

SANDIA REPORT

SAND2007-7775

Unlimited Release

Printed December 2007

The Changing Scope of the Trilinos Project

Michael A. Heroux

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: <http://www.osti.gov/bridge>

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: <http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online>



**SAND2007-7775
Unlimited Release
Printed December 2007**

The Changing Scope of Trilinos

Michael A. Heroux
Scalable Algorithms Department
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185

Abstract

The Trilinos Project started approximately six years ago as a small effort to enable research, development and ongoing support of small, related solver software efforts. The “Tri” in Trilinos was intended to indicate the eventual three packages we planned to develop. Presently the Trilinos repository contains over 40 packages, a number of which are not solvers. Also, the Trilinos user base and platform availability has expanded beyond original goals and will continue to do so.

Obviously the size and scope of Trilinos has changed significantly from the original vision. We foresee that it will continue to change. In this document we discuss the current state of the project and its growth, and motivate the reasons for this growth and future directions.

Acknowledgements: The author would like to thank the ASC, LDRD and ASCR programs that have funded this work.

Table of Contents

1. INTRODUCTION	6
<i>1.1 Scope Changes: Three Areas</i>	<i>6</i>
<i>1.2 Trilinos from Two Perspectives</i>	<i>6</i>
2. SCOPE CHANGES	7
<i>2.1 What is a Trilinos package?</i>	<i>7</i>
<i>2.2 Broader Base of Trilinos Users</i>	<i>8</i>
<i>2.3 Broader Support of Platforms</i>	<i>9</i>
3. ARCHITECTURAL CHANGES	10
<i>3.1 Trilinos and Package Releases and Conceptual Aggregation</i>	<i>10</i>
<i>3.2 Trilinos Capability Leaders</i>	<i>11</i>
4. TRILINOS STRATEGIC GOALS	12
5. SUMMARY AND CONCLUSIONS	12

1. Introduction

The Trilinos Project started approximately six years ago¹ as a small effort to enable research, development and ongoing support of related solver software projects. The “Tri” in Trilinos was originally meant to indicate the eventual three packages we intended to develop. Presently the Trilinos repository contains over 40 packages, 32 of which are part of the Version 8.0 release. This release and the next contain packages that are not solvers. Also, the Trilinos user base and platform availability has expanded and will continue to do so.

Obviously the size and scope of Trilinos has changed significantly from the original vision. We foresee that it will continue to change, and that the changes are positive. Trilinos has grown because it has provided an attractive environment for collaborative development of high-quality, state-of-the-art scientific software. The options to growth are either (1) separate infrastructure for related tools thereby having redundant and incompatible frameworks or (2) having fewer capabilities. If we can manage the growth of Trilinos well, it is surely a more attractive approach than these two options.

In this document we discuss the current state of the Trilinos Project, its growth, motivate the reasons for this growth and discuss future directions.

1.1 Scope Changes: Three Areas

The change in Trilinos scope is happening in three areas:

- **Capabilities:** Trilinos packages were traditionally solver capabilities and supporting components. Trilinos now contains non-solver packages such as Zoltan (which provides partitioning and load balancing capabilities) and Intrepid (which provides discretizations).
- **Customers:** The Trilinos customer base has traditionally been Sandia and other NNSA customers. Presently we are expanding to Office of Science applications, DoD, DHS, CRADAs and WFO.
- **Platforms:** Trilinos has been portable to all platforms for some time. However, we are focusing on easier-to-use installation tools with the goal of providing binary distributions on Windows and Mac.

Each of these areas will be discussed below.

1.2 Trilinos from Two Perspectives

In order to understand the motivation for recent changes to Trilinos it is important to realize that Trilinos plays two distinct roles:

¹ The first source repository commit was made on December 14, 2001.

1. Trilinos is a framework and collection of practical tools for collaborative software research and development.
2. Trilinos is a delivery mechanism for compatible software components with a reputation for providing high-quality, well-supported capabilities to its users.

When Trilinos contained only solver packages, these two roles were equally important and the distinction between them was seldom mentioned. Now the distinction is important.

From the perspective of a framework for R&D, adding packages has value simply from the reduced cost of support and ability to provide new software efforts with a high-quality development environment. The Trilinos package architecture has demonstrated robust support for growth in both size (number of packages) and scope (range of functionality provided). Packages are fully self-contained in the repository and each package has its own identity for mail lists, issue tracking and more. From the very beginning of the project we have respected the autonomy of packages and can still say that a package can remove itself from Trilinos at any time with almost no effort. Furthermore, the flat package architecture, where all packages are peers in the repository removes any preconceived bias to meta-packaging and allows us to dynamically or conceptually aggregate packages as needed, even creating sub-products or even products with different names.

From a delivery mechanism perspective, the large package count in Trilinos forces us to evolve how we manage, explain and deliver software to users. With the increase in packages and the broad diversity of capabilities, we are establishing additional leadership roles in the project, assigning capability leaders who will actively lead specific areas of development. We are also working to define subsets of packages that will be released together as well as a process for releasing individual packages.

2. Scope Changes

The scope of Trilinos is changing in three primary areas. The addition of new non-solver packages is most obvious, but the Trilinos customer base and platform emphasis is also expanding. In this section we discuss these three areas in a statement-discussion-advantages-challenges format.

2.1 What is a Trilinos package?

Statement: Trilinos packages can be any self-contained piece of software that is all of the following:

1. *Embeddable: The package can be incorporated into an application, typically in the form of a library or collection of templated code.*
2. *Reusable: The package is usable in more than one specific setting, even if reuse is only speculative initially.*

3. *Interoperable: The package can be used by other packages in Trilinos, can use other packages, or (preferably) both, even if interoperability is only speculative initially.*

Discussion: Expanding the scope of Trilinos beyond solvers, but limiting it to reusable, interoperable libraries makes sense at this time. From a framework perspective, whether or not a package is solver-related or not appears to be irrelevant, so expanding to other packages that have similar software traits is very easy. However, it seems that adding user-oriented applications (which are not embeddable), special-purpose packages (which are not reusable) or packages that only use Trilinos capabilities (not interoperable) is not prudent at this point.

Advantages: Using Zoltan as an example, the addition of existing packages into the Trilinos framework has the following advantages:

1. Reduced cost of Zoltan support.
2. Improved software environment for Zoltan developers.
3. Adoption of Zoltan best-practices by other Trilinos developers.
4. Opportunities for Zoltan to interact with other packages more easily, e.g., preconditioners can use Zoltan for better partitioning, and Zoltan can use Epetra tools for advanced partitioning algorithms.

Using Intrepid as an example, the addition of new non-solver packages into Trilinos has the same four advantages as existing packages, plus the additional benefit that Intrepid will be distributed with the full Trilinos release, giving immediate broad exposure² that might otherwise take years to develop.

With respect to work for others (see below), the package architecture can be used to segregate software pieces that have different intellectual property agreements from the rest of Trilinos.

Challenges:

1. As more software is developed within the Trilinos framework, innovation in tools and processes can be hindered since change will impact many people. We must continue to monitor and adopt new tools and process and invest in staffing to provide this support.
2. Package teams that are part of Trilinos must put an effort in developing and retaining an identity independent of Trilinos. However, this has always been a challenge and so far has been manageable.

2.2 Broader Base of Trilinos Users

Statement: *The Trilinos user base will expand to include DOE Office of Science applications, DoD, DHS, industrial collaborations and other work-for-others customers.*

² The Trilinos download rate is presently over 2500 per year and growing.

Discussion: Trilinos started as an ASCI-funded project and ASC (the successor to ASCI) continues to be a major funding source, especially for framework and support activities. At the same time, other funding sources have always been a part of the project since most solver algorithms R&D, regardless of funding, eventually made it into Trilinos at some point. Presently with the increased scope in capabilities, the increased popularity of Trilinos in the general scientific community and specific funding from DOE Office of Science and strategic industrial partners, the project has a very diverse funding profile that leads to multiple customers with multiple needs that must be effectively managed.

From the very beginning the Trilinos project has recognized the importance of a broad user base and has made its software available as open source, used open source tools, and collaborated with external research staff. Such collaborations are important for the health of the project and to amortize the cost of development for any particular customer.

Advantages: The wide acceptance of Trilinos as a viable R&D project, both within and outside of Sandia is critical to the health of the project and provides ongoing opportunities for Sandia to obtaining funding and generate new research results.

Challenges: A diverse user base can pose a risk that we lose site of our core customer base. As a result we must be vigilant and keep our priorities in proper order.

2.3 Broader Support of Platforms

***Statement:** Trilinos will build on Windows and Mac Platforms using native installation processes.*

Discussion: Some of our new customers, as well as our development team itself rely on common laptop and desktop systems for using and developing Trilinos capabilities. However we have traditionally been heavily biased to a Unix command line style of installation and integration of Trilinos. Some of our most recent users are completely unaware of Unix and our build processes are prohibitive to them. New build tools such as Cmake present an opportunity to support Windows Visual Studio and Mac Xtools environments in a natural way for developers that use these environments. We do not foresee eliminating our present configure-make approach, but will explore use of Cmake as a side-by-side build system.

Advantages: Installation of Trilinos has always been challenging and we are constantly looking for ways to make it easier. When starting from source, installation is inherently complex. We need access to C, C++, Fortran and Python. We need MPI in most cases, the BLAS and LAPACK and possibly many other third-party libraries. If we can provide the “typical” user with a pre-built collection of libraries for the most common platforms, we will essentially eliminate all of these complexities. Furthermore, environments such as Visual Studio use a very different style of programming and library use. Cmake gives us the ability to provide Trilinos in these environments in a way that is natural.

Challenges: We will need to maintain two build systems and manually keep them consistent. We will also be distributing Trilinos in more configurations. Because of this, we will move forward on this at a reasonable pace.

3. Architectural Changes

With the expanding scope of Trilinos and the large increase in package counts, we intend to add a layer to the architecture of Trilinos. The layer will be added to the distribution of package and to the leadership of the project. It will not affect the organization of the software repository. Again we use a statement-discussion-advantages-challenges format.

3.1 Trilinos and Package Releases and Conceptual Aggregation

Statement: *Trilinos releases will occur as follows:*

1. *Trilinos will continue to be released as a single collection of all packages that are mature enough for public exposure. Presently we see that these releases will happen approximately once a year.*
2. *Individual packages (such as Zoltan) will be released independent of other packages, as package teams see the need, and may be released more frequently.*
3. *Conceptually similar collections of packages will be presented on the Trilinos website in a combined fashion. Furthermore, over time we will develop releases of related packages as a single distribution. The release frequency is difficult to determine at this time.*

Discussion: For many years we have discussed the possibility of breaking up the release of Trilinos into multiple parts, and we have provided some packages that way. However, we have seen very little user interest in this approach. Most users do not mind getting all Trilinos packages in a single distribution since the distribution tarball is small by modern standards and because they can easily control which packages get built.

At the same time, mature packages such as Zoltan, which are frequently used independently, need to be available independently. Furthermore, we believe that, as the vertical software stack of Trilinos continues to grow, we will want to provide subcollections of packages for users who want only a subset of functionality. Presently we will work to provide all linear solver, preconditioner and support packages as a single collection. We will also work to focus on Stratimikos as the primary interface to these packages since Stratimikos provides a single common interface to all linear solver and preconditioner packages already.

As Trilinos grows, we think that the dynamics of Linux and its packages can be a good model to mimic. Linux distributions occur with a similar frequency to Trilinos and contain a collection of all packages that are certified to work well together. For some users, this distribution and its minor updates are sufficient. For other users, who are more aggressive and willing to deal with potential incompatibilities, more frequent updates such as the latest release of an individual package, or a subcollection of packages could be obtained and manually updated against a previous full Trilinos distribution.

Advantages: By providing three tiers of releases, we should be able to provide sufficient flexibility for development teams and users of all kinds.

Challenges: We will need sufficient support resources in order aid package teams in getting these releases out. We may need a better software system to manage package distribution, e.g., something like the Cygwin package manager.

3.2 Trilinos Capability Leaders

***Statement:** In order to more actively manage Trilinos capabilities, we will introduce a new layer of leadership focused on capability areas. Each capability leader, e.g., linear & eigen solvers, will be responsible for describing how relevant Trilinos packages can be used in the area. These leaders will also help to identify where Trilinos coverage needs to be improved within their capability area.*

Discussion: The capabilities that Trilinos provides are too broad for any single person to actively guide and manage. Furthermore, the large number of packages presents a challenge to users who are interested in particular capabilities. Coupled with package release aggregation, active capability leadership will help users understand how to use Trilinos to solve certain types of problems.

Although capability areas will change over time, our initial breakdown is as follows:

1. Framework & Tools.
2. Software Engineering Technologies and Integration.
3. Discretizations.
4. Geometry, Meshing & Load Balancing.
5. Scalable Linear Algebra.
6. Linear & Eigen Solvers.
7. Nonlinear, Transient & Optimization Solvers.

Each capability area will be accessible from the main Trilinos website and capability leaders will provide a discussion of how Trilinos can be used

Advantages: Capability leaders will have a high profile in the Trilinos project and will actively drive the project in new directions. They will also provide a critical resource to users by explaining which packages can be used to provide a specific capability. We foresee that capability leaders will also help to determine aggregation strategies for packaging and distributing parts of Trilinos.

Challenges: The addition of another leadership layer means more communication and coordination is required.

4. Trilinos Strategic Goals

With the expanded scope of Trilinos, the strategic goals of the project must also change. Fortunately, the changes are actually quite modest since the overall goals of the project from a scalable scientific software library perspective are still the same.

Algorithm Goals:

1. Scalable Computations: As problem size and processor counts increase, the cost of the computation will remain nearly fixed.
2. Hardened Computations: Never fail unless the problem is essentially intractable, in which case we diagnose and inform the user why the problem fails and provide a reliable measure of error.
3. Full Vertical Coverage: Provide leading edge enabling technologies through the entire technical application software stack: from problem construction, solution, and analysis to optimization.

Software Goals:

1. Universal Interoperability: All Trilinos packages will be interoperable, so that any combination of packages that makes sense algorithmically will be possible within Trilinos and with compatible external software.
2. Universal Accessibility: All Trilinos capabilities will be available to users of major computing environments: C++, Fortran, Python and the Web, and from the desktop to the latest scalable systems.
3. Universal RAS: Trilinos will be:
 - a. Integrated into every major application at Sandia (Availability).
 - b. The leading edge hardened, efficient, scalable solution for each of these applications (Reliability).
 - c. Easy to maintain and upgrade within the application environment (Serviceability).

5. Summary and Conclusions

The Trilinos Project has always been growing and expanding. In some sense the current growth is no different. However, the speed of growth and the expansion to new capabilities, new customers and broader platform support will challenge us to manage the project carefully. The purpose of this document is to raise the important issues and stimulate discussion about how we move forward on the project.

Comments and concerns are welcome. Please contact the author.

Distribution

1	MS 1318	D. E. Womble, 01410
1	MS 1318	J. R. Stewart, 01411
1	MS 0370	T. G. Trucano, 01411
1	MS 1318	M.S. Eldred, 01411
1	MS 1318	K. F. Alvin, 01414
1	MS 1318	A. G. Salinger, 01414
1	MS 1318	S. S. Collis, 01416
1	MS 1318	M. A. Heroux, 01416
1	MS 0382	H. S. Morgan, 01540
1	MS 0382	S. E. Gianoulakis, 01541
1	MS 0899	Technical Library, 9536 (electronic copy)

