



SOLID AS A ROCK

Selecting a Web Environment For OpenVMS

Mark Daniel

Sponsor: **VSM** Software Services <http://www.vsm.com.au/>

Employer: Defence Science and Technology Organisation

DECUS Australia Symposium 2002

Novotel Hotel Olympic Park NSW Australia - Home of the 2000 Olympics

21 – 23 July 2002



SOLID AS A ROCK

An Impartial Summary :-)

by the author of the WASD Web Package

Opinions my own and on loan from others.



DECUS Australia Symposium 2002

Novotel Hotel Olympic Park NSW Australia - Home of the 2000 Olympics

21 – 23 July 2002



Objectives

1. **Emphasize the availability and suitability of OpenVMS as a Web platform.**
2. **Suggest considerations for evaluating Web packages.**
3. **Examine current VMS options and their comparative characteristics.**
4. **Briefly consider strategies for reducing the impact of differences between them.**
5. **Not necessarily make recommendations!**



Assumptions

Some experience with or exposure to ...

OpenVMS

Web technologies

For example; no time will be taken to explain the differences between ASTs and POSIX Threads, latency and throughput, Perl and PHP.



Mud Map

Introduction

Why VMS!

Considerations

Package Evaluation

Development Environment

Summary

Questions



Disclaimer

As the author of the WASD package it is probably impossible for me to be completely dispassionate discussing this topic.

To the best of my knowledge the information in this session is accurate (or near-enough so). Any errors or omissions are not there deliberately to position one package to the detriment of another.



Why 'The Web'?

Marketing

E-Commerce

Publication

Entertainment

'Middleware'

Ubiquity



Why VMS?

**Well,
Why Not!**



Why VMS!

- **You already have it!**
 - Then why not use it as a Web front/back-end?
- **Currently evaluating?**
 - VMS can provide COTS Web solutions
- **Bet the business ...**
 - Many significant sectors do!
- **Can't afford to be down?**
 - VMS clusters and disaster-tolerant solutions



Why VMS!

- **Security**

- VMS has proved itself highly resistant to penetration
- O/S architecture is designed to provide fine control of activities, even down to what memory can do what!

- **Contagion Proof**

- It's xenolithic nature and robust immune system mean pandemics sweep past OpenVMS not through it!



Can VMS really do what I need on the Web?

as always ...

that depends!

**and hopefully that's what this session will go
some way to answering**



Selected Web and VMS-Web History

- 1989** *Tim Berners-Lee & CERN*
- 1991** **CERN at SLAC**
- 1994** **OSU from OSU**
- 1995** *Apache (nee NCSA)*
Purveyor and Cheetah
- 1996** **WASD (then named HFRD)**
- 1997** **Netscape FastTrack**
- 2000** **CSWS (OpenVMS Apache)**
- 2001** **Tomcat**



Servers Available and 'Working'

| | |
|----------------------|----------------------------|
| CERN | <i>long-since obsolete</i> |
| Cheetah | <i>retired</i> |
| CSWS (Apache) | <u>active development</u> |
| FastTrack | <i>retired</i> |
| GFR HTTPd | available |
| OSU | <u>active development</u> |
| Purveyor | <i>retired</i> |
| WASD | <u>active development</u> |
| Xitami | available |



Package Selection Considerations

1. **Purpose**
2. **Hardware**
3. **VMS Version**
4. **Dynamic Content**
5. **Security**
6. **Load**
7. **Support**
8. **Miscellaneous**



Selection Considerations

Note that some of these will be technical issues, others non-technical.

This session may help with the technical ones but the non-technical are out-of-scope, except to point out they may be considerations.



Selection Considerations - Purpose

- **Publication**
 - Static pages
 - Dynamic pages
- **Data(base) connectivity**
 - Front-end
 - Middle-ware
 - Back-end
- **Ad Hoc serving**



Selection Considerations - H/Ware & VMS Version

- Alpha
 - VAX
 - (IA64)
 - Hobbyist
 - Enterprise
 - Memory
 - CPUs
 - Dedicated
- Alpha
 - VAX
 - Pre-6.0
 - Pre-7.1
 - Pre-7.2
 - Pre-7.3



Selection Considerations - Dynamic Content

- **Scripting** (in autonomous processes)
 - Ease
 - Languages / Environments
 - Standards
- **Pages** (interpretation of markup language)
 - SSI
 - PHP
 - JSP



Selection Considerations - Security

- **Authentication**
 - Credential sources
- **Access Control**
 - Who can do what to which and when
- **Transaction Privacy**
 - Secure Sockets Layer



Selection Considerations - Load

- **Concurrent Requests**
 - Peak
 - Average
- **Response Content**
 - Static
 - Dynamic
 - Script
 - Database



Selection Considerations - Support

- **Documentation**
 - Package
 - Third-party
- **User Community**
 - News Groups
 - Mailing Lists
- **Installation / Update**
 - Availability
 - Ease



Selection Considerations - Miscellanea

- **Policy**

- 'Standard' or mandated product?
- Contractual support

- **Skills Base**

- Site administration
- Developer requirements

- **Comfort Zone**

- Package satisfaction
- Longevity



Package Evaluation

CERN

long-since obsolete

Cheetah

retired

CSWS (Apache)

active development

FastTrack

retired

GFR HTTPd

available

OSU

active development

Purveyor

retired

WASD

active development

Xitami

available



Package Evaluation - CSWS

- **Compaq Secure Web Server**
- **Apache* for OpenVMS**
 - * Currently the most popular webserver on this small, blue-green planet.*
- **Ported & supported by VMS Engineering**
- **Available since 2000**
 - currently in it's third generation (v1.2)
- **Concurrency using multiple 'child-processes'**
- **Alpha only**
- **VMS 7.2-1 or later only**



Package Evaluation - OSU

- **aka. DECthreads Server**
- **Dave Jones of Ohio State University**
 - established and active user community
- **Available since 1994**
- **Single process**
- **Concurrency using POSIX Threads**
 - allows VMS kernel threading
- **Alpha or VAX**
- **VMS 5.5-2 through to 7.3** (min v3.6b for V7.2 or later)



Package Evaluation - WASD

- **Named after a now-extinct race of engineers**
- **Mark Daniel* of DSTO**
 - * That's me, just in case you missed the start of the session!*
 - established and active user community
- **Available since 1996** (under development since 1994)
- **Single process**
- **Concurrency using VMS ASTs**
 - employs other core VMS technologies
- **Alpha or VAX**
- **VMS 5.5-2 (pre-v7.0) through to 7.3**



Package Evaluation - Purpose

- **All three packages provide static and dynamic Web page generation**
- **All packages will interface (at least via scripting) to DBMS such as RDB and Oracle**
- **All have a long development history providing a broad spectrum of services to their users**



Package Evaluation - Hardware & VMS Version

| | <u>Alpha</u> | <u>VAX</u> | <u>5.5-2</u> | <u>6.0</u> | <u>6.1</u> | <u>6.2</u> | <u>7.1</u> | <u>7.2</u> | <u>7.3</u> |
|-------------|--------------|------------|--------------|------------|------------|------------|------------|------------|------------|
| <u>CSWS</u> | X | | | | | | | X | X |
| <u>OSU</u> | X | X | X* | X | X | X | X | X** | X** |
| <u>WASD</u> | X | X | X*** | X | X | X | X | X | X |

* *pre-v3.6*

** *post-v3.6*

*** *pre-v7.0*



Package Evaluation - Scripting

| | <u>CGI</u> | <u>Perl</u> | <u>PHP</u> | <u>Tomcat</u> | <u>Python</u> | <u>Persistence</u> |
|-------------|-------------------|-----------------------------------|-----------------------------------|------------------------|---------------------|--------------------|
| <u>CSWS</u> | Yes (wrapper*) | Module** (persistent) | Module (persistent) | Module (persistent) | Yes (wrapper*) | Yes |
| <u>OSU</u> | Yes (wrapper*) | Yes (wrapper* & persistent) | Yes (wrapper* & persistent) | No | Yes (wrapper*) | Yes |
| <u>WASD</u> | Yes (native) | Yes (persistent) | Yes (persistent) | Coming | Yes (persistent) | Yes |

* A wrapper is a DCL procedure that provides the required support environment.

** Apache provides some core and all additional functionality using discrete code 'modules'.



Digression - Scripting

Why is 'persistence' so important?

- **Process activation expenses**
 - Latency
 - CPU cycles
- **Scripting engine initialization**
 - Latency
 - CPU cycles



Digression - Scripting

So what is 'persistence' then?

- **The ability of the server to reuse resources (such as processes) over multiple requests**
- **A scripting/processing engine retaining it's initialized state over multiple requests**



Digression - Scripting

Interestingly, all three packages have had to tackle this issue (each in it's own inimitable fashion)

- **CSWS**
 - child-processes and loadable modules
- **OSU**
 - reusable DECnet processes and HPSS
- **WASD**
 - reusable detached processes and CGIplus



Package Evaluation - Authentication

| | <u>Local</u> * | <u>SYSUAF</u> | <u>PKI</u> ** | <u>Other</u> *** |
|-------------|----------------|---------------|---------------|------------------|
| <u>CSWS</u> | Yes | Module | Module | Module |
| <u>OSU</u> | Yes | Yes | Perhaps**** | Yes |
| <u>WASD</u> | Yes | Yes | Yes | Yes |

* *Package-unique username/passwords.*

** *Public Key Infrastructure (X.509, etc.)*

*** *User-written authentication support.*

**** *Requires building some experimental modules.*



Package Evaluation - Access Control

| | <u>Virtual Server</u> | <u>Path</u> (allow/deny) | <u>Method</u> (read/write) | <u>Discretionary*</u> |
|-------------|-----------------------|-----------------------------|-------------------------------|-----------------------|
| <u>CSWS</u> | Yes | Yes | Yes | Require/Satisfy |
| <u>OSU</u> | Yes | Yes | Yes | No |
| <u>WASD</u> | Yes | Yes | Yes | 'Conditionals' |

** 'Discretionary' in the sense access is based on some evaluation of characteristics of the request other than the path or authenticated user.*



Package Evaluation - Secure Sockets Layer

| | <u>Which</u> | <u>How</u> | <u>PKI*</u> | <u>Support**</u> |
|-------------|--------------|--------------|--------------|------------------|
| <u>CSWS</u> | OpenSSL | Module | Yes | Yes |
| <u>OSU</u> | OpenSSL | DIY*** | Perhaps***** | Yes |
| <u>WASD</u> | OpenSSL | Optional**** | Yes | Yes |

* *Public Key Infrastructure (X.509, etc.)*

** *Support for self-signed certificates, OpenSSL swiss-army-knife, etc.*

*** *Not part of the official package; requires obtaining and building the full OpenSSL kit.*

**** *May be built either as an SSL or non-SSL executable.*

***** *Requires building experimental modules.*



Package Evaluation - CPU Load

| | <u>Server</u> | <u>Concurrency</u> | <u>Multi-CPU</u> | <u>Scripting</u> |
|-------------|--------------------|--------------------|-------------------------|-------------------------|
| <u>CSWS</u> | Multiple Processes | Per-Process** | Per-Process | Child Process |
| <u>OSU</u> | Single Process* | POSIX Threads* | VMS Kernel Threading*** | Reused DECnet Process |
| <u>WASD</u> | Single Process* | VMS ASTs* | Multiple Instances**** | Reused Detached Process |

** It is broadly acknowledged that, all other things being equal, a single process with internal concurrency is significantly more efficient than multi-process concurrency.*

*** To support 150 concurrent requests CSWS requires a minimum of 150 processes.*

**** On VMS 7.1 or later.*

***** WASD v8.0 supports cooperating, multi-process, per-CPU serving.*



Package Evaluation - Support

| | <u>Package Document</u> | <u>Third-Party Document</u> | <u>News Groups</u> | <u>Mailing Lists</u> | <u>'Official' Support</u> | <u>Distribution</u> |
|-------------|-------------------------|--|--|----------------------|---------------------------|-----------------------------|
| <u>CSWS</u> | Copious | Lots & lots & lots & lots! | comp.os.vms comp. infosystems... | Multiple | Yes | Internet PCSI CONDIST |
| <u>OSU</u> | Meager |  | comp.os.vms | Single | No | Internet ZIP TAR |
| <u>WASD</u> | Copious | | comp.os.vms | Single | No | Internet ZIP |



Product Evaluation - Installation

| | <u>Distribution</u> | <u>Kit</u> | <u>Installation</u> | <u>Configuration</u> |
|-------------|-------------------------|----------------------------|--------------------------------------|-----------------------------------|
| <u>CSWS</u> | Internet CONDIST | PCSI | PRODUCT INSTALL* | DCL Procedure -plus- Manual |
| <u>OSU</u> | Internet | TAR archive ZIP archive | Compile+Link** | Manual |
| <u>WASD</u> | Internet Freeware CD | ZIP archive | Link-only*** -or- Compile+Link | DCL Procedure -plus- Manual |

* **CSWS is distributed with pre-built executables.**

** **OSU is distributed only as source files and requires DECC to build it.**

*** **WASD is distributed as source and optional pre-compiled object modules. This allows a site to fully build the package using DECC or link-only the available object modules.**



Development Environment

What can be done to smooth-out differences between packages, maximize content sharing and portability, while minimizing migration issues when the inevitable looms?



Development Environment

- **Of course package administration specifics cannot be avoided when using any server**
- **Site content is a different matter ...**
 - All packages serve static pages relatively uniformly
 - Dynamic pages are a different issue
- **Choose cross-platform, 'standard' scripting and/or dynamic content tools**
 - This should allow platform migration, and content-sharing, relatively seamlessly



Development Environment

- **Common Gateway Interface (CGI)**
 - Lowest common denominator scripting
 - Supported by all servers
 - Of course the underlying O/S may present migration difficulties ... **SO**
- **CGI using** *(all 'interpreted', cross-platform scripting languages)*
 - Perl
 - PHP
 - Python



Development Environment

- **Persistent Scripting**

- There is no generic, cross-platform, persistent scripting environment
- Instead use a persistent implementation of
 - Perl
 - PHP
 - Python
 - JSP (Java Server Pages)
- Endeavour to deploy a persistent DBMS agent



Development Environment

- **Dynamic Pages**

- SSI (Server Side Includes)
 - Notoriously platform-specific
 - If you must use them, use a 'standard' subset
- Perl, PHP and Python *with page templates*
 - Cross-platform
 - deploy or re-deploy across MS Windows, Unix, VMS
- JSP (Java Server Pages, aka. Tomcat)
 - Cross-platform
 - deploy or re-deploy across MS Windows, Unix, VMS



Development Environment

Other cross-platform advantage

- The original and obvious example is HTML
- Others are
 - XML
 - SOAP
 - LDAP
 - Java



Development Environment

Although this is a *little* out of scope ...

- **Avoid IDEs that insist on server 'extensions'**
 - 'Dreamweaver' works well with generic environments
 - 'FrontPage' tends to assume IIS and an MS platform
- **When client-side processing is required**
 - 'JavaScript' is more universal than say
 - 'ActiveX' which needs a MS platform



Summary

It seems obvious that CSWS, OSU and WASD are all mature and quite capable packages.

No specific package recommendation is made.

Each may have specific strengths or advantages in given environments or for given tasks.

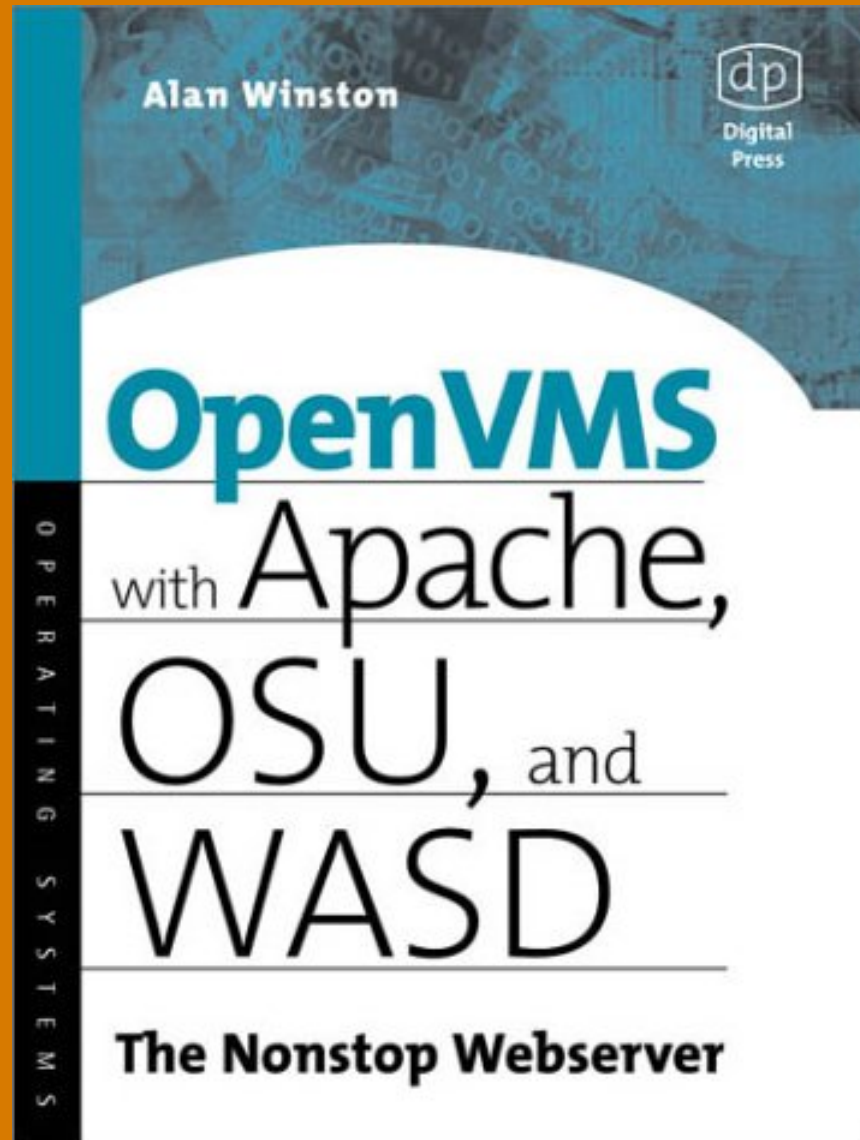


Summary - Longevity

| | |
|-------------------------|-----------------------------------|
| <i>CERN</i> | <i>come+gone</i> |
| <i>Cheetah</i> | <i>come+gone</i> |
| Apache | 7 years (2 as CSWS on VMS) |
| <i>FastTrack</i> | <i>come+gone</i> |
| OSU | 8 years |
| <i>Purveyor</i> | <i>come+going</i> |
| WASD | 8 years |



Recommended Reading



Author: Alan Winston
Manager Central Computing
Stanford University
Synchrotron Radiation Lab

Publisher: Digital Press

ISBN: 155582648

(See me after the session for a flier)



References

OpenVMS

<http://www.openvms.compaq.com/ebusiness/Technology.html>

CSWS

<http://www.openvms.compaq.com/openvms/products/ips/apache/csws.html>

OSU

<http://www.er6.eng.ohio-state.edu/doc/serverinfo.html>

WASD

<http://wasd.vsm.com.au/>



SOLID AS A ROCK

QUESTION TIME

Thanks for attending!

VSM



DECUS Australia Symposium 2002

Novotel Hotel Olympic Park NSW Australia - Home of the 2000 Olympics

21 – 23 July 2002