Software-as-a-Service as a path to software sustainability

Ian Foster
University of Chicago
Argonne National Laboratory
foster@anl.gov

Vas Vasiliadis
University of Chicago
Argonne National Laboratory
vas@ci.uchicago.edu

Steven Tuecke
University of Chicago
Argonne National Laboratory
tuecke@ci.uchicago.edu

ABSTRACT
We argue that the software-as-a-service (SaaS) paradigm has advantages as a sustainable delivery method for scientific software. We report on our experience developing and delivering the Globus Online research data management system using SaaS approaches. We note problems encountered and, for some problems, solutions that appear effective. We also identify conditions that we consider necessary for attaining sustainability and anticipated challenges in meeting these conditions.

Keywords
Sustainability, research data management, software-as-a-service, Globus, software delivery, open source.

1. INTRODUCTION
Scientific software suffers today from three distinct but closely related problems: under-investment, lack of effective economic models, and poor usability. This unfortunate coincidence of factors represents a significant danger to the research enterprise. We believe that it will need new approaches to resolve.

The problem of **under-investment** is due to the important role of software in science [3], the high expense of software development and maintenance, a lack of corresponding growth in the federal research budgets that represent the primary source of support for scientific software, a lack of a good model to choose which software to support, which results in scarce funds being spread over too many projects. These factors are unlikely to change significantly without changes to the current support model.

Neither grant-based funding nor volunteer effort provide an effective **economic model** for software sustainability. In fact, software providers face what economists would term a negative return to scale: with grant funding for software projects typically fixed, the software developer finds that more users bring higher costs (due to increased demand for support) but not increased resources to support operations and continued development.

A frequent consequence of under-investment and inadequate economic models is **poor usability**. With limited resources, scientific software developers must frequently focus on core functionality and de-emphasize ease of use and "user experience"—factors that reduce scientific productivity and encourage unnecessary reimplementation of functionality.

One approach to these problems is to find alternative sources of support. This perspective underpins, although in quite different ways, two frequently proposed solutions to scientific software sustainability, namely commercialization and open source. In the first case, the idea is that software developed for scientists can become sustainable by commercializing it and using the resulting revenues to support further development; in the second, the notion is that many people can be induced to contribute to development of the software at "no cost." In both cases, there is a belief that science can tap some source of support external to funding agencies, namely commercial users or open source contributors. There are certainly impressive examples of success in both cases (e.g., SPSS, and Matlab for commercial software; R, Linux, and Lustre for open source), but many more examples where neither method has worked well. We see no evidence that either approach is a broadly applicable solution to the sustainability problem. The problem with commercialization is that the non-profit research market is small and fragmented, so companies creating products for it cannot attract investment. So the only option is to focus commercialization on other markets, and hope that the research market will benefit from the fallout. That is basically what happened with SPSS and Matlab. However, this approach really only works for general-purpose tools that can be applied to many markets, which is often not true of software for research. The situation for open source software is similar: Linux, for example, has succeeded because of massive investment by companies that see strategic advantage to a high-quality open operating system.

If we cannot identify alternative sources of support, how are we to address the problem of underinvestment? The only alternative is surely to reduce software development and delivery costs. Interestingly, the commercial world has over the past 15 years employed a new approach to software delivery that has been highly successful in achieving such cost reductions, namely software as a service (SaaS). A SaaS provider runs a single version of its software, which many users can access over the network using simple and intuitive Web 2.0 interfaces [2]. Economies of scale mean that the incremental cost to the software provider of adding a user tends to be small. Low barriers to entry for software users, who need not install or operate any software to use SaaS, encourage broad adoption. These factors combine to permit interesting new economic models based on low-cost and usage-based subscriptions that minimize economic barriers to use while providing the positive returns to scale that software providers need to function. The result in industry has been an explosion in the number, variety, and capabilities of the business and consumer software accessible via SaaS. Small businesses, in particular, have been able to slash operational costs while simultaneously improving their performance through access to higher quality software, by outsourcing functions such as payroll, email, accounting, and customer relationship management.

We believe that the benefits of SaaS for business and consumer software, in the form of reduced costs, new economic models, and enhanced usability, can also apply to scientific software. We see widespread adoption of commercial and consumer SaaS in research (e.g., Dropbox, Gmail, Evernote, GitHub). But our interest here is in the use of SaaS concepts and methods by developers and distributors of specialized scientific software, with the triple goal of achieving reduced costs, positive returns to scale

via subscription models, and improved usability. In this way, we posit that it may be possible to achieve a new, broadly applicable, and ultimately far more successful mechanism for sustainable software delivery than those in use today.

Believing in this approach, we started the Globus Online project in 2010 to deliver research data management capabilities via SaaS methods. We have developed and deployed a successful SaaS capability; operated this service for a growing number of users; and, most recently, taken first steps towards sustainability by introducing a (not-for-profit) subscription model for premium services. These experiences form the basis for this paper.

Internet-accessible software is not new to science. In addition to a rich literature on relevant concepts and experiments, we can point to pioneering systems such as the Network Enabled Optimization Server [8], nanoHUB nanotechnology software service [4], and RAST bioinformatics service [6], all of which supports tens of thousands of users. But we are not aware of any such system that has adopted the SaaS business model as a basis for sustainability.

In the rest of this paper, we provide some background on SaaS and its economics; introduce the Globus Online system; review problems encountered implementing this system and solutions that we have explored; and suggest conditions that we argue must be satisfied for a scientific software SaaS system to be successful.

2. AN ECONOMIC MODEL FOR SAAS

While there are interesting technical issues to SaaS, some of which we discuss below, it is above all a business model innovation. Thus we turn next to the economics of SaaS. It may appear odd to discuss economics (and to talk about "customers") in an article on scientific software, but in our view, the sustainability of scientific software is above all an economic issue.

SaaS is commonly supported by a subscription model wherein producers charge consumers a small (relative to traditional licensing models) and recurring (typically monthly or annual) subscription. This fee may be fixed for a certain level of resource or capabilities, vary with usage, or be some hybrid thereof. Many SaaS providers offer basic capabilities or limited resources free to all users, and charge for more advanced capabilities or greater resource usage: the so-called freemium model.

We use the following equations, which describe a high-level economic model for a SaaS service, to frame the discussion.

For a given time period t:

$$RR_{t+1} = RR_t(1 - c) + GR_t$$

$$S_t = RR_t - A_t - F_t$$

where:

RR is the recurring subscription revenue in a period

c is the percentage (on average) of customers who do not renew their subscription (also known as "churn")

GR is the new customer (growth) revenue in a period

S is the net surplus, available for reinvestment (in a non-profit setting) or profit (in a business)

A is the customer acquisition cost

F is the total of all other costs during the period

This model emphasizes that in order for a SaaS provider to be sustainable in a future time period, i.e., in order for S_t to be non-negative, sufficient revenue must be added in the current period to (a) replace revenue lost through churn, (b) offset any increase in fixed costs, and (c) offset new customer acquisition costs.

The model assumes that all costs other than new customer acquisition are fixed within each period; in reality, these costs include a truly fixed component plus variable costs such as delivery operations and user support. But irrespective of these details, what makes SaaS work is that the truly fixed costs can be distributed over a large customer base, and variable costs (delivery operations, user support, etc.) grow at a much slower rate with number of customers than in traditional software delivery. But for these benefits to apply, a SaaS service needs a relatively large number of users.

The explicit representation of customer acquisition costs emphasizes a factor that is not always considered by scientific software developers, namely the need to account for the cost of acquiring new customers, such as marketing, sales, and customer on-boarding, in a software development organization. When using a traditional scientific software delivery models, this issue may seem unimportant—after all, having more users often just means more work. But for subscription-supported SaaS, a large number of users is needed to achieve low subscription fees and thus encourages broad adoption. This seemingly Catch-22 situation can be a challenge for SaaS, as we discuss below. But it emphasizes the need to budget for customer acquisition.

The SaaS model also allows providers to realize a number of intangible benefits: (1) greater flexibility in managing costs—even without new subscriptions—because recurring revenue means providers do not need to make drastic cost cuts as demand fluctuates; (2) multiple opportunities to engage with customers as compared with the traditional one-time licensing purchase, e.g., free trial expiration, renewal, upgrade, add-on service; and (3) increased development flexibility as some aspects of the service may be implemented with less concern for the customer experience when the software is not installed and managed by the customer, e.g., porting, installation software, upgrade scripts, and operations console.

3. EXPERIENCE WITH GLOBUS ONLINE

Our belief that SaaS is well suited to scientific software delivery is informed by our work on Globus Online [1], a system that enables researchers to move, sync, and share big data, whether stored on personal, campus, or national computing resources.

Globus Online allows users to request the movement or synchronization of data between remote storage systems that may be located in different administrative domains. Web, REST, and command line interfaces allow for both interactive use and integration with application tools and workflows. The service also allows users to share data with collaborators directly from the system on which the data resides, without the need for additional storage or costly cloud storage solutions.

Globus Online file transfer and sharing services are underpinned by identity federation, user management, and group management capabilities that make it easy to leverage cyberinfrastructure to improve the user experience (e.g., single sign-on using InCommon), and integrate those services into a science gateway, community portal, or other custom application.

3.1 Development and Early Adoption

We made Globus Online publicly available in November 2010, and since that time have acquired over 10,000 registered users spanning multiple research domains. Currently, >700 researchers use the service each month to transfer >1 petabyte of data. Over

21 petabytes and 1B files have been transferred since launch. The service has been well received, with many users providing unsolicited praise for its performance and reliability (www.globusonline.org/quotes/). One user commented that, "The system is reliable and secure – and also amazingly easy to use. ... It just works." [David Skinner, OSP Group Leader, NERSC]. Another: "I routinely have to move hundreds of gigabytes of data – Globus Online makes it easy, so I can execute these transfers with very little effort." [Jeff Porter, STAR Experiment, Lawrence Berkeley National Lab]. We attribute these positive experiences to the fact that Globus Online does not just help researchers with a tedious but important task, it takes that task off their hands entirely, and does so with unprecedented ease of use.

Globus Online development was supported in part by federal grants. Initial operating costs (e.g., Amazon server charges) were low, commensurate with the level of usage, and were funded by the University of Chicago and resource grants from Amazon. As more institutions (vs. individual users) adopted, our costs have grown significantly: more Amazon resources are required to maintain desired service levels, more people are required to provide the necessary levels of technical support, and more developers are required to address the growing list of user-requested features.

3.2 Subscription Plans

To offset our operations costs, and to take a first step towards sustainability, we recently started offering for-fee subscriptions.

A first set of "Provider" plans give campus computing centers and other resource providers additional management tools and higher levels of support. While we are experimenting to find the "right" pricing model, resource providers appear willing to pay a modest amount that is in line with other infrastructure software costs, e.g., license fees for storage management software.

We also offer individual users a "Plus" plan that allows them to use enhanced features such as file sharing and peer-to-peer file transfer. This plan is too new to permit evaluation of adoption.

3.3 Problems and Solutions

We note some problems that we have encountered in building and operating the Globus Online system.

3.3.1 Building and Operating a Scalable Service

As noted above, we require many users to realize economies of scale. Thus, the SaaS software must be able to scale in multiple dimensions: numbers of users, concurrent requests, and so on. Best practices for building and operating scalable SaaS are by now reasonably well understood in the industry, but are not necessarily known in the scientific software community. We had to climb a steep learning curve in the early stages of the project.

We address scaling of the Globus Online service itself by a combination of careful design and use of commercial cloud services (specifically, Amazon Web Services) to host the software. In addition, we replicate service elements across multiple Amazon availability zones to enhance reliability, and deploy monitoring and alert systems to detect problems early. These methods have allowed us to sustain greater than 99.9% availability despite sustained growth in users and requests.

We have found that merely designing for scalability in the total number of users is not sufficient: it is important to have supporting infrastructure that addresses unexpected demands placed on the service by individual users, such that quality is not degraded for the rest of the user base. For example, we were at one point surprised by a request to transfer 50 million files—a request that would have overwhelmed our request database if we had not detected and responded to it in a timely manner.

3.3.2 Institutional Obstacles to Subscription Services

Introducing and operating a for-fee service within the context of a research institution can lead to interesting discussions with legal departments. We are told that some public universities are prohibited by statute from offering for-fee services of any sort. The University of Chicago, as a private university, has no such prohibition, but had to be comfortable that the Globus Online service is consistent with its non-profit mission.

As we worked to establish billing and payments capabilities within Globus Online, we found that existing vendor arrangements for common services within a university can be a significant hurdle to operating a for-fee service. We evaluated multiple vendors and selected a system that would meet our current and future requirements, based on a combination of features, ease of integration, and strict usage based charge model. However, the University of Chicago has an exclusive agreement with a credit card payment processing vendor that does not support the selected system, forcing us to look at alternative providers. We were also unable to conclude the process with the alternate provider due to university contractual requirements, ultimately requiring us to purchase a more expensive solution.

These considerations emphasize that establishing a for-fee service is not for the faint of heart. Institutional change may be required for SaaS models to be adopted more broadly.

3.3.3 Institutional Obstacles to Subscriptions

Selling Provider plan subscriptions has also proven to be time consuming. Wide variability in the contract terms required by various institutions means that negotiating with each institution individually will add substantially to operations costs. One approach to mitigating this issue is to utilize a channel partner (e.g., Internet2 and their NET+ program) that already has standardized contractual agreements with multiple institutions.

3.3.4 Cultural Challenges

We have encountered challenges getting people to pay even small sums for software. Interestingly, this phenomenon seems to be domain-dependent: anecdotal evidence suggests that NIH-supported researchers like to pay for services, because it provides confidence that the service will persist, while NSF-supported researchers do not. We do not understand this difference in views.

3.3.5 Compliance Issues

Requirements for privacy and information protection can severely limit adoption of software-as-a-service solutions. Data privacy is always a requirement at some level, but the regulations governing personal health information (PHI) are particularly challenging. Serving researchers in certain biomedical disciplines requires that we provide a HIPAA (Health Insurance Portability and Accountability Act) compliant solution, along with Business Associate Agreements (BAA), to those dealing with data from human subjects. The complexity of this requirement is compounded by the nature of SaaS, in that the software may be hosted by a third party Infrastructure as a Service provider (e.g., Amazon Web Services). We expect to be able to adapt Globus Online to work with HIPAA data. However, this was not a requirement that had considered when we began this project.

4. CONDITIONS FOR SUSTAINABILITY

Our experience with Globus Online has only increased our enthusiasm for SaaS. We have been in the business of creating,

distributing, and supporting research data management software for many years. The positive response to our new SaaS offering from both researchers and resource providers, and the exceptionally broad and rapid uptake, has been exciting.

But positive reviews do not solve the problem of sustainability. We have demonstrated that we can operate a high quality service and that we can get some people to pay modest subscription fees for access. To achieve sustainability, we must increase the number of paying users substantially. In working out how to do this, we believe that we can profit from lessons learned in the high-tech industry, as we now discuss.

Geoffrey Moore famously argued that the key to high-tech success is to "cross the chasm" [7] that separates an initial user base—"the early adopters"—from the (substantially larger) set of users that make up the mainstream market and provide the scale necessary for sustainability. We believe that his analysis applies to scientific software as well. In our case, our early adopter market comprises mostly high performance computing (HPC) resource owners and administrators. We provide to them a service that alleviates a simple but widely prevalent pain point: fast, reliable file transfer, synchronization, and sharing for "big data." In addressing this market we have focused primarily on the user experience, and we believe that this focus has been instrumental in both attracting and retaining resource administrators and end-users alike. In order to cross the chasm we must next satisfy (at least) the following conditions.

4.1 Develop the "Whole Product"

The term "whole product" is used in marketing to refer to a "core product … augmented by everything that is needed for the customer to have a compelling reason to buy" (Wikipedia) [5]. As before, we find ourselves referring to marketing texts because sustainability demands a large user base, and to achieve many users one must be able to communicate the value of a product to customers—the very definition of marketing.

In the context of Globus Online, the implications of the whole product concept are that in order to appeal to a broader set of users, we must address gaps in product functionality. Examples include: self-service monitoring and troubleshooting capabilities for less sophisticated system administrators; HIPAA compliance to enable use in research labs that deal with personal health information; and integration with external resource providers such as cloud storage providers (e.g. Amazon Web Services S3). Our choice of new features to add is driven by both careful study of current Globus Online usage and broad consultation with current and potential future users.

4.2 Expand to Adjacent Market Segments

While fruitful, the HPC center market is insufficient to achieve sustainability. Moore's advice is to expand into adjacent market segments, leveraging one's footprint in the initial core area. In our case, our value proposition resonates in campus research computing centers and laboratory computing cores that have needs similar to those of HPC centers but operate under tighter cost constraints. Beyond product development, this condition requires us to invest in more targeted marketing, a costly activity which is not funded by federal agencies. We will need to invest in careful outreach and education to ensure that existing users are not disenfranchised (by any perceived intent to profit at their expense) and to iteratively converge on optimal price points and service plans for attracting new users.

4.3 Implement Core Operations

Two core operational capabilities are needed to capture and serve users over the long-term. The first is an infrastructure for tracking, billing, and payments processing. As described above, we have chosen to use an external provider for this capability; however, it still required substantial investment to enhance Globus Online to make use of this service. The second is user support, a critical operations function that may seem unnecessary, given that SaaS offerings are designed to be self-service. In practice, support demands per user may be lower for a SaaS service, but support remains an integral part of the overall user experience.

5. FUNDING TO CROSS THE CHASM

The conditions described above create a significant funding challenge. Marketing and outreach, development of non-core functionality, and implementation of business operations are rarely funded by federal agencies. The resulting funding gap is perhaps the most significant challenge to sustainable software delivery. Further, our experience has shown that the sales cycle with most institutions will be longer than envisaged. As one would expect, customers want to see a clear demonstration of the value of the service before committing. Typically this requirement translates to a free trial over an extended period, during which multiple stakeholders can evaluate the service and explore how it may integrate with existing data management infrastructure. The resulting challenge to a non-profit service provider is the level of resources required to adequately support multiple such trials concurrently, which exacerbates the funding gap.

We are interested to discuss how such chasm crossing can be supported for scientific software. In industry, venture capital is often used, with initial losses offset by the promise of future profits. For non-profit organizations, conventional venture funds are not an option—the research market that we target is not sufficiently large to provide required returns. Perhaps experiments such as Globus Online can suggest new funding models and organizational structures to allow scientific software SaaS providers to become sustainable at lower cost.

ACKNOWLEDGMENTS

We thank the Globus Online team for their work, and our loyal users that support us by using the service. This work was supported in part by grants NSF ACI-1240726 and DOE DE-AC02-06CH11357.

REFERENCES

1. Allen, B., Bresnahan, J., Childers, L., Foster, I., Kandaswamy, G., Kettimuthu, R., Kordas, J., Link, M., Martin, S., Pickett, K. and Tuecke, S. Software as a Service for Data Scientists. *Communications of the ACM*, 55(2):81-88, 2012.
2. Dubey, A. and Wagle, D. Delivering software as a service. *The McKinsey Quarterly*, May, 2007.
3. Keyes, D. and others National Science Foundation Advisory Committee for CyberInfrastructure Task Force on Software for Science and Engineering: Final Report, http://1.usa.gov/1fMYINU, 2011.
4. Klimeck, G., McLennan, M., Brophy, S.P., Adams III, G.B. and Lundstrom, M.S. nanoHUB.org: Advancing Education and Research in Nanotechnology. *Computing in Science and Engineering*, 10(5):17-23, 2008.
5. McKenna, R. *The Regis Touch*. Addison Wesley, 1985.
6. Meyer, F., Paarmann, D., D'Souza, M., Olson, R., Glass, E., Kubal, M., Paczian, T., Rodriguez, A., Stevens, R., Wilke, A., Wilkening, J. and Edwards, R. The metagenomics RAST server - a public resource for the automatic phylogenetic and functional analysis of metagenomes. *BMC Bioinformatics*, 9(1):386, 2008.
7. Moore, G. *Crossing the Chasm: Marketing and Selling Disruptive Products to Mainstream Customers*. Harper Business, 2006.
8. More, J., Czyzyj, J. and Mesnier, M. The NEOS Server. *IEEE Journal on Computational Science and Engineering*, 5:68-75, 1998.