WinGlob: Globule for Windows

Updated version

Maarten van Steen

June 15, 2004

1 Proposal overview

1.1 Background

Globule is a research project that aims at developing a user-centric Content Delivery Network (CDN). Such a network consists as an overlay in which the nodes are owned by end users rather than ISPs. In Globule, nodes transparently collaborate to provide strong guarantees with respect to performance and availability of Web documents. To this end, we have developed modules that extend the basic functionality of the Apache 2 Web server, and take care of automatically replicating Web documents, and redirecting clients to the replica server that can best service the request.

1.2 Objective

Globule has been developed for UNIX platforms, and has been tested on Solaris and Linux. The objective of the proposed project is to make Globule easily available for a large group of potential users, in particular those that have Windows machines. Users of Globule on Windows will initially be able to set up a Web site on their own machine of which the content is managed by standard tools such as Word and Netscape Composer.

1.3 Motivation

The main reason for wanting to make Globule available for Windows is that Globule is designed to run on end-user machines. We aim at wide-spread deployment, which then makes the Windows community a better target than the focusing only on the UNIX community. There are many benefits in establishing a reasonable-sized user base, including feedback on bugs, performance, usability, functionality, usefulness, and so on. In general, it will lead to an improvement of Globule as a whole.

Another important reason is that by explicitly focusing on general end users (rather than technical, UNIX-oriented users), we believe that requirements for functional enhancements will be properly prioritized. By concentrating on important user requirements, we should be able to more easily establish a user base, which, in turn contributes to the potential success of the project.

1.4 Application

Although the project has its internal goal set to promote the use of Globule and its support for adaptive replication, what would make Globule attractive for (Windows) end users is that they can set up a Web site *on their own personal computer*, which remains available even when that computer is shut down. Availability while being disconnected, combined with pure local development and maintenance will be the selling point for WinGlob.

1.5 Requested funding

We request funding for 1 FTE, being a scientific programmer.

2 Globule

Globule is a research project that can be considered as a successor to Globe. The project aims at developing a large-scale distributed platform for hosting Web applications. This platform is also referred to as Globule. In its initial setup, it is designed to support the automatic and adaptive replication of static Web pages. These are pages that effectively contain only HTML text, images, audio/video, or client-side scripts. We are conducting research to extend Globule's functionality to also support Web pages containing server-side scripts, cgi-bin programs, and, in general, pages that trigger direct calls to server-side components.

Globule uses the Apache 2 software as its middleware platform. All code that is needed to support automatic and adaptive replication is written as Apache modules that can be loaded into standard servers. There is no need to adapt client-side software: any Web browser can provide access to Globule-enabled Web sites. The principal operation of Globule is as follows.

2.1 Automatic replication

Every Web page in Globule has a designated *origin server*, which is responsible for keeping track of client accesses and evaluating what the best *replication strategy* is for that page. In other words, an origin server evaluates when, where, and how an associated Web page should be replicated. Currently, all updates for a specific page also take place at the page's origin server, but this restriction will be relaxed in the future. A Globule server can act as origin server for many pages, and typically covers an entire Web site. Note that every page has its own associated replication strategy.

The core of Globule is formed by its capability for automatically replicating Web pages. The internal design and implementation of a Globule server is described in [3]. Evaluation of replication strategies is based on trace-driven simulations as reported in detail in [2]. There, we show that by replicating a Web site at the granularity of Web pages we can achieve near-optimal performance for client-perceived latency *as well as* consumed bandwidth. In a later study, we further demonstrate that continuous evaluation of replication strategies is necessary to maintain high performance, and discuss how evaluation can be done efficiently [4].

2.2 Client redirection

Automatic replication requires that a client can be redirected to the replica server that is closest. Each Globule server is therefore equipped with a *redirection component* [5]. We support HTTP redirection, but more interestingly, also DNS-based redirection. DNS-based redirection is interesting as it provides replication transparanecy. In addition, it solves several problems related to links as they occur in HTML pages. However, it required us to modify the core of Apache so that it could also process the UDP-based DNS queries. A user is not forced to use our modified version of Apache. The only consequence is that DNS-based redirection is not supported.

An important and novel feature of Globule client redirection, and which is closely tied to automatic replication, is the positioning of nodes in the Internet. This positioning allows us to redirect a client to the *closest* replica of the requested Web page, where the distance between nodes is an estimation of the actual latency. The principles behind this positioning are based on the fact that latencies in the Internet can be modeled in a 6-dimensional geometric space. The details of our positioning system can be found in [6]. Note that this feature has not yet been implemented for Globule; only experimental software running on, for example, PlanetLab and RIPE test boxes, has been developed.

2.3 Current status

Globule version 1.0 has just been publicly released, following a number of internal releases since the beginning of December 2003. It is available as source and binaries from `www.globule.org`. It is currently used to host a number of sites owned by project members. We are now setting up minimal support so that interested parties (such as students from the VU) can participate and evaluate the system. As a side note, we have set up a replicated mirror of the Web site of the Stichting NLnet (see `http://wilfred.cs.vu.nl/nlnet`).

Globule 1.0 supports three simple, static replication strategies. A Web page can be replicated across several servers, one operating as a so-called master. The other servers act as slaves and are to be manually entered into a configuration file. We support strong consistency, meaning that any update at the master will immediately lead to an invalidation at the slaves. We also support weak consistency through TTL-driven replication (a replica is updated after an expiration time), and no replication (a slave simply forwards a request to the master).

Currently, configuration mostly involves compilation of Globule modules and manually changing the Apache configuration file. The documentation appears to be sufficient to allow anyone with basic knowledge of UNIX systems administration to successfully install and run Globule. Binary and source RPMs are available to further ease this process. Installing updates is nearly trivial and takes less than 5 minutes, requiring a simple restart of the Globule server.

There are no special content management tools: Globule synchronizes replicas by detecting file updates. An important advantage of this approach is that users can choose their favorite tools for maintaining a Web site.

It should be noted that Globule can operate in parallel to a standard, nonreplicated Web server, as demonstrated, for example, by the site `www.van-steen.net`.

2.4 Current project members

Globule is a fairly large research effort in our group. At present, the following people are activaly participating in the project:

Maarten van Steen (project leader):

Responsible for supervising the research that is being conducted as part of Globule, and setting and controlling short– and long-term goals.

Guillaume Pierre (temporary assistant professor):

Main architect of Globule and assistant to project leader. Is currently responsible for designing a brokerage system by which replica servers can be identified.

Swaminathan Sivasubramanian (PhD student):

Researcher concentrating on on-demand replication of Web *applications* that are to be hosted by Globule. This work also includes developing solutions to support the replication of dynamic Web content.

Michal Szymaniak (PhD student):

Researcher concentrating on latency-based placement of replicas. To this end, a fully decentralized system has been developed by which Internet latencies can be represented in a 6-dimensional Euclidean space.

Arno Bakker (programmer):

Responsible for Globule development and releases. Assigned part-time to the project.

Berry van Halderen (programmer):

Responsible for Globule development and releases. Assigned full-time to the project on a 3-year external contract.

Wilfred Dittmer (requested programmer):

Programmer requested for carrying out the work as specified in this proposal.

3 Globule for Windows

Globule has been entirely developed for UNIX systems. It has been tested on Solaris and Linux. We propose to develop a Windows-based version that will allow non-technical users to set up their own Web site without requiring Web hosting facilities from their ISP.

3.1 Motivation

Virtually all the research that is conducted in the Computer Systems group at the Vrije Universiteit is targeted towards obtaining scientific results. However, it is clear that the outcomes of our research can be put to practical use for a much broader community than scientists. Globule is an excellent example that illustrates that the gap between high-quality scientific research and practical usage by a non-technical and non-scientific community need not be that big.

Our prime motivation for implementing Globule for Windows, is that we are eager to have a relatively large user base to demonstrate that Globule actually scales, or at the very least to reveal unanticipated scalability problems. Our research on large-scale distributed systems of which Globule forms an important part will be taken more seriously if we can substantiate scalability claims by actual usage. In other words, wide-spread deployment of our software will strengthen our position as a research group in distributed systems.

Whether we like it or not, end users run Windows, not UNIX. Moreover, end users who run Windows are generally very different from those running UNIX. Another motivation for choosing to implement Globule for Windows, is that we expect different requirements for Globule development to surface. For example, where a UNIX user may be expected to manually install and configure a Web server, a Windows user should see only a standard install shield and be offered a simple, easy-to-understand interaction that will get a minimal Globule running. In this respect, the difference between installing Apache under UNIX (requiring the user to go through a configuration file) and under Windows (essentially having to fill in only a domain name), is illustrative. In other words, a Windows implementation of Globule will keep us focused on important user requirements.

Finally, a more idealistic motive is that we do wish to demonstrate that our science can be put to practical use. Globule is one of those (perhaps rare) occasions where it seems that the aforementioned gap between science and user community can be kept small.

3.2 Functionality

We plan to develop a Windows version of Globule that has the following minimal functionality:

- Supports static master/slave replication.
- Supports client redirection to available replicas, even if other replica servers are down.
- Installation requires minimal effort from an end user, and is comparable to the installation of Apache on Windows.
- Supports Web documents that contain either absolute URLs, or relative URLs. Global references relative to Apache's DocumentRoot (i.e., absolute path names as references), are not supported.

The Globule Windows implementation will be released as a major next version, in parallel to the UNIX implementation.

3.3 Static master/slave replication

Globule version 1.0 supports a simple static master/slave replication. In this strategy, a user identifies a master server that is said to *export* a tree of Web documents, defined relative to the server's DocumentRoot. In practice, this tree contains Web pages that can be directly modified by a user and of which the copies are immediately invalidated when a page is updated.

In a typical setup, if Alice wants to use Globule to host a Web site, she will download the software to the machine that is going to act as hosting server. We expect this to be her typical home computer, which has a permanent connection to the Internet (although that machine may regularly be switched off). The master server will be running on her computer, and her Web site will be stored in a folder designated at installation time. For now, we make the assumption that the master server is running on a machine with a permanent IP address (but see below).

Slaves are now set up manually. We plan to change this situation by dynamically allocating slaves through a centralized registry. First, Globule users who require high availability are requested to offer hosting facilities for others by having their machine registered in a slave pool. A user may always refuse to register his/her machine. The consequence is that if the user wants replication, he/she will have to explicitly provide the names of slave machines. We, as project team, also provide hosting facilities by registering several of our machines as slaves.

When Alice requests high availability for her Web site, we dynamically select a number of slaves from the slave pool at installation time. It will also be possible to select specific slave machines manually. For example, a user may install Globule on two computers, one acting as a master and the other as a slave. We will provide a simple interface to make such a configuration possible. We note that users who contribute resources should be allowed to refuse to act as slaves for specific masters (e.g., to prevent having to host illicit content).

In an ideal setup, we can assign slaves from a pool containing only machines from end users. Statistically, the more slaves there are, the higher availability we can offer for hosting a specific site. At this point in time, we do not have any information on the uptime of machines that would participate in Globule. We plan to collect statistics in order to improve availability by proper slave allocation.

3.4 Client redirection to available replicas

In the current system, clients initially contact the master server from where they are redirected to one of the replica servers. If one of the replica servers hosting a specific site is down, redirection may fail. Moreover, if the master server is unavailable, the entire site cannot be reached. For WinGlob, this situation will have been improved.

We plan to let all redirection take place through external redirectors of which one will be hosted at the Vrije Universiteit. Each Globule-supported Web site will need to be identified through a DNS domain. The name server responsible for that domain will handle the redirection by selecting an *available* replica server for the requested Web document, if possible. We assume that the name server itself is highly available using standard primary/secondary DNS replication.

As project team we will provide a name server to handle names in a specific domain such as `globuser.org`. At installation time, Alice will be asked to provide a name for her site, say alice. Provided this name is not yet in use, it may then be registered as `alice.globule.org`, and its servers as `master.alice.globule.org`, `slave1.alice.globule.org`, etc. This simple naming scheme requires some further thought as it may easily lead to name clashes, but the basic idea should be clear.

Obviously, this solution does not scale. Scalability is enhanced by making use of more, independent name servers. As the user base grows, users may also provide name servers (independently from what Globule offers).

We note that we are well aware of the problem of scalable client redirection. During the past 2 years we have been working on a fully distributed scalable solution in which Globule Web servers participate in client redirection [1]. We are currently working on the design and implementation of a component that can be fully integrated into Globule. It is unclear at the time of writing whether this component has been sufficiently matured to incorporate it in the WinGlob release.

3.5 Installation

If there is one important lesson to learn from Globe, it is that deployment requires minimal installation effort from end users. For the proposed project, we plan to pay much attention to the simplicity of installing and maintaining Globule by end users. As a minimal requirement, we will provide an easy-to-use install shield for installing Globule on a Windows machine. Ease-of-use is also dictated by the simplicity of the *model* offered to the users on how they can deploy Globule. For example, a model may contain statements such as

- "Globule allows you to maintain a Web site on your personal computer."
- "In order for others to access your Web site, you need to provide a simple name, such as `alice`. Your site will then be accessible as `http://alice.globule.org`."
- "Globule can automatically copy your site to other machines. Do you wish to make use of this facility?"

These are just examples but should illustrate the point. We plan to develop a simple model and test it with non-experts. Of course, experts will have access to all the details and the actual operation of Globule.

Another point that will receive special attention concerns updating Globule. Just as we need a simple model for installing and explaining Globule, we need a simple model and accompanying implementation for updating Globule. Also, different versions of Globule should be interoperable. One possibility is to allow users to register themselves at Globule so that they can be informed about updates through e-mail. An update then simply involves downloading and reinstalling Globule (while keeping all essential information from previous installs).

The development of an install shield will be based on existing tools, such as Nullsoft's Installation System, which is a freeware product that is also used for installing the Winamp audio player.

3.6 Web documents

Globule currently supports replication of pages that can be completely interpreted at the client side. In particular, it supports pages with absolute and relative URLs, and client-side scripts. It cannot easily deal with pages containing absolute path names as references (i.e., files named relative to a site's DocumentRoot). For WinGlob, we propose to adopt these restrictions unless our ongoing research provides solutions that can easily be incorporated.

3.7 Missing functionality

The most important issue that will not be included into WinGlob is extensive support for security. Currently, we assume that the servers that jointly host a single Web site trust each other. We have built in protection by which servers can mutually authenticate each other, but once authentication has succeeded, we rely on trust.

3.8 Additional functionality

As we do not have extensive experience with developing Windows applications, it is difficult to assess the time and effort to develop Globule for Windows. If time permits, we propose to continue work on version 2.1 by enhancing the basic functionality. There are various possibilities:

- Hosting a Web site on a machine that has a dynamically allocated IP address.
- Provide support for concurrent updates from different locations. This functionality would be needed to create *distributed* Web sites.
- Provide simple tools to enable efficient and high-available Web logging (a.k.a. "blogging").

Other issues may surface during the development of WinGlob. Selection of which functionality to add is done after consultation with NLnet.

3.9 Risks

The Computer Systems group at the Vrije Universiteit has considerable experience in developing advanced systems software for Unices. There is not much experience in developing Windows systems software, although several group members have experience in building Windows applications. This lack of knowledge may endanger the progress of the proposed work. We do note, however, that the API for developing Globule is the one provided by Apache (i.e., libapr). This API reduces the dependencies on specific operating systems.

Another source of potential problems is the portability of Globule modules. Although we have taken the usual measures to ensure portability across UNIX platforms, practice indicates that this provides no guarantees. Moreover, we expect that Globule version 1.0 is based on some hidden assumptions concerning the semantics of various Apache operations that may affect portability even across Unices. For example, we have recently discovered that the way we use shared memory in Apache can probably not work for systems such as HP/UX or Digital UNIX. This may not be a serious problem, but illustrates that there are hidden assumptions. We expect that Globule for Windows will reveal more of these types of problems. It is difficult to assess at this point how easy it is to develop solutions.

3.10 Dissemination

To contribute to the success of WinGlob, we will advertize its availability through regular channels (news groups, mailing lists). Code is made available through sourceforge, as well as the regular Windows freeware sites. By actively advertizing the UNIX version of Globule, we aim at creating awareness among Windows users. For Globule version 1.0, we are advertizing the use of Globule to our own students. Their feedback will be processed first before global announcements.

4 Plan of work

We request funding for 1 FTE to carry out the following tasks:

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

To minimize the risk of not having a release by May 1, 2005, we will start developing the server pool manager and enhancing the Globule DNS server *in parallel* to other Winglows activities as soon as we notice that the planned duration time is not enough. Note that the server pool manager and Globule DNS operate in a UNIX environment. We have enough and experienced programmers to carry out this work.

Costs

The total requested budget is as follows:

Salary scientific programmer, 1 FTE	56.768	incl. 13.1 % insurances
Salary supervisor, 0.05 FTE	5.085	incl. 13.1 % insurances
Overhead (including costs for infrastructure)	8.025	
Total	**€69.878**	

We will take care that a proper hardware and software environment is available to do the Windows development work.

5 Project targets

After extensive consultation with NLnet, it has been decided to aim at the following targets. Targets will be evaluated every 4 months in close collaboration with NLnet.

Time	**Description**
1 Sep 2004	A first version is ported, and adopted by 5–10 testers. Configuration takes place by manually installing binaries, and configuration of the httd.conf file.
1 Jan 2005	30 users have adopted the system. Installation is relatively easy through a simple, graphics-oriented interface, simpler, but comparable to what is offered by Apache. The UNIX version of Globule supports Web site availability in the presence of a disconnected master.
1 May 2005	300 users have adopted the systems. Installation is easy and comparable to the way that Apache offers support. Masters and slave are fully operational, and sites remain available as long as at least one master backup is accessible.

WinGlob will concentrate only on static Web pages. Also, to stimulate dissemination, we aim at articles in magazines Dr. Dobb's and c't.

References

[1] A. Baggio and M. van Steen. "Transparent Distributed Redirection of HTTP Requests." In *Proc. 2nd International Symposium on Network Computing and Applications*, pp. 17–24, Apr. 2003. IEEE Computer Society Press, Los Alamitos, CA.

[2] 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.

[3] 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.

[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] M. Szymaniak, G. Pierre, and M. van Steen. "Netairt: A DNS-based Redirection System for Apache." In *Proc. International Conference WWW/Internet*. IADIS, Nov. 2003.

[6] M. Szymaniak, G. Pierre, and M. van Steen. "Scalable Cooperative Latency Estimation." Technical Report, Vrije Universiteit Amsterdam, Department of Computer Science, Dec. 2003. *Submitted for publication*.