Evolving Interfaces to Impacting Technology: The Mobile TeraGrid User Portal

Rion Dooley¹  Stephen Mock¹  Maytal Dahan¹  Praveen Nuthulapati¹  Patrick Hurley¹
+1 (512) 475-7886  +1 (512) 232-7346  +1 (310) 429-9419  +1 (512) 475-9504  +1 (512) 232-7591
dooley@tacc.utexas.edu  mock@tacc.utexas.edu  maytal@tacc.utexas.edu  praveen@tacc.utexas.edu  phurley@tacc.utexas.edu

ABSTRACT

The TeraGrid User Portal (TGUP) [1] is a web portal that aggregates and simplifies access to TeraGrid information and services for active TeraGrid users. The purpose of the TGUP is to make using the large number of diverse resources and services of the TeraGrid easier for the national open science community, thus increasing their productivity and the impact of the TeraGrid project. As the portal capabilities have expanded and improved TGUP usage surpassed 300,000 hits a month. To continue increasing the impact and visibility of the TeraGrid project and the TeraGrid User Portal the team developed TGUP Mobile. TGUP Mobile is a lightweight, responsive web application providing a subset of TGUP capabilities via a mobile device. This paper describes the architecture, design, and development of the TGUP Mobile application, and examines the community acceptance and synergy created through the development of both the traditional portal and TGUP Mobile.

Categories and Subject Descriptors

H.5.3 [Web]: php
I.7.2 [HTML]: html

General Terms

Design, Human Factors

Keywords

Mobile, TeraGrid, User Portal, Web 2.0, iPhone, AJAX ¹

1. INTRODUCTION & BACKGROUND

Mobile devices are one of the most successful computer-based consumer product of this age and impact the way people interact and connect to their workplace. Mobile web traffic more than doubled last year and industry leader, Cisco, predicts this will continue at least through 2013 [2]. The primary driver behind this growth is the rapidly expanding smartphone market. Today, one third of mobile users own a smartphone with another third indicating they plan on purchasing one as their next mobile device [3].

This increasing trend and dependency on smartphones is changing the way people interact with the Web. Mobile device usage brings with it a different user experience and a different set of user assumptions and requirements. Screen size, user interface design, connectivity, interactivity, and responsiveness are all different on mobile devices compared to traditional web access via a desktop browser. There are two approaches to handling the development of mobile applications and interfaces: building a native application or developing a mobile web application.

For native applications, most mobile operating systems have Software Development Kits (SDKs) and an expanding application catalog. For example, Apple’s iTunes Application Store now has over sixty thousand available applications, Google’s Android Market has over ten thousand, and Blackberry App World has surpassed two thousand entries since its recent launch in March. But in order to develop an application independent of the specific device many services and applications are being built as streamlined web sites appropriate for mobile users. Companies such as ESPN, CNBC, American Airlines, Bank of America, Facebook, YouTube, the Weather Channel and many more have all created mobile versions of their websites.

Taking into account the increased popularity of mobile devices and the need for users to access information outside of the traditional modes of operation the development team released TGUP Mobile in July 2009. The goal of TGUP Mobile is to enable users to interact with the TeraGrid in their increasingly mobile lifestyles. As its name suggests, TGUP Mobile is a web application providing a subset of TGUP capabilities optimized and appropriate for a mobile audience. This paper discusses the design, architecture, and development of the TGUP Mobile application, examines the community acceptance and concludes with planned future capabilities and their expected impact.

2. DESIGN DECISIONS

The TGUP development team focused on developing a mobile application that would have the greatest impact on existing users of TeraGrid. With RIM, the maker of the Blackberry brand of smartphone, and Apple, maker of the iPhone, composing over 78% of smartphone requests on the web in the US [4], it made sense for the Mobile TGUP design to target these platforms first. It is essential that the TGUP team leverage existing knowledge and resources and focus on a cross-platform solution rather than individual solutions for each device. It is also important to recognized that the smartphone market is still rapidly changing. Google’s Android platform and Palm’s Web OS are new entries to the market that are growing in popularity and gaining market

¹ Texas Advanced Computing Center
Research Office Complex 1.101, J.J. Pickle Research Campus
10100 Burnet Road (R8700)
Austin, Texas 78758-4497
share quickly. The decision was to support the devices of today while still leaving an opportunity to support the devices of tomorrow.

To meet these requirements, the development team decided on a web-based solution. The assumptions behind this choice were that all potential users would have an unlimited data plan, or minimally, that a data connection would be readily available to every device accessing TGUP Mobile. Furthermore, it was assumed that every device would have a browser supporting HTML, CSS, JavaScript, and cookies. No assumption was made about the actual device's user interface other than the browser supports traditional mouse events.

The next consideration made was to decide which existing TGUP features fit well within a mobile interface model. Selecting a mobile feature set from the TGUP's existing functionality was a trade-off between hardware capabilities, bandwidth availability, and value to end-users. Popular, interactive applications in the full TGUP like Remote Visualization, SSH Terminal, and File Manager are bandwidth and processor intensive, and require the ability to run Java Applets which makes them impractical candidates, for a mobile device. However, the Systems Monitor, TeraGrid News, and the majority of the portlets in the My TeraGrid section of the portal are lightweight applications suitable for use on a mobile device. Based on the requirements above, the following features were selected for the initial version of TGUP Mobile: TeraGrid Systems Monitor, User Allocation Information, TeraGrid User News, User Profile Service, and Consulting.

3. ARCHITECTURE

While TGUP Mobile provides much the same information as the full TGUP, the application differs greatly from an implementation standpoint. In order to be as efficient and productive as possible the TGUP team leveraged many existing TeraGrid services that are used in both TGUP and TGUP Mobile. The existing TeraGrid service-oriented architecture (SoA) has services in place for job and resource monitoring, user profile discovery, data management, information aggregation, and news distribution. These services already work together and build in a layer of redundancy in the event of a temporary failure in one of the underlying components. These services are accessible through either, and often times both, SOAP and REST interfaces. In addition to these core services, the TeraGrid Central Database (TGCDB) provides auditing and accounting information about users and their actions across the TeraGrid.

By leveraging the existing TeraGrid SoA it greatly reduced the amount of complexity in the TGUP Mobile architecture. Figure 1 shows a diagram of the TGUP Mobile architecture.

The first level of the service stack is security. TGUP Mobile leverages the TeraGrid MyProxy server to authenticate users. MyProxy is open source software for managing X.509 Public Key Infrastructure security credentials [5]. It has become the de facto authentication mechanism for gateways in the TeraGrid.

Visitors to TGUP Mobile have a guest view, which enables access to certain unrestricted information such as System Monitor, User News, and more. Users who authenticate to TGUP Mobile via their TGUP username and password, get access to expanded and personalized information such as their user profile, accounts, allocations, and jobs. The next layer is data abstraction, this creates a interface layer for interacting with data aggregated across the TeraGrid. It is in this layer where information from multiple sources is aggregated into a consistent object representation that can be used by the TGUP Mobile application.

![Mobile TGUP Architecture](image)

The event monitoring and notification services exist as independently scheduled tasks running on the host server. Currently only email notifications are supported in production, however SMS, and Twitter support will also be available in the future. Third party dependencies on corporate service providers for SMS, Tweets, etc have been excluded from the diagram.

The data caching and synchronization services run as scheduled tasks that aggregate information from across the TeraGrid and cache the results to disk. These create the cache files seen in the bottom left of Figure 1. The TeraGrid Central Database (TGCDB), shown in the bottom right of Figure 1 provides user-specific information needed by TGUP Mobile. A set of third party web services shown in the bottom center of Figure 1 provide the remaining information needed by TGUP Mobile.

TGUP Mobile relies on the TeraGrid third party services in Figure 1 to gather information both about the user and about the TeraGrid itself. These services are:

**TeraGrid Integrated Information Services (IIS):** IIS is a federated integrated information service to serve its capability publishing and discovery needs. IIS provides a broad range of information on resources and services across the TeraGrid. TGUP Mobile depends on two services from IIS: kit-services-v1 and tg-resources-v1 [6]. They provide resource names, sites, services, hosts, and endpoints.

**TeraGrid User Profile Service:** The User Profile service is a RESTful TeraGrid web service created explicitly to support the gateway community. It provides authenticated access to user-specific TeraGrid information such as a user’s historical job information, accounts, allocations, profile information for themselves and colleagues.

**GridPort Information Repository (GPIR):** The GPIR is a data persistence service that stores both static and dynamic information about the TeraGrid. It provides the majority of information needed for system monitoring such as resource descriptions,
queues, and load information. The GPIR service is also used in the production TeraGrid User Portal [1].

**Inca**: Inca is a framework to perform periodic, user-level functionality testing and performance measurement of Grid systems. The TeraGrid uses Inca to schedule the execution of information gathering scripts, and to collect, archive, publish, and display data. In the case of TGUP Mobile, Inca provides resources status information such as resource availability and references to news items containing detailed downtime information [7].

**Metadata Directory Service (MDS)**: MDS is an information publishing service aggregating data from across the TeraGrid. MDS provides the real-time queue information published by each Resource Provider (RP) TGUP Mobile uses for job monitoring. [8]

**TeraGrid News**: The TeraGrid News service provides RSS feeds of news items as well as full content articles about TeraGrid events, announcements, and resource outages [9].

## 4. IMPLEMENTATION

Several requirements had to be met when implementing TGUP Mobile. First, it needed to be fast. The latency on mobile networks is significant to begin with, so page serving must be as fast as possible. To keep the web application light and responsive, we chose PHP [10] as our implementation language. This removed the overhead normally found in heavyweight containers and allowed us to prototype quickly and easily transition to production.

Second, because of the bandwidth constraints, pages must be kept small and the number of refreshes minimized. TGUP Mobile uses the iUI framework [11] to provide the JavaScript library, CSS, and stock images necessary to mimic a native iPhone application. In addition to providing a familiar look and feel, iUI loads HTML snippets of code through AJAX calls rather than reloading full pages. This saves bandwidth by not reloading fixed headers and footers as well as supporting images and linked CSS and JavaScript files. Furthermore, TGUP Mobile uses jQuery [12] to simplify the AJAX calls and further reduce the page load size.

Third, the application had to be secure. Mobile devices are lost and stolen more frequently than desktop and notebook computers. The networks they use are always wireless and frequently vulnerable to interception. To address these concerns, TGUP Mobile encrypts all communication using SSL. User authentication information is never sent in the clear, and all session and state information is stored on the server. In the case of inactivity, user sessions time out every 24 minutes.

It is also crucial that TGUP Mobile perform as any expected modern web application, displaying user-centered design, dynamic content, and mashups. The page structure is user-driven and the page content is user-specific. All content is generated dynamically from near real-time data. A single page of data is usually the result of mashing up information from several remote services into one cohesive and comprehensive user view.

Developing TGUP Mobile required the design and consideration of both client and server side pieces. This section describes the implementation considerations in both of these areas.

### 4.1 CLIENT-SIDE

It is important for TGUP Mobile to look and behave like a native application as much as possible while maintaining familiarity to the regular TGUP. With the iPhone and iPod Touch clearly industry leaders in graphical design, TGUP Mobile was styled towards the menu-driven structure familiar on the iPhone. However, due to the standards driven, web-based approach taken to make it available as a web application, the interface is also accessible to any mobile device with a web browser supporting HTML, CSS, JavaScript, and Cookies. Using the JavaScript libraries described above TGUP Mobile accomplishes the required design. Figure 2 shows the main page users see when first visiting TGUP Mobile.

![Figure 2: Main landing page of TGUP Mobile.](image)

The application features both guest and authenticated views. The guest view lists main menu items for User News, Systems Status, Help, and More. User News allows users to view headlines, create custom views and searches, and read full text announcements. System Status displays description, load and status information on all TeraGrid systems. Help contains contact links, FAQ, and feedback links for users to easily get in touch with TeraGrid consulting support using the capabilities of their mobile devices. More contains miscellaneous information relevant to TGUP users. Figure 3 shows unauthenticated views of TGUP Mobile.

![Figure 3: Unauthenticated views of TGUP Mobile.](image)
The authenticated view adds one additional menu item to the main landing page, My TeraGrid. The My TeraGrid section allows users to access user-specific information on the TeraGrid as they would on the regular TGUP. This includes viewing and updating their profile, checking allocations and usage statistics, viewing all their system accounts, and browsing running and finished jobs. Furthermore, users can also request notifications for changes in job states and completions through the job pages. Figure 4 displays authenticated views of TGUP Mobile. The implementation details are discussed in the next section.

4.2 SERVER-SIDE

TGUP Mobile application handles user authentication, session management, information aggregation, and event management in real time at each page request. In support of each of these actions are three background processes that manage caching and notification offline as well as auditing capabilities to provide usage and forensic information. In the remainder of this section, we discuss the activities behind page rendering, the background process, and the auditing.

4.2.1 PAGE RENDERING

While several features of TGUP Mobile such as the Help and More sections are statically generated, the majority of content is dynamically generated with each request. Here is a brief description of the guest features and how the content is generated:

User News: News item listings are generated by aggregating the TeraGrid News RSS feeds published about each institution. The
SimplePie [13] library is used to parse the RSS feeds and maintain a disk cache. Individual news content is obtained by scraping content from the full HTML page views of news items.

**System Monitor:** System loads and summary information are read from an XML file containing the results of a web service call to GPIR. The query is performed by a CRON task, described below, that ensures fresh data. One technical roadblock that needed to be addressed was a namespace mismatch between resource names in GPIR and IIS. GPIR uses full, descriptive names to identify systems. IIS uses an internally defined resource identifier to identify systems. To relate the two resources, we perform a regular expression match between the GPIR resource name and the POPS resource name given by IIS. Given the number of resources, this is a reasonable approach. In the future, as the TeraGrid Resource Description Repository comes online, we will be able to directly match resources.

Individual system descriptions are obtained by directly calling the GPIR web service. This is an extremely fast call that will perform well given significantly more usage. Lastly, real time system outage information is retrieved from Inca using a simple HTTP call. Inca runs a series of tests across the TeraGrid throughout the day. One of the results of these tests is a text file containing any production compute resources currently experiencing down time. This file is downloaded, parsed, and added to the description of appropriate systems.

Authenticated access to TGUP Mobile enables a rich set of functionality to the user. Authentication is performed against the TeraGrid MyProxy server using the user’s TGUP username and password. At login, a server-side session is created to store user-specific information. It is also at this point that a cache of the user’s TeraGrid system information is generated. While not expensive to obtain, this information is used on nearly every request the user makes, thus speeding up nearly every request.

Below, we briefly describe the authenticated features of TGUP Mobile and from where the content originates.

**User Profile, User Accounts and User Allocation:** The information needed for these pages are retrieved from the TeraGrid User Profile service and the TGCDB. The User Profile service provides user-specific information about users. Through a RESTful interface, TGUP Mobile queries for dynamic information about the user’s usage, accounts, profile, allocations, projects and colleagues. From these queries, the TGUP Mobile parses the returned CSV data and uses it to present the appropriate pages to the user.

The User Profile service is a read-only service. In order to update user profile information, TGUP Mobile accesses the TGCDB directly.

**User Jobs:** Job information is retrieved from multiple sources. A background CRON task, described below, is used to create several XML files of current queue information across the TeraGrid. These job cache files are parsed to list the user's active jobs. Active, in this context, means present in some state in the system queue.

This process will continue to improve. Comprehensive job information is not accessible at every TeraGrid site due, primarily, to the fact that not all TeraGrid RP publish queue information. Currently there is no queue information available from NICS, PSC, Purdue, and all Condor systems.

The RP that do publish queue information do so using local usernames, thus in order to locate jobs for a particular user, their username must first be looked up, then compared against the username associated with each job. For features such as viewing all active jobs, this means parsing every job cache file. This is not an ideal solution, but it is the only one currently available in the TeraGrid to find active job information. The portal team is working with TeraGrid information services to work on

Historical job information is acquired from the TGCDB. This is a very time-consuming query. Open-ended queries such as, “select all jobs for user X” may take anywhere from 30 seconds to 2 minutes to complete. To keep the application responsive, the user interface is tailored in such a way that users will refine their search by project, resource, or both to cut down on the query size. Also, to further reduce the query time, all queries are restricted to jobs run in the past year. While listing historical job information is still not a fast page request, these refinements have made it significantly faster. To further improve the response time, optimizations would need to be made to the TGCDB and/or a new discovery mechanism implemented.

**Job Notifications:** Job notifications are managed by a trigger service that creates and manages file-based job registries. A job registry is simply a text file that contains a list of notifications to which a user has subscribed. When a user requests a notification for an individual job, TGUP Mobile fires the trigger service which then adds an entry to the user's job registry file. All registries are then processed by a background CRON task, described below, and used to notify users of job state changes.

### 4.2.2 Background Processes

Three background processes help drive the performance of page requests to TGUP Mobile. These processes are a systems query, a batch queue query, and a trigger processing routine. Each process runs as a CRON task at fifteen-minute intervals. The interval was set at fifteen minutes because much of the information in IIS used to generate the information is updated, at most, every fifteen minutes. This is the maximum effective refresh rate.

**Systems Query:** The systems query process is a PHP script that queries GPIR’s SOAP interface for summary information on the entire TeraGrid virtual organization (VO). Performing a summary query on GPIR often takes more than a second. By caching the results of this query to disk, the most expensive part of rendering System Status page requests is eliminated.

**Batch Queue Query:** The batch queue query process is a shell script that pulls queue information on every TeraGrid compute resource from IIS. This information is obtained by IIS from MDS, cached to disk, and then fetched by the batch queue process. The process is set to run at fifteen-minute intervals to match the refresh rate of the IIS cache.

The alternative approaches to pulling the information from IIS were to query MDS or directly query each of individual compute resource in the TeraGrid for this information. The former would essentially be much more work to obtain the exact same results. The latter would be an extremely time consuming task that would likely fail to execute faster than IIS generates its cache files. This would then cause the job information to be out of sync with the
rest of the information in IIS. Given the practical constraints of data freshness, and complexity, pulling the cached data from IIS was the best approach.

**Trigger Processing Routine:** The trigger processing routine is a PHP script that runs at fifteen-minute intervals immediately after the batch queue query completes. The routine parses each user’s job notification registry and checks it against the existing job cache looking for changes in status. Upon a valid change in job status, the registry entry is updated and a notification is sent to the user. Support is present for email, SMS, and Twitter, though only email is currently available in production due to the lack of access to user cell phone number and Twitter information for TeraGrid users. Progress is being made to make this information available.

### 4.2.3 AUDITING

Auditing is a fundamental requirement of all TeraGrid services. It is the first step in examining security, usage, and trends in the TGUP. In TGUP Mobile, auditing is done primarily through log files. A log is kept for nearly every activity in TGUP Mobile. This approach was chosen over an approach such as Google Analytics strictly for performance reasons. Analytics is an effective solution for certain web sites, however the latency on each page request (which is performed by the client) proves to be noticeable in a mobile application. We chose speed over a more robust auditing mechanism and implemented our own lightweight analytics solution.

Logging information for TGUP Mobile is gathered at a page level. With every page request information on the requesting device, platform, IP address, browser, page, authentication status, username, time, and date is collected and stored. This satisfies our basic policy requirements and provides a simple way for us to track all of the statistics necessary for security, reporting, and development needs.

### 5. COMMUNITY ACCEPTANCE

Since its release in late June 2009, the initial response to TGUP Mobile has been positive. Usage has grown from 900 unique hits the first month roughly 2000 unique hits in the next month. The most popular features thus far are User News, System Status, and User Jobs. iPhones, as expected, are generating the overwhelming majority of hits.

Several unannounced feature updates have been released since the initial production announcement on June 29. The addition of active job information was released in late July and job notification support was added early in August. We anticipate that both of these features, as well as an outreach and education effort leading up to SuperComputing 2009, will significantly increase traffic.

### 6. CONCLUSION & FUTURE WORK

The continuing goal of the TGUP team is to expand the impact and value of the TeraGrid to the scientific community. The goal of TGUP Mobile is to expand the reach and visibility of TeraGrid. TGUP Mobile is a powerful, feature-rich mobile alternative to the full TGUP. It is built upon the existing TeraGrid SoA and is optimized for the iPhone. User response early on has been positive. Over the next few months we expect a significant growth in usage due to increased exposure to the user community and expansion of TGUP Mobile features.

As new features are added to TGUP, they will be migrated to TGUP Mobile when appropriate. In the coming months we plan to add support for managing user consulting tickets, managing training class registrations and viewing video content, and sharing and managing files across the TeraGrid through a virtual home space. As well, a new TeraGrid Service, the Resource Description Repository, will come online in October allowing TGUP Mobile to transition to an integrated resource information service.

### 7. ACKNOWLEDGEMENTS

This work is supported by NSF award OCI-0503697 to TeraGrid’s Grid Integration Group (GIG). The TeraGrid User Portal is funded by the NSF and includes input and participation from TeraGrid Partner Institutions. The user portal team thanks TeraGrid users for their suggestions, feedback, and support in making TeraGrid a highly used, productive service for the national community. The development team also acknowledge the NSF-funded Open Grid Computing Environments (OGCE) project for the technology contributions to the User Portal project.

### 8. REFERENCES

10. PHP, URL: [http://php.net](http://php.net)
