ASF's Karst Index Database

— Under the Bonnet —

Michael Lake

26th Biennial ASF Conference
January 2007, Mt. Gambier SA

Outline

This paper will cover the technology that underlies the ASF's server and the KID, details on why we have two KIDs, the documentation, development and testing, backup and security. Future work will be suggested.

The ASF KID

The ASF's KID contains information on 6600 caves and karst features, 2400 cave maps, 192 organisations, 1308 persons, references to 925 articles and covers 355 caving areas in Australia. It is accessible at http://www.caves.org.au

The KID was written by a professional programmer engaged by the ASF in 2000. It went online in January 2001 as a read-only system. The programmer was contracted again over the period 2003 – 2005 to add updating ability.

Updating has started in NSW by SUSS and in Victoria by VSA.

The Software Licence

The ASF's KID software is released under the GNU General Public License (GPL).

A number of overseas groups have already downloaded the software or expressed interest in our code; the Database Commission of the Portuguese Speleological Federation, the Swedish Speleological Federation and a Russian Speleological Society.

The data stored in the database of course remains the property of the ASF and contributing clubs and is not subject to the GPL.

Examples of the KID

Saved web pages of KID Interface:

- Searches
- Updating
- Documentation
- Administration

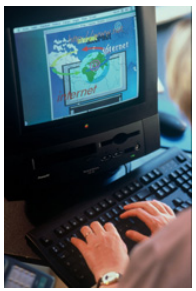
The ASF Server

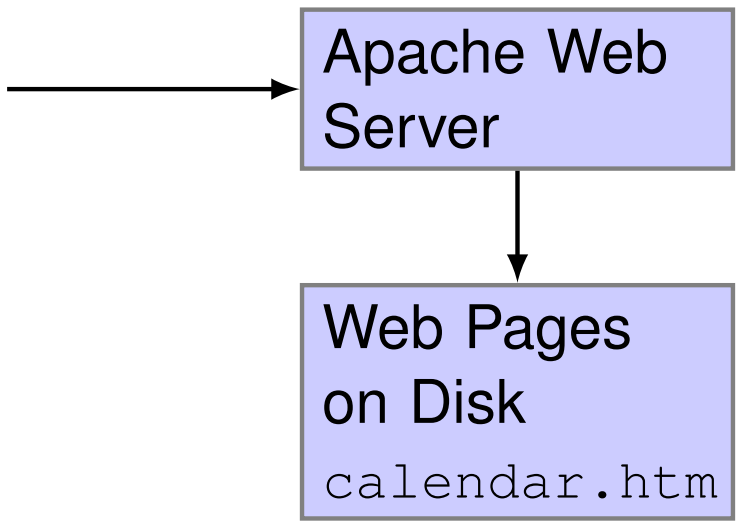
The ASF Server is a virtual server running Debian GNU/Linux.

A virtual server is an operating system hosted by a physical computer which runs multiple operating systems simultaneously. At the hosting provider the real, physical machine is running RedHat Enterprise Linux in a large data centre.

How Web pages are Delivered

A simple static web page request e.g.
http://www.caves.org.au/calendar.htm

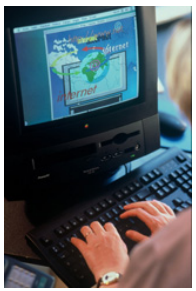




User requests a web page, web server looks for the page.

How Web pages are Delivered

A simple static web page request e.g.
http://www.caves.org.au/calendar.htm



Apache Web Server

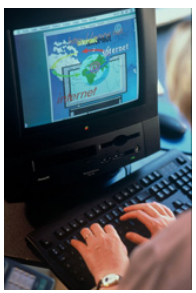
Web Pages on Disk

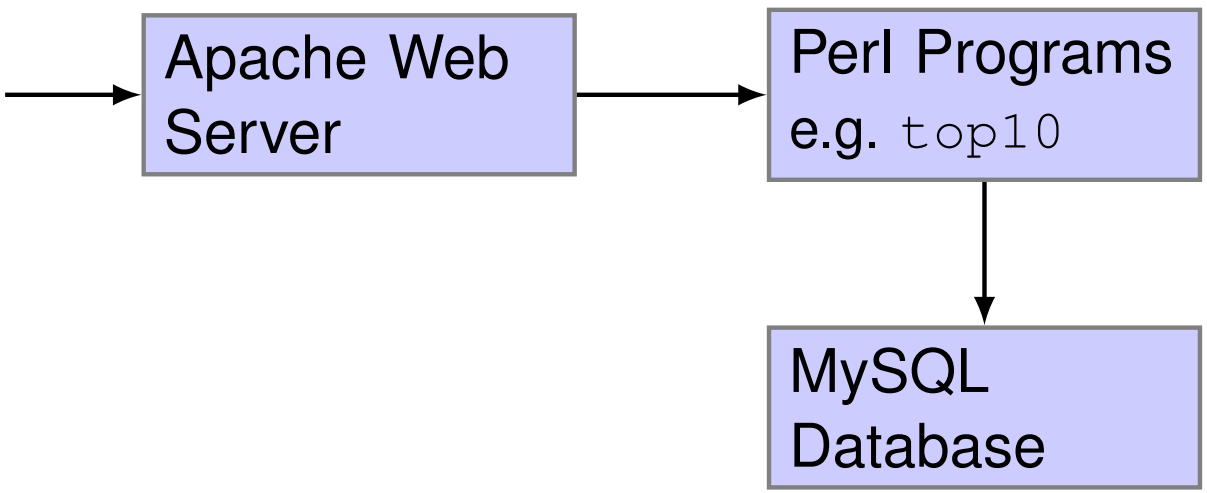
`calendar.htm`

Web server reads the web page & sents it back to the user.

How Web pages are Delivered

A request for a dynamic page e.g.
http://www.caves.org.au/kid/search/top10

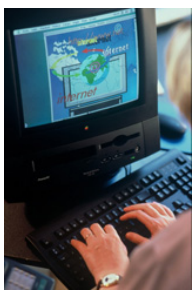


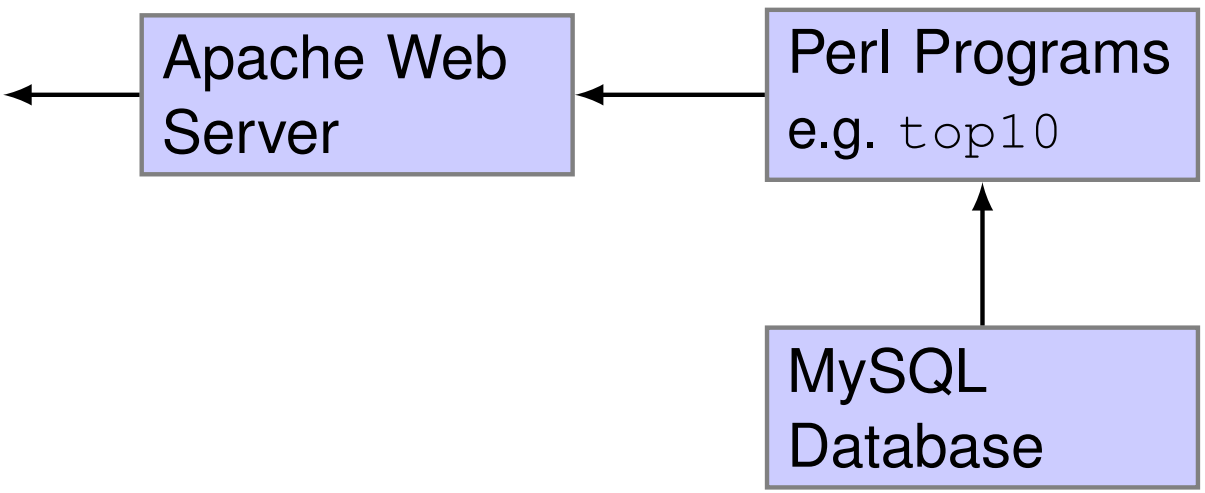


User requests a web page . . .

How Web pages are Delivered

A request for a dynamic page e.g.
http://www.caves.org.au/kid/search/top10





Web pages are sent back to user . . .

The Web Server

We are using the most popular server in the world, Apache (http://www.apache.org). This versatile, high-performance web server is used by about 63% of all web sites in the world.

It features a modular design and supports dynamic selection of extension modules at runtime. Some of its strong points are its security, range of possible customization, dynamic adjustment of the number of server processes, and a huge range of available modules including many authentication mechanisms, access control, caching and many more.

The Perl Programs

The KID uses Perl (http://www.perl.org) and it's these perl programs that do most of the work.

Perl programs generate the web pages that are sent to the user when a user visits the KID website, takes a users query and sends it to the database, gets back lots of data and turns it into nicely formatted web pages and then hands the pages to the web server to send back to the user.

The choice of this language reflects the programmers preference and what was mature and available in 2000 when the KID was first commissioned. Alternatives might have been PHP or Java. Today Ruby or Python would be strong contenders.

Perl Modules

Perl has thousands of small modules that can be used instead of writing your own routines. They are open source, well written and free. We are using modules that provide:

- Authentication and authorization
- Reading config files and parsing command line arguments
- Interfaces for Perl to access databases
- Comma separated values manipulation routines
- SQL parsing and processing
- Routines for processing CSV files
- Interfaces for RSA and MD5 encryption
- Parsing and extracting information from HTML documents
- APIs to the client side of various protocols i.e. email, ftp etc
- Persistence for Perl data structures
- Template processing
- Routines to produce PDF

among others.

Perl Modules

All of those Perl modules are released under an open source licence so we can use them in the KID at no charge. Thanks to the following authors of those modules:

- Abhijit Menon-Sen
- Andy Lester
- Tim Bunce
- Jochen Wiedmann
- Jeff Zucker
- Gisle Aas
- Steve Peters
- Rick Welykochy

and many more.

Important Perl & Apache Modules

To provide separation of the business logic from the user interface we are using a Perl module called Template Toolkit (http://www.template-toolkit.org). This neatly separates the HTML code from the Perl code, making it easier to program, and makes for a cleaner and more easily maintained site.

The other important module we are using is an Apache module `mod_perl`. This integrates Perl into the Apache web server! It can produce anywhere from a 400% to 2000% speed increase on sites using perl scripts, and is used on many large script-based web sites like http://slashdot.org.

The Database

The database used is MySQL (http://www.mysql.com). This is a fast, lightweight and full-featured database which is also reasonably easy to administer.

The database stores the data on the caves, maps, organisations, people and cave areas in tables.

Also stores the user access privileges of the users such as the guest user and the updaters from various clubs.

There are over 500 fields structured into 74 tables. Most of these are the same as the draft database fields and tables published by the Informatics Commission of the IUS.

The Database

```
$ mysql -e "desc PE0000" kid
+--------------------------+-------------+------+-----+---------+-------+
| Field                    | Type        | Null | Key | Default | Extra |
+--------------------------+-------------+------+-----+---------+-------+
| person_id                | varchar(10) | NO   | PRI |         |       |
| surname                  | varchar(30) | YES  |     | NULL    |       |
| usual_first_name         | varchar(20) | YES  |     | NULL    |       |
| middle_initial           | varchar(5)  | YES  |     | NULL    |       |
| initials_for_given_names | varchar(10) | YES  |     | NULL    |       |
| title                    | varchar(20) | YES  |     | NULL    |       |
| address_line_1           | varchar(50) | YES  |     | NULL    |       |
| address_line_2           | varchar(50) | YES  |     | NULL    |       |
| address_line_3           | varchar(50) | YES  |     | NULL    |       |
| address_line_4           | varchar(50) | YES  |     | NULL    |       |
| city_or_suburb           | varchar(25) | YES  |     | NULL    |       |
| state_code               | char(2)     | YES  |     | NULL    |       |
| postcode                 | varchar(12) | YES  |     | NULL    |       |
| country_name             | varchar(25) | YES  |     | NULL    |       |
| country_code             | char(2)     | YES  |     | NULL    |       |
| email_address            | varchar(30) | YES  |     | NULL    |       |
| phone_numbers_prefix     | varchar(15) | YES  |     | NULL    |       |
| home_phone_number        | varchar(18) | YES  |     | NULL    |       |
| work_phone_number        | varchar(18) | YES  |     | NULL    |       |
| mobile_phone_number      | varchar(18) | YES  |     | NULL    |       |
| fax_phone_number         | varchar(18) | YES  |     | NULL    |       |
| pager_phone_user_number  | varchar(20) | YES  |     | NULL    |       |
| organisation_code_1      | char(3)     | YES  |     | NULL    |       |
| organisation_code_2      | char(3)     | YES  |     | NULL    |       |
| organisation_code_3      | char(3)     | YES  |     | NULL    |       |
+--------------------------+-------------+------+-----+---------+-------+
25 rows in set (0.00 sec)
```

Two KIDs

There are two Karst Index Databases;

Production: http://www.caves.org.au/kid
Development: http://www.caves.org.au:8080/kid

They have separate code bases and backend databases.

The development one is used by myself for new code development and testing and by updaters as a “play area” for practice in updating.

We try not to break the production one. 

Documents for Updaters

The documentation page is at http://www.caves.org.au/kid/doc
There are two sections: “Documents for Updaters” and “ASF KID Specifications”.

Documents for Updaters consists of a general introduction and a series of tutorials on how to update caves, people, areas and organisations. These are in PDF format so they can be printed out and referred to easily.

The PDF documents are written using LaTeX – a high level typesetting language which is designed for the production of high quality, technical documents.

ASF KID Specifications

The ASF KID Specifications section is the complete documentation in HTML format for all the database fields used in the KID and the database table schemas.

These HTML pages are produced on-the-fly from the KID database itself. The advantage of this is if the database is changed, this part of the documentation is automatically updated.

These fields and their definitions are based on the field definitions defined by the Informatics Commission of the International Union of Speleology, of which the ASF is a member.

Example: Cave and Karst Definitions

System Documentation

Detailed installation and maintenance documentation is available in HTML format on the software page at http://www.caves.org.au/kid/docs/software. This describes how to install the KID, how to maintain a KID installation, and how to make updates and releases.

For every Perl module that the programmer has written the object classes and methods are documented in the module code itself. The system has scripts which generate HTML formatted documentation from the module documentation in a standard format for all modules. Other third party Perl modules which we have used also have their HTML documentation in the same format on that page.

Source Code Management System

All KID software is managed by a Source Code Management system: Subversion http://subversion.tigris.org

A SCM system allows allows many developers to work collaboratively on the same code, tracks the history of changes including which changes have been made by which developers and allows one to revert to any previous version.

I can work on the KID code on my laptop whilst on the train and later connect to the ASF server and commit my changes. The SCM system knows exactly what files I have changed and where.

Inbuilt Regression Testing

The KID also has some inbuilt regression tests available to test that code changes have not broken the system.

Before a code change the output pages of several searches can be saved. After code changes the search pages afterwards are compared to the ones before. If changes to the search pages are not expected the test suite should report that the files match.

Inbuilt Regression Testing

The CSV (comma separated variable) upload functionality allows a user to upload bulk cave updates.

There are many errors that could occur in a bulk file generated or edited outside of the KID system. For instance in a CSV upload from a spreadsheet a record in the file may have unbalanced quotes, a field may be missing or an extra field may have been inserted.

For each possible error that could occur is a test file with that error deliberately inserted. During the test each of these test files is loaded and the test suite should flag an error.

Backups

Regular backups of the data and the KID software are critical!

A backup script runs automatically as a cron job at about 4 am Sydney time each week. This sends a copy of the current software and data to a site in the US (kindly provided by the Informatics Commission of the IUS).

Also included with the software and data is documentation to assist a System Administrator in a recovery procedure.

Backups

The following is done each time the backup script runs:

1. The current KID software is copied to a backup directory.
2. A dump of the entire database is performed, the data is compressed and placed in the backup directory.
3. Logs of the above operations are written to the backup directory.
4. A checksum is generated for the backup files.
5. All these backup files are copied to remote servers.
6. An email is sent to the KID System Administrator to indicate if the backup was successful or not.

Backups

```
Subject: ASF KID Backup Successfull
From: kid@www.caves.org.au
To: mikel@speleonics.com.au
Date: 1 Jan 2007 04:00:44 +1100

Hi

The KID backup script /home/kid/bin/backup_asfkid at
caves.org.au successfully made a backup of database
kid on 2007.01.01.  There is a checksum
asf-kid-data-v1.32-2007.01.01.cksum with the data file
that can be used to check that data transfer was
successful.

Admin, caves.org.au
```

Security

Two main security concerns for the ASF server:

- The KID contains sensitive information such as records of aboriginal remains in caves or, in the future, exact location information.
- We don't want the server to be compromised and a trojan installed or an extra login account created. Such trojaned servers are referred to by crackers as being '0wned'. They can be used to mount subsequent attacks against other servers such as government or military sites.

Security

Security was considered by the web programmer during the coding stages and a number of measures taken in the KID code. For instance, all user input into the KID is filtered to guard against SQL injection attacks.

Additionally the following policies are adhered to:

- Minimal services are run on the ASF Server.
- The server is kept up-to-date with security patches.
- Intrusion detection systems are used. These compute multiple checksums for important files on a system.
- All login access is via secure shell version 2 and the use of public/private keys.

Future Work

- Generate summary information for all caves in an area for use on PDAs by updaters
- Production of State-based Karst Index Books in PDF format
- Internationalisation
- The Next Generation KID

References

- ASF “Australian Karst Index 1985”, Edited by Peter G. Matthews ISBN 0 9588857 0 2
- The ASF Karst Index Database http://www.caves.org.au
- GNU General Public Licence http://www.gnu.org/licenses/licenses.html
- Informatics Commission of the Union International de Spéléologie http://www.uisic.uis-speleo.org
- LaTeX Project Site http://www.latex-project.org

Author Contact Information

Michael Lake

2 Derribong Place

Thornleigh NSW 2120

Sydney, Australia

MikeL@speleonics.com.au