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

Steffen Möller*[1], Daniel Bayer[1], David Vernazobres[2], Albrecht Gebhardt[3], Dirk Eddelbuettel[4]

[1]University of Lübeck, Institute for Neuro- and Bioinformatics, Ratzeburger Allee 160, 23538 Lübeck, Germany
[2]The Westphalian Wilhelms University of Münster, Institute for Evolution and Biodiversity, Division of Bioinformatics, Münster, Germany
[3]Universität Klagenfurt, Institute for Statistics, Universitätsstr. 65-67, 9020 Klagenfurt, Austria
[4]Debian Project, Chicago, IL, USA

Email: Steffen Möller*- moeller@inb.uni-luebeck.de;

*Corresponding author

Abstract

Scientific communities have proven successful in developing highly adapted yet flexible information technology infrastructures for research. Genome research projects and high energy physics have lead the way in the collection, analysis and sharing of data at ever increasing volumes. On the other hand, sharing of hardware resources was introduced only relatively recently with the advent of grid technologies. A challenge remains to exchange operating-system- or hardware-dependent libraries across all nodes of a heterogeneous computational grid.

This paper presents a solution that utilizes unique features of the community-driven Debian Linux project. The Debian auto-builders prepare binary packages for 11 platforms and provide a well-established source for general purpose as well as scientific software. Respective libraries in scientific communities shall be collected, transformed and annotated for a platform-spanning application on computational grids. The feasibility of this approach is demonstrated via the fully automated creation of over 1700 packages for statistics and bioinformatics, based on package sources from the two main source repositories for the R language: CRAN, Omegahat and BioConductor.Org.

Availablility

http://pkg-bioc.alioth.debian.org

Abbreviations

ARC Advanced Resource Connector, **CRAN** Common R Archive Network, **DRE** Dynamic Runtime Environment, **QTL** Quantitative Trait Locus, **RDF** Resource Description Framework, **SNP** Single Nucleotide Polymorphism, **VO** Virtual Organisation

1 Introduction

Research departments in industry, as well as academic groups within and across research institutes, often share computational resources. These resources may be shared across space or time, and can be deployed in a distributed computing setup. We can consider these groups as ad-hoc *communities*. More formally, a community may present a *virtual organisation* (VO), and membership may be aligned with particular duties [1, 2]. Though without a strong common ground among the organisation's contributors, e. g. related research aims, the active maintenance of the collaborative infrastructure may stall. Absent any imposed directions by the VO, the community participants may have heterogeneous software- and hardware-setups that will typically be too diverse to be centrally organised. However, homogeneity can hardly be enforced without compromises.

If large homogeneous groups are needed to form successful computational grids, then the number of grid-savvy communities (and consequently of users and scientific problems) is severely constrained. With hindsight, to find today's large grids dedicated to research in particle physics is not surprising given the degree of homogeneity, which seems a natural consequence of the shared huge investments for their experimental setups[1]. Other research communities (e. g. in bioinformatics) control medium-sized to large batch systems. Because of their relative heterogeneity (as shown by the variation among software programs and databases), these benefit far less directly from inter-institutional links. A grid with heterogeneous users will consequently become segmented into smaller homogeneous segments of more strongly cooperating groups. These may not use the grid to its full potential. Hence, new technologies are required that ease the addition of software to these computing grids.

To address these issues, this paper applies techniques from the Open Source community in order to create cross-platform runtime environments for computational grids. This allows for automatic installation of software or databases. An implementation entails both technical and social requirements:

- an operating system that is available for all computer architectures of the grid,
- trusted developers or virtual organisation members who prepare the automated packaging,
- automated packaging for all participating platforms with a feedback mechanism for the developers,
- a community of users, possibly also outside the grid, that can test these packages on each platform and are willing to report problems,
- fully automated downloads, unpacking, installation and configuration of packages.

Thus, to overcome the given heterogeneity of the computing platforms, automated cross-platform builds and installations of software packages are performed. This work summarises the technology, development steps and selected work flows for a grid-enabled suite of libraries for the statistics environment R [3] as provided by the CRAN, BioConductor [4] and OmegaHat [5] initiatives.

2 Methods

The operating system used is Debian Linux. It supports more than ten different hardware architectures, or platforms, and, along with its derivative distributions such as Ubuntu, may have the largest number of installations and users.[2] Debian pioneered the community effort to create a complete Operating System. Among Linux distributions, Debian was the first to match all of the implementation criteria listed in the previous section[3] which makes it a natural platform for our work.

[1] CERN's GridCafé site at http://gridcafe.web.cern.ch presents an overview on the launch of grid technologies in the European research communities.

[2] While this is arguably hard to measure, we can consider the long-running Linux Counter (at http://counter.li.org/reports/machines.php) which shows a share of 20% for Debian with an additional 11% for Ubuntu/Kubuntu giving over 30% for Debian whereas Red Hat and Fedora Core add up to around 15%.

[3] http://www.debian.org/doc/manuals/project-history/

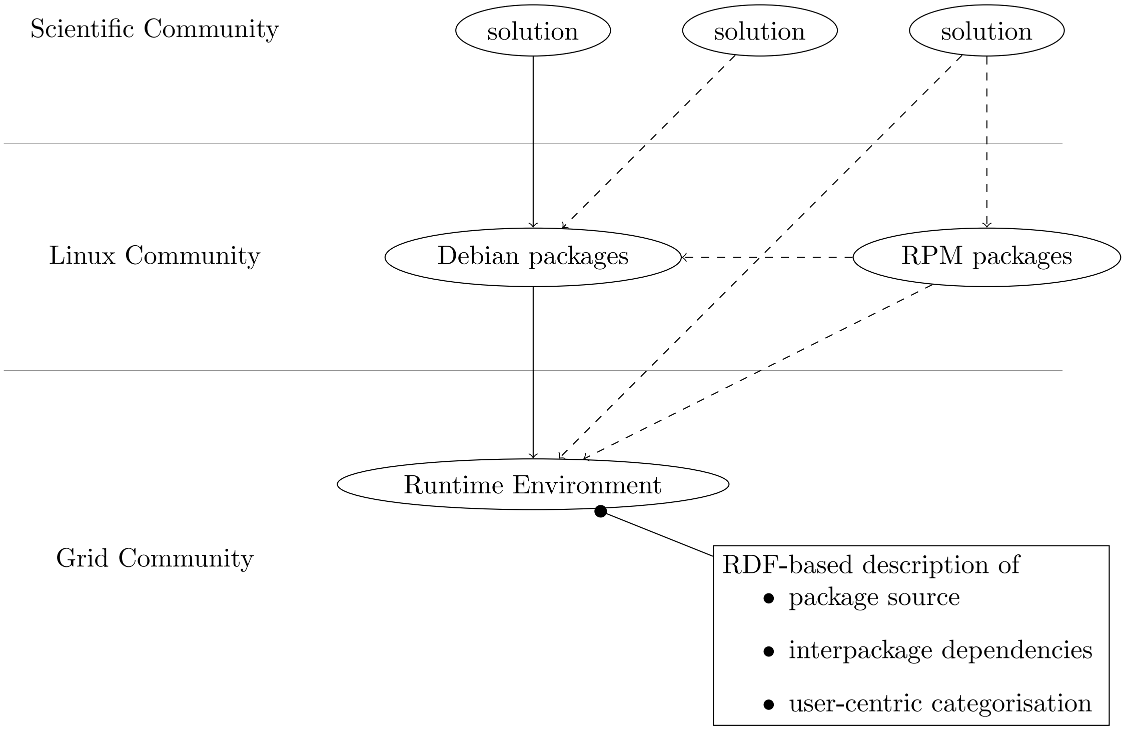

Figure 1: **Communities and the transition of packages for scientific grid computing.** *The paper demonstrates that an automated transition (solid arrows) from the packages of a scientific community towards homogeneous runtime environments in grid computing is achievable. Other routes yet comprise manual interventions (dashed arrows). The RDF-formatted catalog of available runtime environments integrates manual and automated means for the provisioning of package. It is essential to automatically decide on the eligibility of a package for a particular site and offers instructions for an unsupervised installation.*

Social networking Special interest groups within the Debian Linux distribution communicate via mailing lists and wiki pages (based at http://wiki.debian.org). Several large sub-communities have created so-called Custom Debian Distributions. They provide an even finer-grained distribution of Debian suitable for particular communities or groups within the larger universe of Debian users.

Individuals authenticate themselves and their work to the community via asymmetric cryptography. The trust 'network' between individuals is established via a graph that links individual members. Any group of individuals, independently from being accepted as developers of the main Debian distribution, can prepare a dedicated repository server that hosts user-signed packages. Also, each such group can decide on a set of users whom to trust in preparing these packages.

Provisioning of Debian Packages The homogeneity of Debian is achieved by the automated building and error reporting mechanisms of the distribution. In cases of build failures Debian developers have access to machines of all supported platforms for investigation and a helpful community to assist.

Special interest groups that are inclined to distribute software outside the Debian's main distribution can setup their own auto-builders. For this example, we consider the CRAN, BioConductor and Omegahat sets of packages for R. A set of programs and scripts prepared by the authors of this paper, along with other contributors of the pkg-bioc project[4], provides the infrastructure for automated builds of all these packages. The package maintainer only controls the logs of the the build process and may possibly decide to amend a

[4] See the wiki page http://wiki.debian.org/AliothPkgBioc and the project page http://alioth.debian.org/projects/pkg-bioc/.

package or its description.

Integration with Computational Grid All packages were made available for the recently developed *dynamic runtime environments* (DRE) [6] for the Globus-based [7] Advanced Resource Connector (ARC) grid infrastructure [8] of the NorduGrid[5]. For the work presented here, Debian packages are transformed automatically into runtime environments for the ARC middleware. In this process, a Perl script transforms each Debian package into a compressed 'tar' archive file. The information from the respective package's control file is used to formulate the description of the package for the grid's software catalog. This catalog is based on RDF and used by the ARC middleware for deployment and by various tools for displaying informations about available DREs to the user (Figure 1).

3 Results

As of March 2007, the repositories of CRAN, BioConductor and OmegaHat offer approximately 2170 different source packages. Several of these may be platform-specific and not operational under Linux, or may require specialised third-party development libraries (e. g. for particular hardware, or commercial software).

Using the tools described in the previous section, we were able to create automated builds of over 1700 of these source packages. It should be noted that the package building process in R provides the means to automatically check the built packages for correctness in several aspects via the `R CMD check` command. This also incorporates both explicit regression tests (where included in the source package) as well as implicit tests using sample code that is automatically extracted from the examples in the package documentation. Debian packages generated from CRAN source packages strive to follow the (draft) Debian R Policy [9]. Moreover, it is possible to generate those Debian packages in an automatic way given the CRAN package sources and the Debian build tools [10].

From these 1700 Debian packages, we create a corresponding set of packages which are installable as dynamic runtime environments (DRE) by the ARC grid middleware. Upon job submission, the grid middleware checks if all required runtime environments are installed and dynamically installs missing components and packages. For a proof of principle, an installation on a shared NFS directory was performed. Thus the packages are immediately available on all compute nodes of a cluster. This approach does not require superuser privileges as needed for chroot calls, or any explicit kind of virtualisation. Every package is installed in a separate directory, and will not influence the execution of other grid jobs.

The integration process directly mirrors dependencies between packages. These are declared at the R source level and are mapped into Debian package dependencies and further into dependencies between the DRE 'tar' packages. For CRAN and BioConductor, the vast majority of dependencies does not require packages from the 'main' section of the Debian distribution. While the procedure of a conversion is automated for libraries and R packages by amending environment variables, this approach is not generally applicable for arbitrary packages. The problem is not the actual provisioning of libraries but a dependency to Debian packages that are either not yet available as grid runtime environments or require to be preinstalled on all compute nodes.

The grid management unit for runtime environments uses RDF to describe its packages. This way, manually provided runtime environments, which may describe both hard- and software, and dynamically installed packages are specified in one coherent format. For our transformation of Debian packages into directly installable tar files, the control files of the package were transformed into RDF and subsequently merged with the catalog of runtime environments. If desired, the newly transformed files are directly available to all users of the grid as runtime environments.

An important feature of the catalog is the support of tags. Each DRE may have multiple associated tags. So administrators of grid sites may not only filter eligible packages by name but also by tag. This allows the enforcement of site-specific policies. As the catalog is based on RDF, it is even possible for the administrators to add additional tags locally. The initial set of tags is, if available, copied from the Debian

[5] http://www.nordugrid.org

packages[6] during the conversion process.

4 Discussion

The suitability of Debian for a cross-platform setups in distributed computing was previously discussed for agent technologies [11]. Here, the scope is taken further for grid computing and the setup of environments external to the Debian main distribution.

Interaction between communities Particularly appealing is the direct adaption of the infrastructure behind the package building process for the benefit of grid computing. The wealth of available software for bioinformatics brings with it a considerable load for the management of software updates that is difficult to cope with if the burden is not shared in communities. For larger groups this may be less of a problem. However, on the other end, self-trained technical assistants in wet-labs may have to create most of the required packages – and can therefore benefit from the automated provisioning of readily-useable package within the Debian framework.

Virtual Organisations were previously discussed in the context of interacting partners in e-Commerce [2]. Collaborating groups in open source communities show strong parallels: anchoring in the infrastructure, cybernisation, interoperability, provisioning of special products, and aggregations in web sites or social contracts. Noteworthy extensions are (i) *increased decentralisation* in administration and the provisioning of resources, (ii) *openess* for software and subcommunities with special interest, (iii) dedicated *feedback loops* through bug reports, and (iv) agglomeration of *scientific know-how* and its dissemination in tutorials, software documentation or problem-specific discussions on mailing lists.

Integration of packages with Grid infrastructure The concept of embedding the products of a community Linux distribution for the grid is novel. This approach aims at providing the highest degree of flexibility with regards to available computational power, and would work well together with the Lattice project that combines computational grids with volunteer computing to further adopt CPUs for a project [12].

The automated build of Debian packages based on CRAN and BioConductor package source for the integration with the Debian infrastructure is fairly direct and straightforward. However, this does not scale for the whole of Debian and neither for very large numbers of parallel installations of runtime environments at a particular site. In this case the configuration could become too complicated with long lists in the environment variables, e. g. a new entry in `$LD_LIBRARY_PATH` for every package with a library. Packages with absolute paths hard-coded into the software are not relocateable at all in an automated fashion. Another disadvantage is that the system software needed by the requested DREs must be already installed. These can not be installed automatically as they might conflict with other, already installed system software, which is needed by other DREs.

A lot of recent work in the grid community aims at exploiting the advantages of virtualisation [13]. With worker nodes running in virtual machines the actual deployment becomes easier. It will not be necessary to transform the Debian packages into software packages for the ARC middleware. In this scenario the Debian packages themselves can be installed directly into the system directories. Thus there is no need to set environment variables. Another advantage of this approach is that the packages to be installed can depend on any other package provided by Debian which would be automatically pulled in during installation.

Applications Generally speaking, the use of grid technologies brings supercomputing facilities to the desktop of all researchers. The need for the computing capacity is driven by large-scale problems as for example

Combinatorics An algorithm is applied on a large list of items (e. g. all genes, pairs of proteins or gene features like SNPs) in any combination also with other data like patient and their phenotypes. If the different algorithms allow for data parallelism, the jobs can run independently with only the cost for increased data transfer.

Monte Carlo simulations A particular result is compared with thousands of similar runs with similar

[6] The DebTags project provides a scheme for the classification of software in the Debian archive, presented also on http://alioth.debian.org. Similarly, CRAN and BioConductor offer user-centric ”views” as subsets of packages.

randomised input, often permutations of the original input. The quantile of the in the distribution of the simulated efforts approximates the p-value of the result.

With combinatorics and Monte Carlo simulations being required together for a particular problem, i. e. statistical genetics with expression data and clinical phenotypes for the investigation of expression QTLs [14], local clusters tend to falter under the load. The demand of a load sharing across sites is imminent. Problems of combinatorical nature may also require a heterogeneous set of tools, which brings us back to the initial problem to form communities for the management of software for research.

5 Conclusion

Computational grids provide an infrastructure to foster the collaboration between scientific groups for the analysis of data. The system presented here further enhances the breadth and the depth as the software components can now be dynamically adapted to reflect the changing requirements of virtual organisations.

We have presented tools to further augment the already large pool of over 18,000 readily-available Debian packages with over 1,700 new packages drawn from the three main repositories for the R environment and system. This includes fully automated builds across up to ten different architectures, combined with automated conversion into DREs for the ARC grid.

The cooperative provisioning of software infrastructure follow naturally from the existing open source activities in scientific communities. Technologies from the semantic web describe sites, runtime environments and constraints for their dynamic installations, to seamlessly bring contributing sites together.

Acknowledgements

We would like to thank the contributors to the free software projects mentioned in this paper, particularly to all those behind CRAN, BioConductor.org, Omegahat and Debian.

References

1. Foster I, Kesselman C, Tuecke S: **The Anatomy of the Grid: Enabling Scalable Virtual Organizations**. *International Journal of High Performance Computing Applications* 2001, **15**(3):200–222.

2. Travica B: **Virtual Organization and Electronic Commerce**. *The DATABASE for Advances in Information Systems* 2005, **36**(3):45–68.

3. R Development Core Team: *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria 2007, [http://www.R-project.org]. [ISBN 3-900051-07-0].

4. Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, Hornik K, Hothorn T, Huber W, Iacus S, Irizarry R, Leisch F, Li C, Maechler M, Rossini AJ, Sawitzki G, Smith C, Smyth G, Tierney L, Yang JY, Zhang J: **Bioconductor: open software development for computational biology and bioinformatics.** *Genome Biol.* 2004, **5**(10):R80.

5. Temple Lang D: **The Omegahat Environment: New Possibilities for Statistical Computing**. *Journal of Computational and Graphical Statistics* 2000, **9**(3), [http://www.amstat.org/publications/JCGS/index.cfm?fuseaction=Lang2000].

6. Bayer D, Bhimdi T, Oechsler G, Orellana F, Wäänänen A, Kónya B, Möller S: *Specification of Dynamic Runtime Environments for Grid Computing.* [Submitted].

7. Foster I: **Globus Toolkit Version 4: Software for Service-Oriented Systems.** In *IFIP International Conference on Network and Parallel Computing*, *Volume 3779*, Springer-Verlag LNCS 2006:2–13.

8. Ellert M, Gronager M, Konstantinov A, Kónya B, Lindemann J, Livenson I, Nielsen JL, Niinimaki M, Smirnova O, Wäänänen A: **Advanced Resource Connector middleware for lightweight computational grids**. *Future Generation Computer Systems* 2007, **23**(2):219–240.

9. Bates D, Eddelbuettel D: **Debian R policy: Draft proposal** 2003, [http://lists.debian.org/debian-devel-0312/msg02332.html].

10. Bates D, Eddelbuettel D, Gebhardt A: **Using R on Debian: Past, Present, and Future** 2004, [http://www.ci.tuwien.ac.at/Conferences/useR-2004/abstracts/Eddelbuettel+Bates+Gebhardt.pdf].

11. Tille A, Möller S: **Debian-Med Free Software in Health Care**. In *MAS*BIOMED Workshop, AAMAS*. Edited by Admano G, Denzinger J, Martin A, Merelli E, Miles S, Tianfield H, Unland R, Utrecht, NL 2005:72–85, [http://people.debian.org/~tille/debian-med/talks/200507_biomed/index_en.html].

12. Bazinet AL, Cummings MP: *The Lattice Project: a Grid research and production environment combining multiple Grid computing models*, Tectum 2007 :to appear.

13. Keahey K, Foster I, Freeman T, Zhang X, Galron D: **Virtual Workspaces in the Grid**. In *Euro-Par 2005 Parallel Processing*, *Volume 3648*, Springer-Verlag LNCS 2005:421–431.

14. Rockman M, Kruglyak L: **Genetics of global gene expression**. *Nat. Rev. Gen.* 2006, **7**(11):862–872.