CrowdLabs: Social Analysis and Visualization for the Sciences

Phillip Mates[1], Emanuele Santos[1], Juliana Freire[1], and Cláudio T. Silva[1]

SCI Institute, University of Utah, USA

Abstract. Managing and understanding the growing volumes of scientific data is one of the most challenging issues scientists face today. As analyses get more complex and large interdisciplinary groups need to work together, knowledge sharing becomes essential to support effective scientific data exploration. While science portals and visualization Web sites have provided a first step towards this goal, by aggregating data from different sources and providing a set of predesigned analyses and visualizations, they have important limitations. Often, these sites are built manually and are not flexible enough to support the vast heterogeneity of data sources, analysis techniques, data products, and the needs of different user communities. In this paper we describe CrowdLabs, a system that adopts the model used by social Web sites, allowing users to share not only data but also computational pipelines. The shared repository opens up many new opportunities for knowledge sharing and re-use, exposing scientists to tasks that provide examples of sophisticated uses of algorithms they would not have access to otherwise. CrowdLabs combines a set of usable tools and a scalable infrastructure to provide a rich collaborative environment for scientists, taking into account the requirements of computational scientists, such as accessing high-performance computers and manipulating large amounts of data.

Keywords: Computational Sciences, Cyberinfrastructure, Visualization

1 Introduction

The infrastructure to design and conduct scientific experiments has not kept pace with our collective ability to gather data. This has led to an unprecedented situation: data analysis and visualization are now the bottleneck to discovery. This problem is compounded as interdisciplinary groups collaborate and need to perform a wide range of analyses targeted to multiple audiences.

We posit that by facilitating the *social analysis of scientific data*, we can overcome many of these challenges. When users share their analyses and visualizations, they can benefit from the collective wisdom: by querying analysis specifications which make sophisticated use of tools, along with data products and their provenance, users can learn by example from the reasoning and/or analysis strategies of experts; expedite their scientific training in disciplinary and inter-disciplinary settings; and potentially reduce the time lag between data acquisition and scientific insight.

In this paper, we describe *CrowdLabs* , a system that adopts the model used by social Web sites and integrates a set of usable tools and a scalable infrastructure to provide

a rich collaborative environment for scientists. Similar to social Web sites, CrowdLabs aims to foster collaboration, but unlike these sites, it was designed to support the needs of computational scientists, including the ability to access high-performance computers and manipulate large volumes of data. By providing mechanisms that simplify the publishing and use of analysis pipelines, it allows IT personnel and end users to collaboratively construct and refine portals. Thus, CrowdLabs lowers the barriers for the use of scientific analyses and enables broader audiences to contribute insights to the scientific exploration process, without the high costs incurred by traditional portals. In addition, it supports a more dynamic environment where new exploratory analyses can be added on-the-fly.

Another important feature of CrowdLabs is the support for provenance [5, 8]. Publishing scientific results together with their provenance—the details of how the results were obtained, not only makes the results more transparent, but it also enables others to reproduce and validate the results. CrowdLabs leverages provenance information (*e.g.,* workflow/pipeline specifications, libraries, packages, users, datasets and results) to provide a richer sharing experience: users can search and query this information. In addition, provenance is made accessible through the Web site and an API. This allows users to connect results published in an article or wiki page to the pipelines and data served by CrowdLabs, greatly simplifying the creation of provenance-rich publications.

The remainder of the paper is organized as follows. We review related work in Section 2. In Section 3, we describe the main components of CrowdLabs. Information on deploying CrowdLabs at www.crowdlabs.org is given in Section 4. In Section 5, we describe the different ways of sharing content and of making reproducible documents using CrowdLabs. We conclude in Section 6, where we outline directions for improvements and future work.

2 Related Work

While there have been several efforts focused on sharing scientific data, relatively little work has gone into sharing analysis and visualization specifications (pipelines). To this end, closely related to our approach is myExperiment [13], a collaborative environment for sharing pipelines and other digital objects. myExperiment supports versioning of pipelines and can execute certain types of pipelines. However, because its focus is on pipelines that integrate bioinformatics-related Web services, myExperiment does not support data- and compute-intensive pipelines.

Recently, a number of sites have come online which aim to support social analysis and visualization of small, tabular data. Tableau Public[1] provides infrastructure for users to publish interactive visualizations on the Web. Many Eyes [18] and Swivel [17] (no longer available) are public social data analysis Web sites, where users can upload data, create visualizations of that data, and leave comments on either visualizations or datasets. Built upon Many Eyes is Many Eyes Wikified, which is based on Dashiki [11], a wiki-based Web site for collaboratively building visualization dashboards.

The ability to run and interact with compute-intensive pipelines or simulation jobs is rapidly becoming essential for most scientists. The HUBzero Platform for Scientific Collaboration [12] allows researchers to access and share scientific simulation and

[1] http://www.tableausoftware.com/public

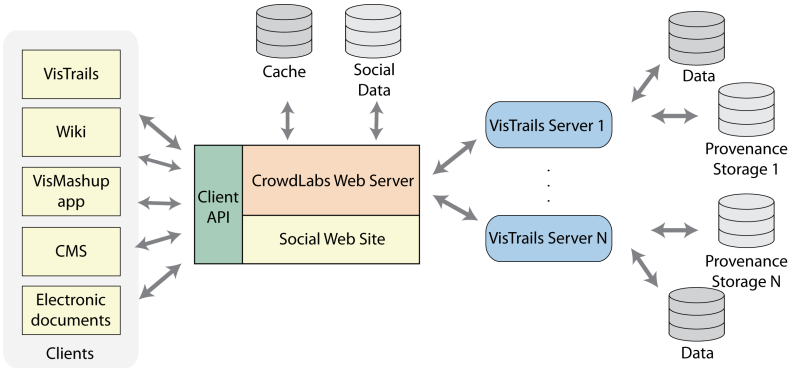

Fig. 1: CrowdLabs system architecture.

modeling tools. It was created to support nanoHUB.org [16], an online community for the Network for Computational Nanotechnology (NCN). To publish a tool, developers have to connect to a special workspace machine, where they compile their code and use NCN's open source toolkit Rappture to create friendly GUIs. End users access the resulting tools using an ordinary Web browser and launch simulation runs on the national Grid infrastructure using virtual machines, without having to download or compile any code. Similar to HUBzero, CrowdLabs enables the use of HPC resources and is flexible enough to be deployed on the cloud. But the use of the pipeline computation model allows it to provide additional functionality: besides the support for provenance and queries over this information [15], it is possible to deploy customized (and easy-to-use) Web applications [14].

3 System Overview

The CrowdLabs infrastructure is general and can be integrated with any workflow management system that runs in server mode and exposes an API. In this paper we describe how the CrowdLabs infrastructure is used with the VisTrails [2, 9] and VisMashup [14] systems. VisTrails is an open-source provenance management and scientific workflow system that was designed to support the scientific discovery process. VisTrails provides unique support for data analysis and visualization, a comprehensive provenance infrastructure, and a user-centered design. The system combines and substantially extends useful features of visualization and scientific workflow systems. For more details about VisTrails, please refer to [2, 9]. VisTrails was modified to provide access to workflow provenance in a client-server mode.

The VisMashup [14] system simplifies the creation, maintenance, and use of customized, workflow-based applications (or mashups). It supports the tasks required in the construction of custom applications: from querying and mining workflow collections and their provenance (for finding relevant workflows and parameters that should be exposed in the application) to automatic generation of the application and associated user interface, that can be deployed as a desktop or Web application. For use within CrowdLabs, VisMashup was extended to support a Web-based user interface for interacting with workflows.

The CrowdLabs architecture consists of two main components: *CrowdLabs Web Server*, which provides the *Social Web Site*, and a *Client API* and the *VisTrails Server*, which handles workflow-related tasks. The CrowdLabs architecture is depicted in Figure 1 and a description of each component follows.

3.1 CrowdLabs Web Server

Client API. CrowdLabs provides a Web-based interface for sharing workflows and provenance. The system includes a repository of workflow results (e.g., visualizations), datasets, and libraries, and while the CrowdLabs Web site provides a useful platform for sharing and collaboration, the social and provenance data can be useful in other contexts. In order to expose CrowdLabs resources to a diverse set of clients, the site employs a RESTful HTTP API [6]. This API identifies visualization and social resources, providing uniform resource identifiers (URIs) for clients to retrieve, add, update and delete them.

The data analysis and visualization resources defined by the system are vistrails, workflows, vismashups, packages, and datasets, while the social resources include profiles, projects, groups, and blogs. Adhering to the RESTful architecture, each of these resources has various different representations associated with them. Visualization resources might include provenance data, meta-data (modules, documentation, *etc.*), application files (vistrails, data files), as well as visualization results. Social resources might include blog posts, discussion topics, and notices, along with ratings, tags, and comments, which are linked to the visualization resources. For example, for a client to access the XML representation corresponding to the workflow with id 117, they would use the URI *http://www.crowdlabs.org/vistrails/workflows/get_xml/117/*.

The RESTful API enables basic CrowdLabs functionality to be integrated into the VisTrails desktop application and other extensions. Users can login, add, and update vistrails and datasets to CrowdLabs through the *Web Repository Options* dialog from within the VisTrails desktop application. The LaTeX extension described in Section 5 also uses the RESTful API for embedding workflows in PDF documents.

Following the example of myExperiment's Google Gadget and Facebook App [13], providing an API encourages developers to extend functionality and creates an open development environment.

Social Web Site. To foster user interaction, CrowdLabs incorporates a social Web site that is based around user-created content and social networking tools. Users can make friends, join groups, write blogs, and create projects, topics, and wikis. In addition, they can add, edit, and delete VisTrails related data such as vistrails, workflows, vismashups, packages, and datasets. Tied to each of these VisTrails related objects are ratings, tags, comments, and projects. This social data not only encourages an environment of user discussions and interaction, but enables the use of crowd sourcing to find good-quality visualizations through user ratings, better categorize datasets and workflows by user tagging, and troubleshoot problems through comments and discussion topics. We also let users share their work off-site by providing syntax to embed interactive vismashups on Web sites as well as static visualization results on the Web, wikis, and within LaTeX documents (see Section 5).

Cache. CrowdLabs is set up to generate content dynamically. This creates potential efficiency issues, since some workflows can take a long time to run. It is important to

avoid delays when presenting pages to users, otherwise they can get discouraged and avoid using the site. We use different forms of caching to speed up common operations.

In the CrowdLabs Web server there are two caches: the results cache and the provenance cache. The results cache is used to store images and other files generated by workflows and vismashups. When there is a request for a workflow execution result, for instance when a vismashup run, the system first checks if that workflow has been executed before. If so, it uses the files already in the cache, otherwise it will forward the request to the proper VisTrails server instance.

The provenance cache stores information about the workflows and vistrails uploaded to the CrowdLabs Web site. This information currently includes the named workflows in a vistrail, who created them and the packages and modules referenced in the workflows together with their documentation. The system takes advantage of the fact that VisTrails change-based provenance model records information about workflow evolution [9]: all the workflows are versioned.

3.2 VisTrails Server

The *VisTrails server* is one of the most important components of CrowdLabs. It provides the link between the workflow provenance and the rest of the system. The VisTrails server is a multi-threaded server that uses the XML-RPC protocol to answer client requests. The most common requests are: execute a workflow or a vismashup, add or remove a vistrail or a vismashup from the database, get the packages and modules used in a vistrail or workflow and other information associated with the workflows.

The CrowdLabs Web server communicates with VisTrails server instances via XML-RPC requests, enabling communication with multiple remote VisTrails servers. Scientific teams can thus host their own VisTrails servers as a way to meet their computing and data storage needs.

Another key feature of the VisTrails server is that it maintains its own cache (separate from CrowdLabs results cache) for keeping the results of executed workflows or vismashups. When both components are in the same machine, CrowdLabs can be configured to use the VisTrails server cache to avoid redundant storage. The VisTrails server also has the ability to start and communicate with other VisTrails server instances using the same XML-RPC protocol. This allows the creation of clusters of servers that work transparently with the rest of CrowdLabs.

4 Deploying CrowdLabs

Depending on the particular application, it is possible to use different deployment configurations for CrowdLabs (see Figure 2). Here we will describe the current system deployment at www.crowdlabs.org and explain some of its key capabilities. We encourage readers to access the site, but bear in mind that it is constantly under development.

System configuration. CrowdLabs is currently deployed as shown in Figure 2(a). The core system and the four instances of the VisMashup server share a 8-core Intel Xeon 2.66 GHz machine with 24 GBs of RAM running Linux. The CrowdLabs webserver cache and social data along with VisMashup workflow specifications and provenance are stored in MySQL databases. The CrowdLabs Web site is implemented using the Python Web framework Django and the VisTrails server is implemented in Python.

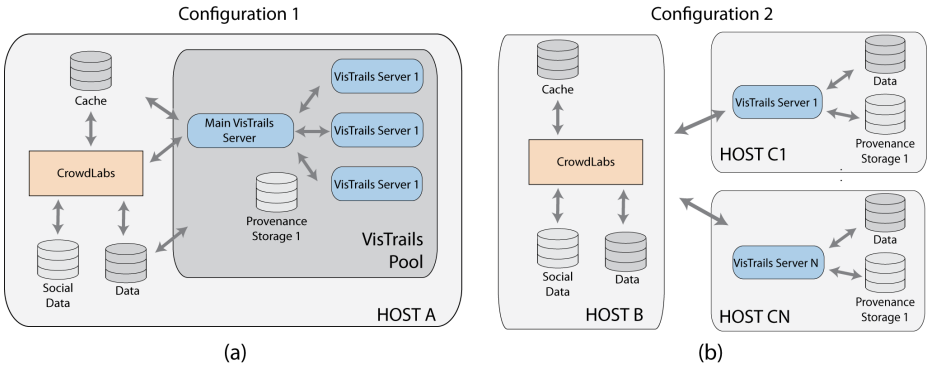

Fig. 2: Different configurations of deploying CrowdLabs. (a) All the components are located on the same machine. (b) VisTrails servers execute on dedicated machines.

Projects and servers. It is possible that users would like to organize their content into different projects. An example is the ALPS [1] project, which contains all the vistrails, vismashups, workflows and the information they use together with the discussions and blog posts created about them. This allows for defining different levels of visibility: Groups can have discussions and upload workflows that are only visible to the people involved in the project. The ability to selectively disclose information for people outside the group is extremely important for scientists, who may work for many years before deciding to release certain types of data. Another advantage is that a project can have its own dedicated VisTrails server. This creates the possibility of having specialized servers for different types of workflows. For example, for the ALPS project, we are in the process of deploying a VisTrails server on one of their machines so users on the CrowdLabs Web site can execute workflows and vismashups that access all the required resources on ALPS's file servers for running the simulations. These different servers allow the system to grow in functionality without compromising the overall performance.

5 Sharing Content and Supporting Reproducible Research

An important motivation for us to create CrowdLabs was to make it easier for scientists to publish provenance-rich, reproducible results. While it is widely accepted that scientific publications should include detailed provenance so that others can both reproducte and validate the results, in practice, doing so is challenging, both for authors and reviewers. Even when authors provide data sets and computer code, reviewers must configure their systems so that they can compile and run the code; they must also navigate between code, data and text, identify important parameters and manually enter the values specified in the text. Recently, the renewed interest on this subject in different communities has led to different scientific publishing approaches (see [7] for an overview). CrowdLabs simplifies the process of packaging workflows and results for publication. Authors can create documents whose digital artifacts (*e.g.,* figures) include a deep caption: detailed provenance information which contains the specification of the

computational process (or workflow) and associated parameters used to produce the artifact. CrowdLabs supports the publication on wikis, other Web sites, and scientific documents. Readers need not install any special software and can interact with the results through a Web interface. Next, we briefly describe the different mechanisms that CrowdLabs supports for sharing content.

Interactive versus static content. CrowdLabs supports the generation of static content (e.g., images, animations, tables, XML pages) as well as interactive ones. Static content is generated directly from the workflow specification. In particular, we use VisTrails' flexible spreadsheet infrastructure to generate the different types of output. Anything that can be displayed in a spreadsheet cell can be re-routed through CrowdLabs. Due to limitations of current browsers, it is hard to provide fully interactive content, in particular, for 3-D visualizations. Instead, we use the capabilities of VisMashup to allow for interactive widgets to be placed next to an output on a browser (see Figure 3). As the user modifies the exposed inputs, the system computes the resulting visualization, and makes it available.

Publishing on the Web. To publish workflows onto other Web sites, the CrowdLabs site provides embeddable HTML of either a static image linking back to the workflow page or an embeddable vismashup object. CrowdLabs also integrates with Wikis by extending the wiki markup language. Users can embed either static visualizations or VisMashups onto Wiki pages by including `<vistrail />` or `<vismashup />` tags (provided through the *Embed this Workflow in a Wiki* link present on the workflow pages of CrowdLabs Web site). When using these tags on a CrowdLabs-enabled Wiki, they are replaced with images generated from XML-RPC execution requests submitted to a VisTrails server. Not only does this create an easy way to share workflow results, it provides versioning of scientific results to Wiki technology lacking such possibilities.

Publishing scientific documents. We believe that one of the most interesting applications of a system like CrowdLabs is the impact that it can have on printed media. Often we see published scientific results and wonder how they were generated or would like to compute new results using different data. Currently this is nearly impossible. Every time an image is cut and pasted from a workflow or visualization tool to a paper, most of the lineage information is lost. In CrowdLabs, we advocate a direct linkage from images and workflows results presented in documents to their provenance. To make this possible, CrowdLabs uses a technique analogous to the one used to extend the wiki. We defined a `\vistrail` command that takes in the information necessary to identify and execute the workflow and include the images produced in-place. A LaTeX style file parses the information inside the command, sends it to a python script that builds and makes a HTTP request to CrowdLabs. The images are then downloaded and included as a hyperlinked regular `\includegraphics` command.

Sharing content from other tools. The techniques presented here are easily extensible to any other system that supports provenance and is capable of producing result images or files. For instance, ParaView running with the VisTrails Provenance Explorer plugin [3] could be easily extended to to share visualizations on CrowdLabs.

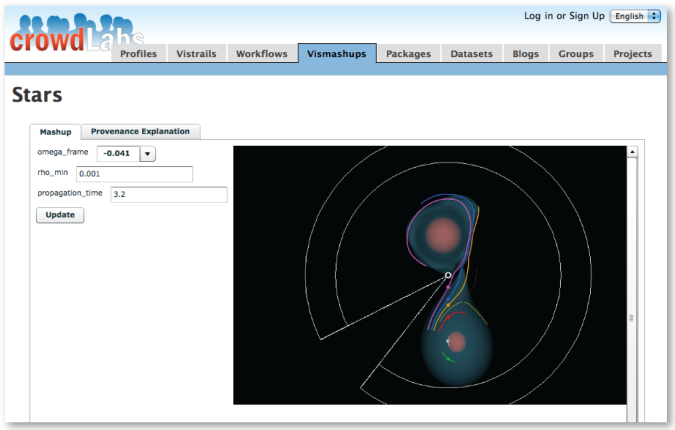

Fig. 3: Interacting with the visualization of a binary star system simulation using VisMashup. Users change parameters on the left and see the resulting visualization on the right. Available at http://www.crowdlabs.org/vistrails/medleys/details/5/.

6 Conclusions and Future Work

The CrowdLabs system builds on infrastructure that our group has been working on since early 2005, and it provides the "last mile" to the scientists. For its most basic use, it does not require any installation of tools in the user's machine, and we see this as an enormous advantage. The barrier of entry is quite small, and it is possible for users to perform a wide range of data analysis and visualization tasks without ever having to install any tools. Besides being deployed on www.crowdlabs.org and on the VisTrails Wiki (www.vistrails.org), the system is also being used at the CMOP [4] site at the Oregon Health & Science University and the ALPS [1] site at ETH in Zurich. We believe many small research groups that do not have all the resources and infrastructure needed for data analysis and visualization tasks can benefit from CrowdLabs. In order to accommodate a growing community, we expect the need for the following new functionality:

Provenance querying and analytics. By mining the data in the CrowdLabs provenance repository, we will be able discover of patterns that can potentially simplify the notoriously hard and time-consuming process of designing and refining scientific workflows [10]. Also useful are advanced querying capabilities that allow users to better explore the workflow, provenance and data.

Improved Web-enabled interfaces and graphics. Although it is possible to use CrowdLabs completely from a Web browser, some advanced functionality, such as interaction with 3D visualization, is not currently supported. One of the big challenges is that Web 3-D graphics are not standardized at this moment, creating a major obstacle in supporting high-end visualization over the Web. Due to some data being remote, we believe that we will need to also add streaming and multi-resolution techniques to our data analysis and visualization workflows.

System improvements. There are a number of system improvements that are needed, including improved scalability in terms of the size of provenance information and data; and a more sophisticated security model.

Acknowledgments. Our research has been funded by the National Science Foundation (grants IIS-0905385, IIS-0746500, ATM-0835821, IIS-0844546, CNS-0751152, IIS-0713637, OCE-0424602, IIS-0534628, CNS-0514485, IIS-0513692, CNS-0524096, CCF-0401498, OISE-0405402, CCF-0528201, CNS-0551724), the DoE SciDAC (VACET and SDM centers), and IBM Faculty Awards (2005, 2006, 2007, and 2008).

References

1. The ALPS project, http://alps.comp-phys.org
2. Bavoil, L., Callahan, S., Crossno, P., Freire, J., Scheidegger, C., Silva, C., Vo, H.: VisTrails: Enabling Interactive Multiple-View Visualizations. In: IEEE Visualization 2005. pp. 135–142 (2005)
3. Callahan, S.P., Freire, J., Scheidegger, C.E., Silva, C.T., Vo, H.T.: Towards provenance-enabling paraview. In: IPAW. pp. 120–127 (2008)
4. NSF Center for Coastal Margin Observation and Prediction (CMOP), http://www.stccmop.org
5. Davidson, S.B., Freire, J.: Provenance and scientific workflows: challenges and opportunities. In: SIGMOD. pp. 1345–1350 (2008)
6. Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures. Ph.D. thesis, University of California, Irvine (2000)
7. Fomel, S., Claerbout, J.: Guest editors' introduction: Reproducible research. Computing in Science Engineering 11(1), 5 –7 (jan-feb 2009)
8. Freire, J., Koop, D., Santos, E., Silva, C.T.: Provenance for computational tasks: A survey. Computing in Science & Engineering 10(3), 11–21 (May-June 2008)
9. Freire, J., Silva, C., Callahan, S., Santos, E., Scheidegger, C., Vo, H.: Managing rapidly-evolving scientific workflows. In: IPAW. pp. 10–18. LNCS 4145 (2006)
10. Koop, D., Scheidegger, C.E., Callahan, S.P., Freire, J., Silva, C.T.: Viscomplete: Automating suggestions for visualization pipelines. IEEE TVCG 14(6), 1691–1698 (2008)
11. McKeon, M.: Harnessing the Web Information Ecosystem with Wiki-based Visualization Dashboards. IEEE TVCG 15(6), 1081–1088 (2009)
12. McLennan, M., Kennell, R.: HUBzero: A Platform for Dissemination and Collaboration in Computational Science and Engineering. Computing in Science & Engineering 12(2), 48 – 53 (2010)
13. Roure, D.D., Goble, C., Stevens, R.: The design and realisation of the virtual research environment for social sharing of workflows. Future Generation Computer Systems 25(5), 561 – 567 (2009)
14. Santos, E., Lins, L., Ahrens, J., Freire, J., Silva, C.: VisMashup: Streamlining the Creation of Custom Visualization Applications. IEEE TVCG 15(6), 1539–1546 (2009)
15. Scheidegger, C., Koop, D., Vo, H., Freire, J., Silva, C.: Querying and creating visualizations by analogy. IEEE TVCG 13(6), 1560–1567 (2007)
16. Strachan, A., Klimeck, G., Lundstrom, M.: Cyber-Enabled Simulations in Nanoscale Science and Engineering. Computing in Science & Engineering 12(2) (March/April 2010)
17. Swivel, http://www.swivel.com
18. Viegas, F.B., Wattenberg, M., van Ham, F., Kriss, J., McKeon, M.: ManyEyes: A site for visualization at internet scale. IEEE TVCG 13(6), 1121–1128 (2007)