May 27, 2004

APPLICATION FOR 2004 LARRY L. SAUTTER AWARD FOR INNOVATION IN INFORMATION TECHNOLOGY

UNIVERSITY OF CALIFORNIA, RIVERSIDE
COMPUTING & COMMUNICATIONS STATUS PAGE

Submitted By:
Larry McGrath, Director, Computing Support Services

Project Team:
Phyllis Bruce  Bart Kats
David Gracey  Mike Kennedy
Bob Grant  Josee LaRochelle
Russ Harvey  Sohail Wasif
Amber Jones

SUMMARY

When there is a disruption of your Email service or online ordering system you need to know immediately where the problem exists. To provide this information the System Status web page was developed to enable you to view important information regarding the tools used to conduct your daily campus business. The site is organized to enable you to quickly view information on critical systems.

As UCR’s enterprise-wide technology services organization, Computing and Communications (C&C) provides a wide variety of technology services and support to campus faculty, staff, and students. C&C is the campus provider for Email, Financial Service applications, Internet Services, student resources like PAWS and GROWL, Blackboard online courses, and the entire network infrastructure for UCR: if you are doing it online, C&C is making it possible, and all with one of the most robust networks in the UC system. These systems are critical to the day-to-day operation of the campus and with this new notification system users will always know the status of critical network resources.

PROJECT DESCRIPTION

https://statmon.ucr.edu/status

C&C has developed a methodology by which the health of critical systems is monitored:

- Software agents were developed to monitor health of critical systems
- Data stored in relational database provides concrete and historical data for development of service level agreements
- Email and page messaging alerts support staff to system problems
- Web page is updated regularly. Clients can view the status of systems that affect their units.
Typically, systems are checked every 5-10 minutes. The agents have been internally developed and report back to the database the health of each system. This methodology of centralizing and reporting system health is state of the art and 100% home developed. The system provides Email notification to support staff and messages to users when systems go down.

Perceived Need

C&C saw the need for a notification system whereby campus management and IT staff could quickly determine the status of critical campus systems. In addition the new system is easy to read, provides more information, and is better organized.

Intended Audience

The system is intended for executive management and leads in business function areas to monitor critical systems, and to support the establishment of service level agreements with units on campus.

Project Overview

The project was begun in October 2003.

- *API was designed for the systems being monitored*
- *API sends a status message to an Event Monitor*
- *Event Monitor sends data to Oracle database*
- *Event Processor reads the data and determines problems. Email and page message are sent to support staff*
- *Status Page draws data from database and updates the web page*
- *Management Console can be used to create custom status pages*

The Status Page offers users a quick “at-a-glance” opportunity to determine the status of campus system resources in 4 major categories: Academic and Student Applications; Email; Financial/Administrative Applications; and Network Infrastructure. These resource categories represent groups of related services, applications and infrastructure systems. Each category shows the summary status of the systems contained therein, and users can expand the categories to view information on specific resources.

Notes on Design and Architecture

The diagram below shows a block representation of the overall architecture of the Status system. All of the system elements center on the status database. Each of the elements seeks to modify the database in such a way as to record events from remote and local agents or SNMPtraps, notify the correct people in the event of a system failure, or allow viewing of the current status of the various services being monitored.
All status monitoring in the system is done by agents or via SNMPtraps. An Agent is simply a process running on a machine that can monitor anything from CPU load to the uptime of a particular service. What an agent actually monitors is entirely up to the creator of the agent. These agents send updates periodically regarding the status of what it is monitoring to the Event Monitor. The SNMPtrap gateway receives SNMPtrap messages and converts these traps into status messages to be sent to the event monitor.

**Agents**
Agents serve as the primary reporting mechanism for problems and other status changes within the system. These agents communicate via a status messaging protocol with the Event Monitor.

**Protocol States**
The status messaging protocol consists of two states, the authentication state (AUTH) and the status messaging state (MESG).

Once a connection is established, the client agent issues an AUTH command to send a password to authenticate itself before sending status messages. The server responds with AUTH OK if the password is correct or AUTH BAD and closes the connection if the password is bad. Once authentication has succeeded, the client is free to send any number of status messages to the server. Each message sent will be answered by the server with a MESG OK if the status message was accepted or a MESG BAD if there was a problem accepting the message. A
message may not be accepted if there is a formatting error, an invalid resource id or if a particular value is not within specific parameters. The client can continue to send status messages until it drops the connection.

**Status Messaging API**
Agents can be written in any language, provided that they follow the protocol outlined above. However, most agents will be written in PERL, and as such an API has been developed in the form of a PERL module to implement the status messaging protocol.

**Example Agent**
Below is an example agent (the StatusMessage Module) written in Perl. The Perl API allows one connection to be used to report events from any number of resource ids.
```perl
#!/usr/local/bin/perl

use UCR::StatusMessage;

$sm = new UCR::StatusMessage("statuspw", "statmon.ucr.edu", "7777");

unless($sm->connect()) {
    print $sm->lastError() . "n";
    exit(-1);
}

$sm->message(1111, 600, "memory usage normal");
$sm->message(3333, 600, "cpu 1 at 90%", 3, 10);
$sm->heartbeat(2222, 600);
$sm->ignoreEvents(4444, 600);
$sm->close();

This script acts as an agent reporting status messages for resource ids 1111, 2222, 3333, 4444 and 5555. After creating the initial UCR::StatusMessage object, the script attempts a connection to the event monitor running on mon.ucr.edu port 7777.

The first message sent is simply an informational message stating that the memory usage level on the machine is normal. It has a time to live (ttl) value of 600 seconds (10 minutes). No severity is given so it sends the default level of 0.

The second message is a count type message; this is known because of the addition of the last parameter, a count value of 10. This means that if the server receives this message 10 times, it will then register in the database as an active event and its type set to normal.

The third message sent is a heartbeat. The ttl value here is again 600 seconds. If the script doesn’t send another heartbeat before the ttl expires, the status message server will generate a heartbeat failed message for the resource id.

The final message tells the status message server to ignore events from agents about resource id 4444.

**The Web Interface** Two main pages provide information for users:
Status Main View
Columns provide information on system status:

**Resource**
The name of the category or individual system

**Status**
The status of a system or collection of resources is represented by a colored bar. The color refers to the system health. This bar can also represent partially up or partially down. A solid green bar for a collection of resources means all systems under that group are up.

**Last Changed**
Is the last date/time that the current status began (so if it shows green, it has been green since this date/time). This also includes the change from any scheduled maintenance where the systems have a planned outage.

**Last checked**
How recent is the information you are viewing. The status represented is as of this date/time. The average status check is every 5-10 minutes so the last checked times may differ for each system.

**Messages**
Any information pertinent to the resource will be posted on an as-needed basis.

Quick Stats Page
This section displays UCR's most commonly used systems with icons representing the following items: E-Mail, Calendar, UCR Web, iLearn, UCRFS. Selecting one of these icons will change the corresponding status display in the resource detail section. The displayed systems can be changed as needed.

Customer Satisfaction

“The new status system is helpful because it alerts system administrators when it detects a system is down. It's much nicer to find that out from an automatic e-mail message than to suddenly get flooded with calls. Plus it gives us a chance to solve the issue before the campus users even notice there's a problem.”

Matt Giers, Programmer Analyst, UCR Computing and Communications

“The new status system will provide a much needed method for UCR Executive Management to check on the status of mission-critical information technology systems. The web interface is easy-to-navigate and intuitive in design. This is a model effort for system status design at all UC campuses.”

Joel Nylander, Associate Director of Computing Support Services, UCR Computing and Communications

"…a way to determine accurately the up status of a system, be it academic or institutional in nature…now anyone can find out what's up and what's down."

Dr. Leo Schouest, Manager Faculty & Student Technical Support, UCR Computing and Communications

“The new status page will allow a user to quickly see the current condition of UCR applications and/or procedures. By drilling down into specific resources a user will quickly be able to tell if the system or procedure their working with is in the ‘Up’ status. Potential users of UCR applications will now be able to more effectively manage their time in the rare case when an application and/or procedure is in the ‘down’ status.”

Larry Ott, Programmer Analyst, UCR Computing and Communications
APPENDIX

Status Main View
Systems are grouped by top-level categories. Expand the categories to display the status of specific systems.
Quick Stats Page
Quick Stats shows the status of frequently used systems including Email and the Blackboard learning management system. This page can be customized for the needs of specific users.