
- kb#name

kb#TarPackage
- kb#name

kb#BaseSystem
- kb#name

kb#instance
- http://. . . /weka-3.4.8.tar.gz

kb#depends
- http://. . . /jre_1.5.0.10-1.tar.gz

kb#basesystem
- Debian Sid

WEKA
- APPS/BIO/WEKA

Möller, Bayer, Vernazobres, Gebhardt, Eddelbüttel

Scientific Grid Computing via Community-Controlled Autobuilding Across Architectures
SPARQL for the retrieval of packages

SPARQL is an intrinsic component for the retrieval of information from RDF files:

- Retrieval of packages in catalog
- Request for a constraint to match

```
PREFIX kb: <http://knowarc.eu/kb#>
SELECT ?url
WHERE {
  ?id kb:name "APPS/BIO/WEKA" .
  ?id a kb:MetaPackage .
  ?package kb#url ?url .
  ?basesystem kb#name "Debian Sid" .
}
```
Arbitrary constraints can be implemented:

- Maintainer of package (Virtual Organisations, ...)
- Category of software
- Access to software
- ...

SPARQL for the retrieval of packages II
Why are you Preparing for Heterogeneity and Complexity in the Grid

1. Allow for a heterogenous set of users
2. Allow for complex interactions
   - User-driven modelling of workflows (Taverna, ...)
   - Automated Agents for cross-site communication
Strengths of RDF exploited

RDF is not essential for the current functionality, Debian provides core facilities today, but

1. it can be extended easily
   ▶ more attributes
   ▶ for more categories

2. database-like features
   ▶ catalogs/ontologies are easily associated with entries
   ▶ query language

3. it is a standard
Summary

- Integration of heterogeneous communities for Grid Computing
- Reference to software categorisation from within RDF Catalogs
  - No single system - allowing for multiple communities
  - Basis for decision of eligibility of packages for installation

Outlook

- Virtualisation: Mainstream Linux → Grid
- Complex workflows on the Grid
Acknowledgements

Grid Computing

▷ The KnowARC EU project and
▷ the NorduGrid at large (www.nordugrid.org)

Debian

▷ pkg-bioc Community (pkg-bioc.alioth.debian.org)
▷ Maintainers of alioth.debian.org

R - CRAN - BioConductor - Omegahat

▷ all contributors