Debian for Computational Grids

Packages and the Dissemination of Know-How

Daniel Bayer Steffen Möller

University of Lübeck, Institute for Neuro- and Bioinformatics

DebConf 7, Edinburgh
June 2007

KnowARC

Outline

What is a Grid?

Ian Foster (2002) proposed:

"A Grid is a system that

- *coordinates resources that are not subject to centralized control . . .*
- *. . . using standard, open, general-purpose protocols and interfaces . . .*
- *. . . to deliver nontrivial qualities of service."*[1]

KnowARC

[1] Foster, I.: *What is the Grid? A Three Point Checklist*

NorduGrid and ARC

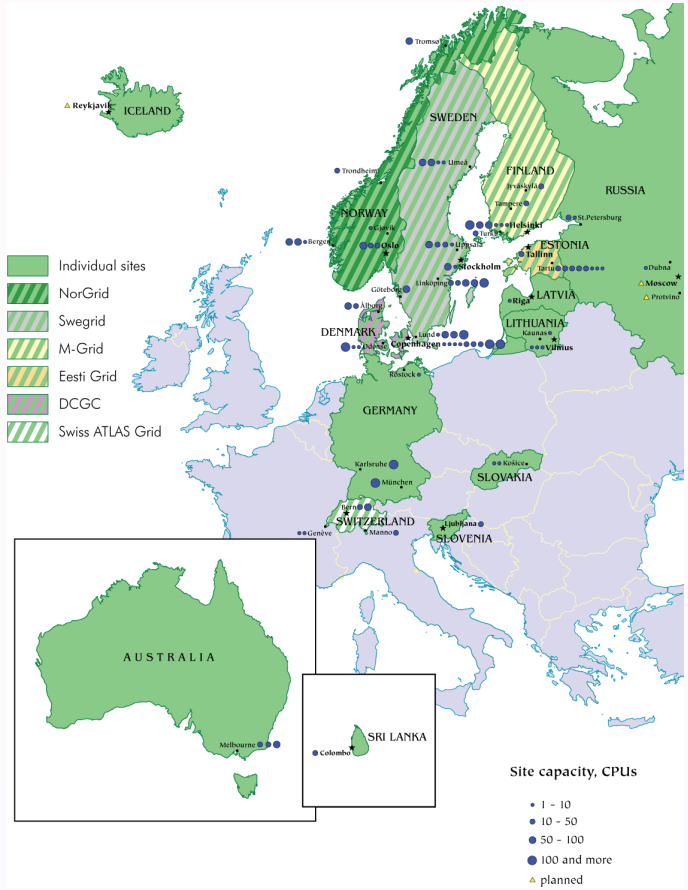
Nordugrid:

- is a compute and data sharing grid
- was launched in 2001
- has $>$ 7000 CPUs
- developed own middleware: ARC

Special features of ARC:

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

Details on `http://www.nordugrid.org`



Sites running ARC

KnowARC

How the ARC works

- The Grid's Compute Elements are batch systems
- Grid Information Systems inform about
  - architecture
  - memory
  - available software
- Jobs
  - find suitable execution environment
  - additional data downloaded to target site

Users of Computational Grids

Physicists:

- Many interconnected experiments with huge datasets
  ⇒ CERN's ATLAS project (`http://www.atlas.ch`)

Biologists:

- Many smaller independent projects
  - DNA and Protein Sequence analysis
  - Modelling Protein-Structures and -Interactions
  - Statistical genetics

Others: Computer scientists, Astronomers, Chemists, ...

Embedding Grid Computing in Workgroups

Large groups:

- Extension of local batch system
- Reduces compute time, not essential

Small groups:

- Provisioning of infrastructure
- Community of colleagues on grid become essential for exchange of knowledge
  - Software
  - Data
  - Workflows
- Often only a few, overwhelmed IT-people.

KnowARC

Debian in Science

Debian-Science:

- Melting pot of people contributing to research

Debian-Med:

- CDD with software for microbiology, bioinformatics, . . . , management of clinical practices

These projects provide

- heterogeneous set of established software solutions and
- mailing lists

Thus integrating different working groups.

Why Using Debian?

Technical Reasons:

- no hunting for packages
- security updates for many years
- superior package system
- it just works

KnowARC

Why Using Debian?

Technical Reasons:

- no hunting for packages
- security updates for many years
- superior package system
- it just works

Non-technical Reasons:

- helpful community
- personal preference

KnowARC

Why Using Debian?

Technical Reasons:

- no hunting for packages
- security updates for many years
- superior package system
- it just works

Non-technical Reasons:

- helpful community
- personal preference

But there are some Grid Communities requiring the use of a non-Debian OS.

Deploying ARC

A system running ARC needs:

- Globus Toolkit, which contains:
  - OpenLDAP
  - Cyrus SASL
  - Globus Replica Catalog
  - Globus RLS
  - OpenSSl (optional)
- Grid Packaging Tools (GPT)
- VOMS (optional)
- some OS tools and libraries

Deploying ARC

A system running ARC needs:

- Globus Toolkit, which contains:
  - OpenLDAP
  - Cyrus SASL
  - Globus Replica Catalog
  - Globus RLS
  - OpenSSl (optional)
- Grid Packaging Tools (GPT)
- VOMS (optional)
- some OS tools and libraries

What about packages?

Packaging ARC

Some problems arise:

KnowARC

Packaging ARC

Some problems arise:

- Globus
  - non-standard buildsystem (using GPT)
  - comes with own OpenLDAP
  - not designed to be integrated into distributions

Packaging ARC

Some problems arise:

- Globus
  - non-standard buildsystem (using GPT)
  - comes with own OpenLDAP
  - not designed to be integrated into distributions
- sometimes specific versions of libraries are needed
  - openssl (= 0.9.7)
  - gSOAP (more complicated)

Packaging ARC

Some problems arise:

- Globus
  - non-standard buildsystem (using GPT)
  - comes with own OpenLDAP
  - not designed to be integrated into distributions
- sometimes specific versions of libraries are needed
  - openssl ($= 0.9.7$)
  - gSOAP (more complicated)
- License Issue:
  - ARC is licensed under GPL but links against openssl.

Using Debian on the Worker Node

Much more interesting is Debian on the Worker Node:

- Jobs are send via a local batch system to Worker Nodes.
- Jobs need applications. Must be preinstalled on Worker Node.
- Local sysadmin can not install all the software manually.

Idea: Usage of

1. Debian as OS and
2. Debian Packages for providing software.

Current Situation

To deploy a new/updated RTE on a grid site the admin has to:

- install the Software either on all Worker Nodes or on a shared medium
- prepare a runtime script which is sourced by the job to enable the RTE

The Middleware:

- periodically searches for runtime scripts and
- announces the availability of RTEs

The Client:

- searches for grid sites which provide all needed RTEs and
- submits the job, requesting the needed RTEs

KnowARC

Current Situation

RTEs are organised via a web site: `http://gridrer.csc.fi/`

Runtime Environment Registry

This registry contains a list of Grid Runtime Environments (REs) and serves the members of NorduGrid, Nordic DataGrid Facility, SweGrid and M-grid and other Grid projects using NorduGrid ARC middleware.

For more information, see:

- Introduction to Runtime Environments and Registry.
- How to use RE (users)
- How to install RE (site admins)
- How to set up and maintain RE (RE maintainers)

The concept of the RER itself is under development. Please contribute through the nordugrid-discuss@nordugrid.org mailing list or mail directly to RER maintainer.

List of Runtime Environments

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

APPS/BIO/JASPAR-CORE-1.0	
Description:	JASPAR-CORE
RE Homepage:	http://www.grid.tsl.uu.se/RTEs/JASPAR-CORE/
Status:	Available
Last update:	2006-09-02
APPS/BIO/LAGAN-1.2	
Description:	LAGAN

Current Situation

List of Runtime Environments

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

APPS/BIO/JASPAR-CORE-1.0	
Description:	JASPAR-CORE
RE Homepage:	http://www.grid.tsl.uu.se/RTEs/JASPAR-CORE/
Status:	Available
Last update:	2006-09-02
APPS/BIO/LAGAN-1.2	
Description:	LAGAN
RE Homepage:	http://www.grid.tsl.uu.se/RTEs/LAGAN/
Status:	Available
Last update:	2006-09-21

KnowARC

Automated RTE Deployment: The Janitor

The Janitor

- installes RTEs automatically if they are requested by a job.
  $\Rightarrow$ Dynamic RTEs
- uses a RDF based Catalog which describes how to deploy a Dynamic RTE on a specific system.
  $\Rightarrow$ the Dynamic RTEs itself are abstract
  $\Rightarrow$ site admins specify constraints for RTEs to be eligible for installation
- removes unused dynamically installed RTEs if they are not needed anymore.
- integrates with the middleware in such a way that it is transparent.

KnowARC

The Pieces of a Dynamic RTE

A dynamic RTE consists of

- a uniq name (containing the version number),
- a description and
- the information how to deploy.

Easiest way to deploy is using a `tar`-package which contains

- the needed software,
- an `install`-script and
- the runtime script.

Dependencies between Dynamic RTEs and between `tar`-packages are supportet

Advantages and Disadvantages of this Approach

Advantages:

- `tar`-packages are easy to create and
- easy to install on otherwise unmodified systems
- no problmes with Conflicts (`tar`-packages are enabled by runtime script)

Disadvantages:

- Dependencies on system software are not possible.
- How to install Dynamic RTEs which needs a conflicting set of system software?
- Creating the `tar`-packages may be a lot of work.

Usage of Debian Packages

To solve this problems we intend to use Debian Packages and Virtualisation.

Idea: Dynamically create virtual machines fitting the jobs requirements and execute the job within the VM.

- Adding Debian Packages to a existing minimal VM image is easy:

  `fork()`, `chroot()`, `exec("apt-get")`.

  Also the Build Daemons solve a similar problem.
- A lot of packages are already available (Debian Science etc.)
- Dependencies on system software are handled automatically.
- Because of virtualisation there are no conflicts.
- But some overhead. $\Rightarrow$ Caching of created VM images is necessary.

KnowARC

Current State

- ARC middleware is used in production since 2001
- Ongoing development
  - Framework for verification of RTE installations
  - Autobuilding of Debian packages
  - . . . web services . . . workflow managment . . .
- Wanted!
  - Technical: A batch system which knows about Xen.
  - Social: Open communities for Grid computing

Conclusions

1. Debian renders Grid computing a commodity
   - provisioning of middleware to users
   - establishment of runtime environments to servers

2. Grid computing embraces (and extends) many principles of the Debian Society
   - shared access to resources (computing, storage)
   - trust network for access control

KnowARC

Acknowledgements

NorduGrid and KnowARC people:

- Anders Wäänänen
- Balázs Kónya
- Frederik Orellana
- Ferenc Szalai
- Оксана Смирнова
- and more . . .

This work contributes to the EU project *"KnowARC"*

www.nordugrid.org, www.knowarc.eu

Manual Installation of RTEs

To manually install a RTE the site admin

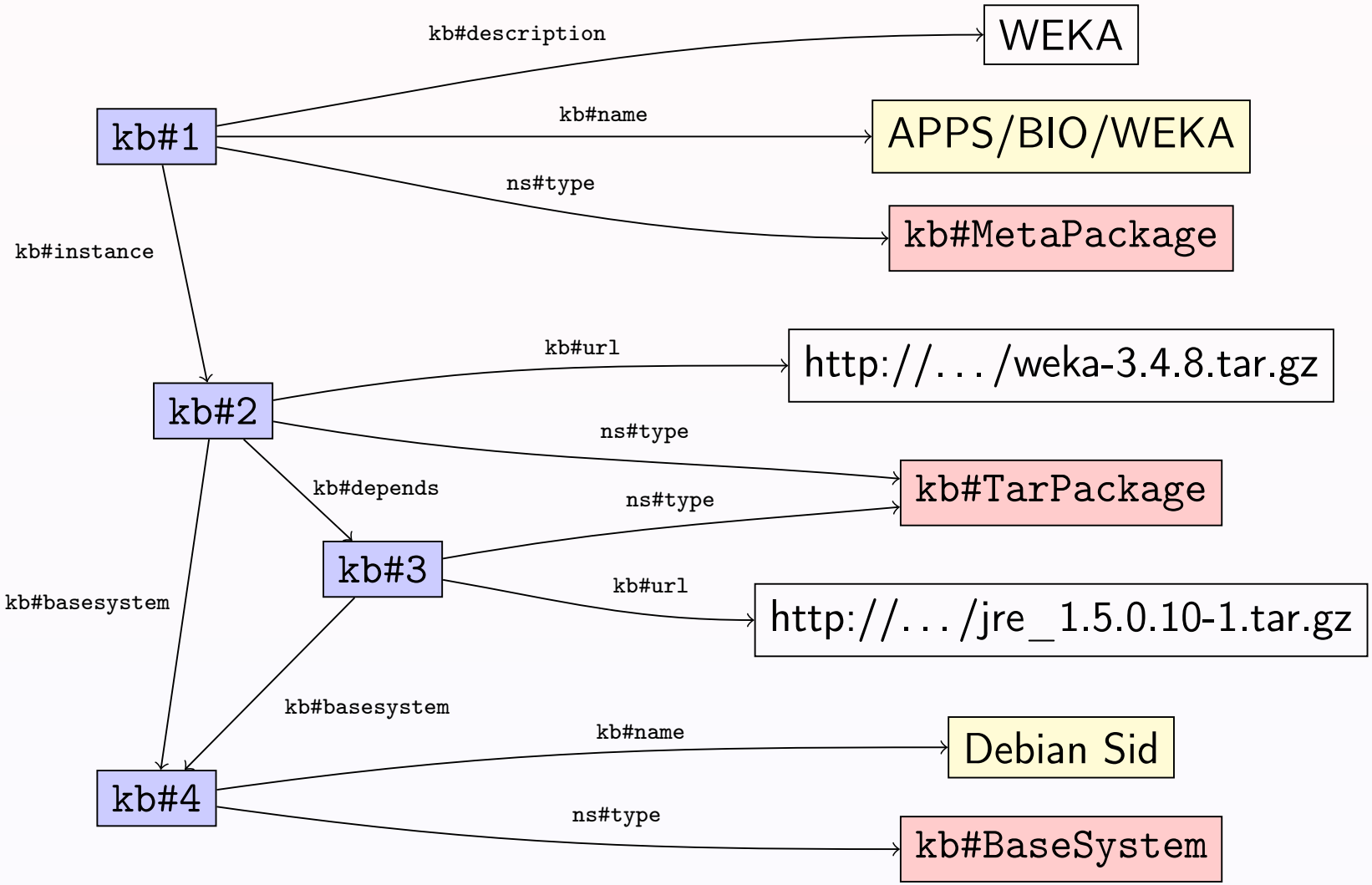
1. installs the needed software either
   - directly on all worker nodes or
   - on some medium shared by all worker nodes.
2. creates the *runtimefile* `$RUNTIMEDIR/$RTE_NAME` which is sourced in three different modes:

   Mode 1 before the job is submitted to the worker node
   Mode 2 called from worker node before job execution
   Mode 3 called from worker node after job termination

The Grid Infosystem looks for files in `$RUNTIMEDIR` and publishes the availability of RTEs.

KnowARC

Example: The APPS/BIO/WEKA Dynamic RTE



KnowARC