Service-Oriented Science

Ian Foster*

Abstract:

New information architectures enable new approaches to publishing and accessing valuable data and programs. So-called service-oriented architectures define standard interfaces and protocols that allow developers to encapsulate information tools as services that clients can access without knowledge of, or control over, their internal workings. Thus, tools formerly only accessible to the specialist can be made available to all, previously manual data processing and analysis tasks can be automated by having services access services. Such service-oriented approaches to science are already being applied successfully, in some cases at substantial scales, but significant further effort is required before these approaches are applied routinely across many disciplines. Grid technologies can accelerate the development and adoption of service-oriented science, by enabling a separation of concerns between discipline-specific content and domain-independent software and hardware infrastructure.

Paul Erdös claimed that a mathematician is a machine for turning coffee into theorems. The scientist is arguably a machine for turning data into insight. However, advances in information technology are changing the way in which this role is fulfilled, by automating time-consuming activities and thus freeing the scientist for other tasks. In this article, I discuss how service-oriented computing—technology that allows powerful information tools to be made available over the network, always on tap and easy for scientists to use—may contribute to that evolution.

The practice of science has of course already been affected dramatically by information technology and in particular by the Internet. For example, the hundreds of gigabytes of genome sequence available online means that for a growing number of biologists, "data" is something they find on the Web, not in the lab. Similarly, emerging "digital observatories" [already several hundred terabytes in dozens of archives (1)] allow astronomers to pose and answer in seconds questions that might previously have required years of observation. In fields such as cosmology and climate, supercomputer simulations have emerged as essential tools, themselves producing large datasets that, when published online, are of interest to many (2). An exploding number of sensors (3), the rapidly expanding computing and storage capabilities of federated Grids (4), and advances in optical networks (5) are accelerating these trends by making increasingly powerful capabilities available online.

Sometimes, however, the thrill of the Web seems to blind us to the true implications of these developments. Human access to online resources is certainly highly useful, putting a global library at our fingertips. But ultimately, it is automated access by software programs that will be truly revolutionary, simply because of the higher speeds at which programs can operate. In the time that a human user takes to locate one useful piece of information within a Web site, a program may access and integrate data from many sources and identify relationships that a human

* Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, U.S.A., and Department of Computer Science, University of Chicago, Chicago, IL 60637, U.S.A. E-mail: foster@mcs.anl.gov. Also Founder and Chief Open Source Strategist at Univa Corporation.

might never discover unaided. Two dramatic examples are systems that automatically integrate information from genome and protein sequence databases to infer metabolic pathways (6) and systems that search digital sky surveys to locate brown dwarfs (7).

The key to such success is uniformity of interface, so that programs can discover and access services without the need to write custom code for each specific data source, program, or sensor. Electric power transmission standards and infrastructure enabled the electric power grid and spurred the development of a plethora of electric tools. In a similar manner, service technologies enable the development of a wide range of programs that integrate across multiple existing services for such purposes as metabolic pathway reconstruction, categorization of astronomical objects, and analysis of environmental data. If such programs are themselves made accessible as services, the result can be the creation of distributed networks of services, each constructed by a different individual or group, and each providing some original content and/or value-added product (8).

We see this evolution occurring in the commercial Internet. As the Web has expanded in scale, so the preferred means of finding things has evolved from Yahoo's manually assembled lists to Google's automatically computed indices. Now Google is making its indices accessible, spurring development of yet other services. What makes Google's indices feasible is the existence of large quantities of data in a uniform format (HTML) and—two important factors that must be considered when we turn to science—smart computer scientists to develop the algorithms and software required to manage the 100,000 computers used (at last count) to analyze Web link structure, and smart businesspeople to raise the money that pays for those computers!

The term *service-oriented architecture* refers to systems structured as networks of loosely coupled, communicating services (9). Thus, *service-oriented science* refers to scientific research enabled by distributed networks of interoperating services. [The term eScience, coined by John Taylor, has a similar but broader connotation (10).]

Creating and Sharing Services

Creating a service involves describing, in some conventional manner, the operations that the service supports; defining the protocol used to invoke those operations over the Internet; and operating a server to process incoming requests. A set of technologies called *Web services* (9) are gaining wide acceptance for these purposes. A variety of commercial and open source Web services tools exist for developing services, deploying and operating services, and developing client applications. A fair degree of experience has been gained with the creation of services and applications in different science domains. While problems remain (e.g., efficiency, interoperability of different vendor offerings), the technology is well beyond the experimental stage. Nevertheless, it can still be a significant step to realize the full potential of service-oriented science, for reasons that I now discuss.

Interoperability. Services have little value if others cannot discover, access, and make sense of them. Yet, as Stein has observed (11), today's scientific communities too often resemble medieval Italy's collection of warring city states, each with its individual legal system and dialect. Web services mechanisms for describing, discovering, accessing, and securing services provide a common alphabet, but a true lingua franca requires agreement on protocols, data formats, and ultimately semantics (12). For example, the definition of VOTable, a standard XML-based representation for tabular data (13), has been a powerful force for progress in astronomy.

Scale. Services must often deal with data volumes, computational demands, and numbers of users beyond the capacity of a typical PC. Responding to a user request—or to the arrival of new data—can involve large amounts of computation. For example, the Argonne GNARE system

searches periodically through DNA and protein databases for new and updated genomes and then computes and publishes derived values (14). Analysis of a single bacterial genome of 4,000 sequences by three bioinformatics tools (BLAST, PFAM, and BLOCKS) requires 12,000 steps, each taking on the order of 30 seconds of run time. GNARE is able to perform these tasks in a timely fashion only because it has access to distributed resources provided by two U.S. national-scale infrastructures, TeraGrid and Open Science Grid (see below).

The impact of automation on service load must also be considered. It is improbable that even a tiny fraction of the perhaps 500,000 biologists worldwide will decide to access Genbank, GNARE, or any other service at the same time. However, it is quite conceivable that 50,000 "agents" operating on their behalf would do so—and that each such agent would want to generate thousands of requests.

Management. In a networked world, any useful service will become overloaded. Thus, we need to control who uses services and for what purposes. Particularly valuable services may become community resources requiring coordinated management. Grid architectures and software—a set of Web services technologies focused on distributed system management—can play an important role in this regard (15).

Quality control. As the number and variety of services grow and interdependencies among services increase, it becomes important to automate previously manual quality control processes—so that, for example, users can determine the provenance of a particular derived data product (8, 16). The ability to associate metadata with data and services can be important, as can the ability to determine the identity of entities that assert metadata, so that consumers can make their own decisions concerning quality.

Incentives. A scientist may work long hours in the lab to obtain results that may bring tenure, fame, or fortune. The same time spent developing a service may not be so rewarded. We need to change incentives and enable specialization so that being a service developer is as honorable as being an experimentalist or theorist. Intellectual property issues must also be addressed so that people feel comfortable making data available freely. It is perhaps not surprising that astronomy has led the way in putting data online, given that its data has no known commercial value.

Scientists are certainly not alone in grappling with these challenges. However, science is perhaps unique in the scope and scale of its problems, the number and diversity of potential contributors, and the subtlety of the questions that service networks can be used to answer.

Rethinking Infrastructure

As scale increases, creating, operating, and even accessing services become increasingly challenging. How do we ensure that service-oriented science realizes its promise of being a democratizing force, rather than increasing the gap between the "haves" and "have-nots"?

Part of the solution is a familiar idea in commercial information technology, namely, *outsourcing*. Building and deploying a service require expertise and resources in three distinct areas: (i) the domain-specific *content*—data, software, and processes—that is to be shared, (ii) the domain-independent software *functions* need to operate and manage the service and to enable community access, such as membership services, registries, metadata catalogs, and workflow orchestration services, and (iii) the physical *resources*—networks, storage, and computers—needed to host content and functions.

The last two capabilities—functions and resources—can, in principle, be handed off to specialist providers. If such specialists can deliver resources or operate required functions for many communities, then (again, in principle) economies of scale can be achieved, while scientists can

focus on what they are good at, providing content and advancing science. In addition, individual services can scale more easily and efficiently when needed.

To see how this strategy can work, consider the SourceForge system, which provides hosting capabilities for communities developing open source software. A new open source project is provided with access to code archiving, mailing lists, and other related functions, as well as the hardware required to host those functions. This outsourcing of function and resource is made possible by the existence of the Internet infrastructure along with standard Web servers, browsers, and associated protocols, which together allow users (in this case, open source communities) to focus on providing content (code) while SourceForge runs Web servers and related infrastructure.

In a similar manner, a "SourceForge for science" would both host scientific communities—operating community membership services, catalogs, storage services, workflow orchestration services, and so forth—and provide access to the hardware resources required to operate both those functions and the application-specific services that constitute the communities "content." In this case, the supporting infrastructure must provide a significantly richer set of capabilities than does SourceForge, encompassing, for example, access control, accounting, provisioning, and related management issues. As noted above, Grid architectures and software (15) address many of these concerns, allowing users to focus on providing "content," which in this case comprises not just Web pages but also services, data, and programs.

SourceForge's hardware requirements are not substantial and thus can easily be provided by a centralized system. However, "cyberinfrastructure" (17) to support scientific communities need not be centralized. For example, the Open Science Grid (OSG) collaboration has constructed a distributed "Grid" linking clusters at 30 sites across the United States that total thousands of computers and tens of terabytes of storage (18). The Enabling Grids for eScience in Europe project, EGEE, has a similar structure. Major research universities and national laboratories participate in OSG and EGEE, but so do smaller institutions, which can thus enhance educational and research opportunities. For example, Florida International University is a significant OSG resource provider, thanks to its 92-processor Linux cluster. All participants can obtain access to large quantities of distributed storage and computational power when they need it. These systems are being used by researchers in high energy physics, biology, chemistry, radiology, and computer science.

This separation of concerns also suggests new roles for campus information technology organizations. In addition to operating commodity services such as Internet and e-mail, these organizations can host functions and provide resources.

Approaches to Scaling

The many groups working to apply service-oriented techniques to science are each exploring one or more of three different approaches to the problem of scaling. In a first *cookie cutter* approach, researchers create dedicated domain-specific infrastructures, in which uniformity is enforced across the board, at the content, function, and resource level. Here, the community standardizes the domain-specific software—and often also the hardware—that participants must deploy in order to provide required functions and resources. I give three examples of such systems.

The Biomedical Informatics Research Network, BIRN (19), is a National Institutes of Health initiative to facilitate collaboration in the biomedical sciences. BIRN has deployed standard compute and storage clusters at 19 sites across the United States. These systems, plus various functions such as catalogs and ontologies, support a variety of collaborative research programs in such areas as mouse brain morphology (20).

The National Science Foundation's Network for Earthquake Engineering Simulation, NEES, is a national collaboratory enabling community access to specialized instrument, data, and simulation resources for earthquake engineering. Each of its 17 instrument sites runs a NEES Point of Presence (a modest PC with a standard hardware configuration) with standard software enabling teleobservation, teleoperation, data collection, and related functions. Central services include catalogs and data archives. NEES has already enabled unique distributed experiments involving facilities at multiple sites (21).

The PlanetLab computer science testbed is a collection of several hundred PCs at universities and research laboratories worldwide, each with a standard configuration and each running standard software (22). Computer scientists can obtain access to "slices" on distributed collections of these PCs, on which they can deploy and evaluate experimental distributed services.

Pushing the electric power grid analogy perhaps farther than we should, cookie-cutter approaches give each participant their own electricity generator. This strategy has the advantage of achieving a high degree of central control and thus uniformity. On the other hand, the cost of expanding capability is high, requiring the acquisition and deployment of new hardware.

In a second, more bottom-up approach, researchers develop *service ecologies* in which agreements on interfaces allow participants to provide content and function in any way they see fit.

I referred above to the international virtual observatory community's VOTable format and to work in bioinformatics. The Department of Energy's Earth System Grid, ESG (2), is another example of a discipline-specific service that emphasizes the definition and implementation of standard interfaces. Building on the widely used OPeNDAP protocol for publishing and accessing environmental data, ESG has deployed services that provide access to over 100 TB of climate simulation data from the National Center for Atmospheric Research's Community Climate Simulation Model and other models involved in the International Panel on Climate Change assessment. Many terabytes of data are downloaded from these services each month.

As a second example, the U.K. ^{my}Grid project (8) has developed tools that allow biologists to define workflows that integrate information from multiple sources, including both biological databases and bioinformatics applications. These workflows can be archived and then run periodically to identify new phenomena of interest, as for example in a recent study of Williams-Beuren syndrome (23).

For a third example, the Department of Energy's Fusion Collaboratory (24) operates services that enable online access to simulation codes. By reducing barriers to use, these services are increasing use of advanced computational techniques. Project members have also demonstrated on-demand coupling of simulation capabilities with physical experiments.

Continuing the electric power grid analogy, such service ecologies define relevant standards, but leave each site to acquire and configure their own equipment. This approach has the advantage that the cost of entry can be low, particularly if appropriate software is available. On the other hand, individual service providers have no immediate means of scaling capability beyond acquiring more hardware.

The third approach involves the definition and deployment of *general-purpose infrastructures* that deliver discipline-independent resources or functions. I have already mentioned OSG and EGEE. As a third example, the National Science Foundation's TeraGrid links resources at nine sites across the United States, with each site deploying a common software distribution that permits secure remote access to computers and storage systems, monitoring of system components, accounting for usage, and so on. TeraGrid targets not only high-end "power users"

but also the larger community via the deployment of "science gateways," discipline-specific services hosted on TeraGrid in support of specific communities.

General purpose infrastructures can be compared with power plants, which operate to provide electricity to any consumer connected to the electric power grid. Like power plants, they have the potential to achieve economies of scale, but also must grapple with the challenges of supporting many users with diverse requirements.

In addition to these national or transnational efforts, many university campuses are deploying "campus Grids" to support faculty and students. For example, Purdue University's NanoHub provides students and faculty with access to various applications, while the UCLA Grid federates multiple clusters across campus and provides online access to popular simulation codes.

These projects, and many others like them, are important experiments in the policies, organizational structures, and mechanisms required to realize service-oriented science. Elements of all three approaches will be required if we are to achieve broad adoption. In particular, it cannot be efficient for every scientist and community to become a service provider. Instead, individual communities—especially smaller communities—should be able to outsource selected functions and physical resources, thus allowing them to focus on developing their domain-specific content. The successful creation and operation of the service providers that support this outsourcing requires both Grid infrastructure software and organizational and funding structures that expose real costs so that "build vs. buy" decisions can be made in an informed manner.

Summary

Service-oriented science has the potential to increase individual and collective scientific productivity by making powerful information tools available to all, and thus enabling the widespread automation of data analysis and computation. Ultimately, we can imagine a future in which a community's shared understanding is no longer documented exclusively in the scientific literature but is documented also in the various databases and programs that represent—and automatically maintain and evolve—a collective knowledge base.

Service-oriented science is also a new way of doing business, with implications for all aspects of the scientific enterprise. Students and researchers must acquire new skills to build and use services. New cyberinfrastructure is required to host services, especially as demand increases. Policies governing access to services must evolve. Above all, much hard work must be done in both disciplinary science and information technology in order to develop the understanding needed for this potential to be fully exploited.

References

1. A. Szalay, J. Gray, *Science* **293**, 2037 (2001).

2. D. Bernholdt *et al*., *Proc. IEEE* **93**, 485 (2005).

3. D. Culler, H. Mulder, *Sci. Am.* **290**, 52 (June 2004).

4. I. Foster, *Sci. Am*. **288**, 78 (April 2003).

5. T. DeFanti, C. de Laat, J. Mambretti, K. Neggers, B. St Arnaud, *Comm. ACM* **46**, 34 (November 2003).

6. R. Overbeek *et al*., *Nucleic Acids Res.* **28**, 123 (2000).

7. Z. Tsvetanov *et al.*, *Astrophys. J.* **531**, L61 (2000).

8. C. Goble, S. Pettifer, R. Stevens, in *The Grid: Blueprint for a New Computing Infrastructure* (Morgan Kaufmann, San Francisco, ed. 2, 2004), pp. 121-134.

9. D. Booth *et al*., *Web Services Architecture* (W3C, Working Draft, 2003; www.w3.org/TR/2003/WD-ws-arch-20030808).

10. T. Hey, A. Trefethen, *Science* **XX**, XX (2005). [this issue]

11. L. Stein, *Nature* **317**, 119 (2002).

12. T. Berners-Lee, J. Hendler, O. Lassila, *Sci. Am.* **284**, 34 (June 2001).

13. F. Ochsenbein et al., *VOTable Format Definition Version 1.1* (International Virtual Observatory Alliance, 2004; www.ivoa.net/Documents/latest/VOT.html).

14. D. Sulakhe *et al*., *J. Clinical Monitoring and Computing*, in press.

15. I. Foster, C. Kesselman, J. Nick, S. Tuecke, *IEEE Computer* **35**, 37 (July 2002).

16. J. Myers, A. Chappell, M. Elder, A. Geist, J. Schwidder, *IEEE Computing in Science and Engineering* **5**, 44 (May/June, 2003).

17. *Revolutionizing Science and Engineering through Cyberinfrastructure: Report of the National Science Foundation Blue Ribbon Advisory Panel on Cyberinfrastructure* (National Science Foundation, Washington, DC, 2003; www.communitytechnology.org/nsf_ci_report).

18. I. Foster *et al*., in *IEEE Intl. Symp. on High Performance Distributed Computing* (IEEE, 2004), pp. 236-245.

19. M. Ellisman, S. Peltier, in *The Grid: Blueprint for a New Computing Infrastructure* (Morgan Kaufmann, San Francisco, ed. 2, 2004), pp. 109-120.

20. M. Martone, A. Gupta, M. Ellisman, *Nature Neuroscience* **7**, 467 (2004).

21. L. Pearlman *et al*., in *13th IEEE Intl. Symp. on High Performance Distributed Computing* (IEEE, 2004), pp 14-23.

22. A. Bavier *et al*., in *1st Symp. on Networked Systems Design and Implementation* (Usenix, 2004), pp. 253-266.

23. R. Stevens *et al.*, *Bioinformatics* **20 Suppl. 1**, i303 (2004).

24. K. Keahey *et al*., *Future Generation Computing Systems* **18**, 1005 (2002).

25. I gratefully acknowledge helpful discussions with Charlie Catlett, Carl Kesselman, Miron Livny, Alex Szalay, and Rick Stevens. This work was supported in part by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract W-31-109-Eng-38, and by the National Science Foundation.

Fig. 1. What it can take to build a service—A powerful approach to the interpretation of newly sequenced genomes is comparative analysis against all annotated sequences in publicly available resources. Currently the largest sequence database at NCBI contains 2.3 million protein sequences. The precision of genetic sequence analysis and assignment of function to genes can be increased dramatically by the use of multiple bioinformatics algorithms for data analysis. The GNARE system discussed in the text precomputes analysis results for every sequence, finding protein similarities (BLAST), protein family

domains (BLOCKS), and structural characteristics. Grid resources are used to run the resulting millions of processes, a task that must be repeated frequently due to the exponentially growing amount of data. Image credit: Bioinformatics group, Mathematics and Computer Science Division, Argonne National Laboratory.

Fig. 2. The Open Science Grid links storage and computing resources at more than 30 sites across the United States to support a variety of services and applications, many concerned with large-scale data analysis. Circles show a subset of Open Science Grid sites; lines indicate communications, some with international partners. Image credit: Iosif Legrand, Caltech.