

**Second Australian Open Computing in Government Conference
18 – 19 April 2005
Canberra**

Presentation by Steve McInerney
Health/Insite Systems Administrator
www.healthinsite.gov.au

Using Open Source Software – the HealthInsite Experience

Introduction

Health/Insite is the Australian Government's Internet gateway designed to provide consumers with easy access to reliable, high quality and relevant information about health and wellbeing so that they can make more informed healthcare decisions.

Health/Insite links users to information on the web sites of approved information partners, each of which have been assessed by an independent Editorial Board to ensure the quality, currency and relevance of the information they provide. Current information partners include some of Australia's most authoritative government and non-government health organisations.

Since its launch in April 2000, Health/Insite has been expanded to cover over 800 major health topics, and links to almost 12,000 resources from 77 respected health organisations.

During 2004, Health/Insite received approximately 1.4 million visitors who viewed 9.6 million pages. Currently, in early 2005, we are attracting 8-9000 visitors viewing 40-50,000 pages a day. Recent estimates put the number of Australians visiting Health/Insite at approximately 70-80% of all visitors to the site.

This paper will examine:

- the relationships involved in managing and using Health/Insite;
- communication between users and the managers of Health/Insite;
- methods used to monitor and evaluate the web site;
- system efficiency from a business perspective.

It will also demonstrate how Open Source Software is used to solve the issues raised.

Relationships

There are four major groups of people involved in producing the end system that is Health/Insite:

- *Consumers* - The end users of the system and hence the targeted audience. Feedback from users is actively encouraged as an indicator of the extent to which HealthInsite is meeting user needs.
- *Information Partners* - Provide the information resources to which HealthInsite links.
- *Editorial Team* - Provides content and technical management support. All of the work to manage and organise the resources available from the

site is done by this team. The team also develops the policy that determines the strategic directions that the site will pursue.

- *Technical Team* - Provides infrastructure maintenance, management and support including programming, database and server management.

In addition to these four groups, an independent Editorial Board ensures that all information partners are approved in accordance with specific quality criteria.

None of these groups acts in isolation. It is critically important that methods of communication between all these parties are efficient and fast. Effective communication between the Editorial and Technical teams is particularly crucial to ensure that the site functions well.

Communication

The communication requirements include:

- Rapid and easy contact mechanisms for both individuals and groups (groups may have thousand of members)
- Automated reporting on system status and failures
- Access on a 24x7x365 basis
- Ease of use
- Speed to enable timely response to identified problems
- Cost effectiveness
- Reliability

As a communication medium, email meets the above business requirements and has the additional benefits of enabling backups, archives and multiple interfaces to access the stored information.

To adequately address these issues several related solutions were required. After some investigation of various packages, we settled on the following elements to deliver the identified requirements:

- *Postfix* for the general email handling. "Postfix attempts to be fast, easy to administer, and secure, while at the same time being sendmail compatible enough to not upset existing users".
- *Mailman* for email-list management. *HealthInsite* currently maintains seven active lists. Some are announce only lists while others are full discussion lists. *Mailman* provides a complete set of features for list managers and provides an easy to use management interface.
- *Courier-IMAP* using the IMAP protocol to centralise all email stores for ease of backup and access. Being a back-end service, *Courier-IMAP* can be a little more arcane to configure and manage, however, after investigating other IMAP products, *Courier-IMAP* was the most appropriate fit for our needs.
- *SquirrelMail* a web mail front-end to the IMAP store, for remote world wide access.

A significant advantage of this architecture is that the four major components are all easily replaceable with similar products, of which there are several in each category. For example, we could substitute *Sendmail* for *Postfix*. This is an obvious advantage as we can choose the tool that best meets our immediate needs and easily replace it should our needs change or when a more appropriate tool becomes available.

Monitoring and Evaluation

If *Health/Insite* is to increase its popularity, return users need to be able to rely on the system being available to them as and when they need it. Like all popular systems, if it is not reliable, users will lose confidence. From a business perspective, it is simply unacceptable for a system failure to occur, regardless of the cause, without the technical team being alerted to it immediately. It is critical that we are able to constantly monitor and evaluate the status of the system. This includes:

- knowing when critical services fail
- identifying emerging issues before they trigger system failures
- ignoring routine indicators in favour of reporting on exceptions across a range of indicators
- reporting on system usage
- providing web site statistics over time
- conducting capacity planning based on extrapolated estimates of future needs

In our case any solution must also deal with the fact that we cannot do SNMP (Simple Network Management Protocol) through our firewalls. No Traps. No Queries.

Given that communication via email is such a major component of the *Health/Insite* system, it is highly desirable that any alerting and/or reporting is also done via email. The obvious danger is that of email itself failing.

It should be stressed that a failure alert that the system is "*Not Working*" **also includes** "*Not Working Fast Enough*". This latter issue is particularly relevant when dealing with a web site where speed of access is an important indicator.

Within *Health/Insite* all of these problems are solved by using a suite of tools, both general and highly specialised. We have not found a single "one size fits all tool" for this task and we rely on several different alerting tools for detecting failures within the various parts of *Health/Insite* system.

The simplest tool we use scans through the log files generated by the various systems and emails any entries that either do not match known "good entries" or alternatively do match known "bad entries". The tool used to provide this very 'low tech' but very effective alerting mechanism is known as *logcheck*.

A far more sophisticated tool is *Nagios*. *Nagios* is used for providing a variety of system feedback, reporting and alerting tasks. A simple plug-in called *Apan* provides additional graphing capability. Within *Nagios* we use the email alerting capability to notify slow response times (eg Web Server), service unavailable,

disk free space alerts and so on. *Nagios* is also accessible via a simple web interface.

The other major tool we use is aimed at providing detailed graphs of system level statistics over time rather than reporting faults provided by *Nagios*. This particular tool (which is actually several unrelated tools), is known as *Orca*. It reports, for example, bits per second on disk or network activity and CPU usage. *Orca* also includes some of the more exotic kernel level statistics like “Mutex Counts” or “Disk Inode Percent Usage”. Again, all this information is accessible via a web interface and is, at most, 15 minutes out of date.

Monitoring network activity is another area of concern. Here we rely on two unrelated tools. The first, *Arpwatch*, simply reports, via email, on new machines connecting to our network. It also assists in tracking IP address usage. *Arpwatch* is also useful for identifying cabling errors between otherwise isolated networks.

The second network tool used is *Argus*. *Argus* is extensively used within the ISP and related industries for calculating network traffic volumes. We use *Argus* to monitor all traffic entering and leaving the *Health/Insite* network and hence can verify billed volumes. Being, in effect, a permanent ‘network sniffer’, *Argus* is very useful for debugging the more esoteric network and server problems that occasionally appear. The data section of the traffic is not stored, just the headers, so user privacy is not compromised.

Analysing *Health/Insite* web site statistics provides information about end user behaviour, the impact of enhancements and indicators of the extent to which user needs are being met. We primarily use three separate tools to extract the major areas of interest.

1. For general usage statistics we use the very popular tool *Webalizer*.
2. For tracking *how* users come to *Health/Insite* and to some extent, *why* they come to *Health/Insite*, we use a program known as *Relax*. This tool produces the best referral and search engine analysis of any tool we have found, including both commercial and Open Source tools.
3. To determine how many actual **people** visit *Health/Insite* we use the tool known as *Visitors* in combination with an *Apache* add-on module, *mod_usertrack*. This provides an accurate counts of visitor numbers. It also tracks repeat vs. new visitors and summarises their usage.

System Efficiency

Research confirms that users will go elsewhere if a web page takes too long to load. It is essential that response times are as fast as possible, regardless of the technology available to the end user.

It should be stressed here that these responses are not just web site responses. There are many areas within the overall system that comprises *Health/Insite*, where a response needs to be fast. With any complex system, failures will occur as hardware reaches end of life or as system capacity is

reached. A means for alerting site managers to a potential problem and providing the maximum amount of information is essential to assist in developing solutions in a timely manner.

Because Health/Insite centralises alerting via email and reporting via an internal web site, the various teams have immediate access to information at any time. Health/Insite staff can access graphs of the uptimes for Health/Insite for 2005 or 2004 or both; the rate of disk usage on all production servers for the past 15 months or the past 90 minutes; how rapidly a page is being retrieved from Health/Insite for the last hour, day and month; identify how many referrals from Google Australia to Health/Insite there have been last week.

Health/Insite has experienced situations where a part of the system has failed. Routine monitoring functions have enabled rapid identification and diagnosis allowing appropriate action to be initiated. Some standard diagnostic questions include:

- What was the server in question doing prior to the failure? CPU bound? Disk bound?
- Have disk SCSI errors been logged?
- Has something else failed that may have cascaded?
- Are we suffering a Denial of Service Attack?

One tool that has had a major positive impact on the efficient running of Health/Insite is known as *rsync*. This program enables highly customisable file synchronisation between machines. With this tool, maintaining duplicate data between the various disparate systems becomes a trivial matter with a fully automated solution. In the production server farm, only a single server has programmer developed changes applied to it. These changes are then automatically applied to the other servers. This is just one example where *rsync* has increased efficiency.

Conclusion

All Health/Insite system areas work seamlessly together to ensure that Health/Insite stays available and hence maximises the experience of the Australians and others who access the site. Additional features are rapidly and easily progressed into production, faults and failures are identified and fixed rapidly and potential problems are solved before they become failures.

The vast majority of the technical components that make up Health/Insite are Open Source products. Those components are completely invisible to a casual visitor to the site, but without them, it is likely that Health/Insite would function with much less efficiency and reliability.

Links

Health/Insite Summary Statistics

http://www.healthinsite.gov.au/static/Health/Insite_Statistics

Open Source Packages

It should be pointed out that this is by no means a complete list of all the Open Source packages used within Health/Insite. The complete listing of all Open Source packages used runs into the hundreds, if not thousands.

Apache: <http://httpd.apache.org/>
Samba: <http://www.samba.org/>
OpenLDAP: <http://www.openldap.org/>
Perl: <http://www.perl.org/>
PHP: <http://www.php.net/>
Python: <http://www.python.org/>
Postfix: <http://www.postfix.org/>
Courier-IMAP: <http://www.courier-mta.org/imap/>
Mailman: <http://www.list.org/>
SquirrelMail: <http://www.squirrelmail.org/>
RRDTool: <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/>

Nagios: <http://www.nagios.org/>
Apan: <http://apan.sourceforge.net/>
Argus: <http://qosient.com/argus/>
SEToolkit: <http://www.sunfreeware.com/setoolkit.html>
Orca: <http://www.orcaware.com/orca/>
Arpwatch: <http://www-nrg.ee.lbl.gov/>
Logcheck: <http://sourceforge.net/projects/sentrytools/>

Subversion: <http://subversion.tigris.org/>
RCS: <http://www.cs.purdue.edu/homes/trinkle/RCS/>

BIND: <http://www.isc.org/index.pl?sw/bind/>
ISC DHCP: <http://www.isc.org/index.pl?sw/dhcp/>
NTP: <http://www.ntp.org/>
RSYNC: <http://samba.anu.edu.au/rsync/>

Webalizer: <http://www.mrunix.net/webalizer/>
Relax: <http://ktmatu.com/software/relax/>
Visitors: <http://www.stedee.id.au/visitors/>