

# Final report and assessment WinGlob project

Maarten van Steen

September 2005

## Contents

<b>1</b>	<b>Background</b>	<b>2</b>
1.1	Globule . . . . .	2
1.2	Deployment . . . . .	2
<b>2</b>	<b>Goals</b>	<b>3</b>
2.1	Windows support . . . . .	3
2.2	Deployment support . . . . .	3
<b>3</b>	<b>Assessment</b>	<b>3</b>
3.1	Technical achievements . . . . .	3
3.2	Dissemination achievements . . . . .	4
<b>4</b>	<b>Future Work</b>	<b>6</b>
<b>5</b>	<b>Lessons learned</b>	<b>7</b>
<b>6</b>	<b>Agreement on this report</b>	<b>7</b>
<b>7</b>	<b>Financial Report</b>	<b>8</b>

# 1 Background

The WinGlob project has been carried out as part of a larger project, called Globule, that aims at developing adaptive replication techniques for large-scale collaborative content delivery in the Internet. In this section, we briefly describe the Globule project and focus on one important aspect, namely ease of deployment.

## 1.1 Globule

Globule is a research project that was initiated several years ago. Its main goal is to investigate various techniques for replicating Web content across the Internet, and to provide the means for dynamically adapting to a replication strategy that optimizes the performance of a Web site.

The notion of performance is to be taken in the broadest sense of the word, but typically covers metrics such as bandwidth, client-perceived latency, availability, and result consistency. The latter expresses to what extent the content presented to a client matches the content as currently stored on a Web site. Initially, Globule concentrated only on static content, such as simple Web pages, but has by now made significant progress towards the replication of dynamically generated content as well.

A main achievement of our research has been identifying the need to differentiate, when, how, and where to replicate content [1]. As it turns out, even for relatively stable sites, regularly evaluating and adapting strategies at a fine grain can lead to significant performance improvements [4].

Globule has been implemented as an extension to the Apache Web server [2], including an extension for DNS redirection [7]. As such, it offers an excellent platform for hands-on experimentation of the various replication strategies that we develop.

## 1.2 Deployment

Globule is in a sense a successor to the Globe system, an object-based distributed system also developed at the Vrije Universiteit. One of the important lessons that we learned from Globe, is that special attention should be paid to deployment. For Globule, we have decided to concentrate mainly on technical deployment: it should be easy for a user to download and install the system, and to participate in the Globule Content Delivery Network (CDN).

This ease of deployment has resulted in a simple packaging of our software, including binary packages wrapped as RPMs for various Linux distributions. In principle, a user need only modify a configuration file in order to get Globule up and running. We are now in a stage that these configuration files can be automatically generated and installed on a user's computer.

## **2 Goals**

As mentioned, one of the goals of Globule is be easily deployable. However, deployment has concentrated mainly on technical issues such as ease of installation. Our goal is also to reach a large user community of which the members will actively use the system. Such a community would be able to provide valuable feedback on the technical issues, but would also give us better insight in what is needed and what not.

Globule was initially suited only for UNIX platforms, notably Linux. We felt that support for Windows was a prerequisite in order to reach a large number of users. The main goal of the WinGlob project was therefore to port Globule to Windows, while continuing to provide easy technical deployment.

### **2.1 Windows support**

Globule has been ported to Windows, and should work on Windows 2000 and upwards. The installation is done by means of a standard Windows installer, resulting in the installation of an Apache server along with the Globule extensions. The Windows support is provided for Globule 1.3.0, which is identical to the support for Linux.

### **2.2 Deployment support**

Besides the Windows support we have set up a broker system that will allow users to register machines. Registration allows users to have their content replicated to other machines, and vice versa, but also to generate configuration files that could otherwise create a hassle for users unfamiliar with technical details of Web servers.

## **3 Assessment**

The WinGlob project essentially had two strongly related goals: (1) porting Globule to the Windows platform, and as such (2) stimulate the use of Globule for a broader audience.

### **3.1 Technical achievements**

Although a number of technical issues still need to be solved, we can safely say that WinGlob has met its main technical achievements:

- There is a simple procedure for installing Globule on a Windows machine, which is no more complicated than installing other Windows applications. The look-and-feel for the Globule installation is very similar to that of other installs.

- There is a simple means for users to register their machine and to make use of other machines, while at the same time the corresponding configuration files will be automatically generated.
- Globule servers have been enhanced so that clients will not be redirected to unavailable servers. Moreover, we have provided a means to have traffic be redirected to replica servers in spite of the unavailability of the home machine hosting a user's Web site.

In parallel to these developments, we have made considerable progress towards the support for dynamically generated Web pages (notably those supported by a combination of PHP and MySQL) [6, 5]. Replicating PHP code has been incorporated in Globule 1.3.0 (but installing PHP is not yet part of the Windows installer).

These achievements match those formulated in the original project proposal:

start	duration	description	status
01.05.04	1 m	Investigating portability problems Globule 1.0	<b>DONE</b>
01.06.04	3 m	Porting a version of Globule to Windows	<b>DONE</b>
01.09.04	2 m	Developing an easy-to-use install shield that supports updates	<b>DONE</b>
01.11.04	2 m	Development of server pool manager for registered replica servers	<b>DONE</b>
01.01.05	1 m	Enhancing Globule DNS server: support name provisioning and slave assignment	<b>DONE</b>
01.02.05	1 m	Enhancing Globule DNS server: support redirection to only available servers	<b>DONE</b>
01.03.05	2 m	Preparation of software for release as Winglob version	<b>DONE</b>

### 3.2 Dissemination achievements

Regarding the dissemination of Globule, the project has not been successful. We aimed at an installed user base of 300 users by May 1, 2005, but reality is that only a handful of users are deploying Globule. The number of downloads is low (only a few per week at the most) and we know of only a few people that have actively shown interest in the project. Several activities have been undertaken to stimulate Globule usage:

- Articles have been published in popular magazines, including the Dutch Linux magazine and the highly popular Dutch version of c't. These publications generated several reactions, but did not lead to more users.
- A competition was organized to stimulate students to test and improve Globule. The response, even after active promotion by the research team, was surprisingly low.

- Presentations at SANE 2005 and a KDE meeting. We have submitted proposals for ApacheCon Europe (rejected), and the O'Reilly European Open Source Convention 2005 (rejected).

In addition, a separate steering committee had been formed to explicitly address dissemination of results (consisting of Wytze van der Raay, Hans Onvlee, and Maarten van Steen). The committee has met on several occasions to discuss ideas, most of which have been followed up.

Triggered by the unsuccessfulness of disseminating Globule, I have begun to explore a different avenue, starting with analyzing why it is apparently so difficult to bring projects like Globule out to the masses. I know that many other NLnet-sponsored projects suffer from the same problem: technically sound, but poor distribution.

Systems such as Globule can be considered to be innovations. The rate of adopting innovations by individuals depends largely on the following characteristics [3]:

**Relative advantage:** To what extent is the innovation actually better than current solutions to the problem at hand? It is important to note that this relative advantage is subjective: different people will value the innovation differently. One should not think in terms of an objective relative advantage.

**Compatibility:** To what extent is the innovation compatible with current solutions. A high degree of compatibility will stimulate the innovation's use.

**Complexity:** To what extent is the innovation experienced as being complex, or difficult to understand.

**Trialability:** To what extent can the innovation be tested, preferably aside the currently deployed solution in order to allow for immediate comparison.

**Observability:** To what extent are the results of an innovation visible: the more visible the (positive) effects of using an innovation are, the more likely it will be adopted.

When considering Globule, it can be easily seen that these fundamental characteristics have been overlooked. The relative advantage over using a standard hosting solution is very difficult to explain to an end user. For Globule, we argued that people can simply maintain their own site, even on their laptop, without the need for being continuously connected to the Internet. There are other technical advantages, but the real question is whether an end user will perceive these as being so much better. The relative advantage of Globule has not been sufficiently addressed.

Regarding compatibility and trialability, I believe Globule is in good shape due to our adoption of standard solutions such as Apache. In fact, Globule can simply

replace an existing (Apache) server while continuing to serve content without using any Globule-specific feature.

When it comes to complexity, we must admit that Globule is difficult to understand by the “average” end user. It requires a technical computer science background to understand what Globule achieves. Already the concept of replication is difficult for most people, let alone its realization through Globule replica servers. Although we have developed something like a mental model (e.g., as part of our documentation), this model does not suffice for normal users. More, and explicit attention should have been paid to this topic.

Strangely enough, observability is at the same time very good and very poor. The problem we are addressing here is that of distribution transparency: it has been our deliberate choice to *hide* the fact that Globule can handle performance and availability problems. Yet, our solutions can be appreciated only when users have experienced the associated problems as being real.

These observations bring us to another point related to diffusing innovations. In order to discuss transfer of technology, we need to distinguish knowledge, from use, from commercialization. These three indicate different levels of transfer. At the knowledge level, the user should already be capable of understanding what is being transferred. This is the level that we are demanding for Globule. At the level of usage, the user will actually have put the innovation to work so that others can observe its relative advantages. Globule has never reached this level, but without specific efforts, it is questionable whether we actually will. Finally, at the level of commercialization, the innovation is actually being sold.

Note that, implicitly, we have been aiming at achieving all three levels at the same time. We could never have reached beyond the simplest level, namely transferring knowledge about the technology, as this is already problematic: Globule is by itself too difficult to understand by the targeted average end user.

## 4 Future Work

Work on Globule continues. At present the Windows version of Globule is in synch with the UNIX version. In order to keep this situation, we will ensure that at least one Globule replica server is hosted on a Windows machine (most likely a Win2K machine running at my home). Furthermore, support for Windows will continue, most likely in combination with student projects. We still believe that having Windows available is important for dissemination. In addition, we are now seriously considering Mac OS X support as well (we have a version running on this operating system as well).

Starting in January 2006, we plan to have a PhD student work on the (self-)management issues for Globule. In particular, we are interested in providing a completely nonintrusive version of Globule, capable of automatically handling configuration issues, failures, updates, performance aspects, and so on. Again, features for UNIX will be ported to WinGlob as well.

## 5 Lessons learned

In hindsight, I believe that the WinGlob project has been technically successful. Although the dissemination of results is poor, new insights have been reached that may hold for many other NLnet projects. In particular, I come to the conclusion that trying to disseminate Globule as a stand-alone solution is a *bad idea*. No dissemination plan would have succeeded for the simple reason that the relative advantage of Globule is not easy to perceive by the originally targeted audience. In addition, Globule has a relatively high technical complexity, which severely burdens its adoption.

I now firmly believe that adoption of Globule is highly independent of its current technical status. Adding support for dynamically generated Web pages, or massively improving the brokerage system will not substantially contribute to its success. The main problem is that we cannot explain in simple terms (1) what advantages Globule has, and (2) how those advantages are achieved. Of course, being technically sound is a minimal requirement, as bugs will definitely hinder wide-spread dissemination.

Instead, it may very well be the case that Globule technology should have been wrapped in an easy-to-understand and easy-to-use application, such as a Web site development package. The difference with existing packages may then have been the increased quality of service when it comes to availability and maintainability.

I recommend that for future NLnet proposals, proposers should explicitly address the characteristics of innovation (relative advantage, compatibility, complexity, trialability, and observability), such that potential problems regarding each of them are identified and tackled. Addressing these characteristics at least paves the way to a proper dissemination and adoption of results. This could then be accompanied by a dissemination plan.

## 6 Agreement on this report

With respect to this report, NLnet Foundation and the Globe team at the Vrije Universiteit have agreed on the following:

1. The technical deliverables specified in the original WinGlob plan have been formally provided.
2. The efforts that have taken place during the WinGlob project contract period to disseminate the results match what was planned.
3. The assessment of the WinGlob project as given in this report is sufficient.
4. Both parties explicitly express their commitment to further disseminate Globule/WinGlob, and to pro-actively distribute current and new results to the interested community.

## 7 Financial Report

A financial statement has been added separately to this report.

### References

- [1] G. Pierre, M. van Steen, and A. Tanenbaum. “Dynamically Selecting Optimal Distribution Strategies for Web Documents.” *IEEE Transactions on Computers*, 51(6):637–651, June 2002.
- [2] G. Pierre and M. van Steen. “Design and Implementation of a User-Centered Content Delivery Network.” In *Proc. 3rd Workshop on Internet Applications*, June 2003. IEEE Computer Society Press, Los Alamitos, CA.
- [3] E. M. Rogers. *Diffusion of Innovations*. Free Press, New York, NY, 5th edition, 2003.
- [4] S. Sivasubramanian, G. Pierre, and M. van Steen. “A Case for Dynamic Selection of Replication and Caching Strategies.” In *Proc. 8th Web Caching Workshop*, Sept. 2003.
- [5] S. Sivasubramanian, G. Pierre, M. van Steen, and G. Alonso. “GlobeCBC: Content-blind Result Caching for Dynamic Web Applications.” Technical Report, Vrije Universiteit, Department of Computer Science, June 2005.
- [6] S. Sivasubramanian, G. Alonso, G. Pierre, and M. van Steen. “GlobeDB: Autonomous Data Replication for Web Applications.” In *Proc. 14th International World Wide Web Conference*, May 2005.
- [7] M. Szymaniak, G. Pierre, and M. van Steen. “Netairt: A DNS-based Redirection System for Apache.” In *Proc. International Conference WWW/Internet*, Nov. 2003. IADIS.