Open Source update

Camiel Vanderhoeven and Brett Cameron

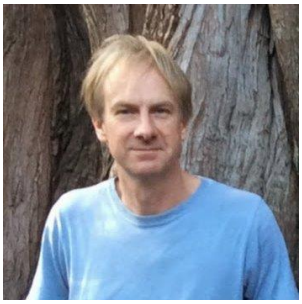
September 2016

Abstract

Providing and supporting more Open Source software on OpenVMS is an integral component of VSI's roadmap for the future of the OpenVMS operating system. In this talk Brett will provide an overview of work that has been done by VSI to date, and will present details of VSI's current and future plans and activities around Open Source, including information about Open Source products that VSI are intending to port (or are currently working on) and support, additional services that will be provided to help OpenVMS users to make best possible use of available Open Source technologies, and information on how VSI are engaging with the wider community around various Open Source initiatives. Camiel will then provide an update on the Java 1.8 port, including an overview of some of the challenges that have been encountered to date, current status and schedule, important differences between 1.6 and 1.8. Camiel will also show a few simple examples that illustrate some of the new language features that are available in Java 1.8 and will discuss future plans for Java on OpenVMS.

About me

Brett Cameron works as a senior software engineer at VMS Software (http://www.vmssoftware.com), helping to define and implement the company's Open Source strategy for the OpenVMS operating system. Prior to joining VMS Software, Brett worked as a senior software architect with HP's Cloud Services and Enterprise Services groups. Brett lives in Christchurch, New Zealand and has worked in the software industry since 1992. In that time he has had experience with a wide range of software technologies, many of which have long since been retired to the software scrapheap of dubious ideas. Over the past decade Brett has spent considerable time travelling the world helping organisations to modernize legacy application environments, to integrate the old with the new, and to better leverage Open Source technologies. Brett has been involved in several interesting Open Source projects, and he has been responsible (or should that be irresponsible) for porting various pieces of Open Source software to the OpenVMS platform. Brett holds a doctorate in chemical physics from the University of Canterbury, and maintains close links with the University, delivering guest lectures and acting as an advisor to the Computer Science and Electronic and Computer Engineering departments on course structure and content. In his spare time Brett enjoys listening to music, playing the guitar, and drinking beer.



3

AGENDA

- **Programming languages**
- Integration technologies
- Databases
- Web
- Libraries/utilities
- Software development tools
- Cloud
- UNIX compatibility
- Analytics
- Add-ons
- Other considerations
- Summary/conclusions
- Questions

Programming languages

- Scripting languages
  - Lua
    - Kit available (5.2.3)
  - Perl
    - Craig Berry and others in the community do a great job here
    - Happy to leave it in their capable hands
  - Tcl
    - It's there if anyone wants it (somewhat old version)
  - Python
    - Part way through a port of 3.5; John Malmberg also working on a port
  - Ruby
    - Beta kit available (2.2.2)
  - PHP
    - Updated kit available (5.6.10), includes more extensions…
  - JavaScript
    - Ideally node.js
    - Currently have V7 and (old) SpiderMonkey interpreters


- Also need to consider tools and packages commonly used with these languages

Programming languages

- Interpreted languages
  - Scala
    - Uses the JVM
    - Have an old version that works with Java 6
    - Need Java 8 to run later versions (more on this topic later)
  - Clojure
    - Comments as per Scala
  - Erlang
    - Potentially a good fit with OpenVMS
    - Can get good support
    - Can probably add Elixir (http://elixir-lang.org/) in here too
    - See my talk "Porting Erlang to OpenVMS and getting it to do something useful"

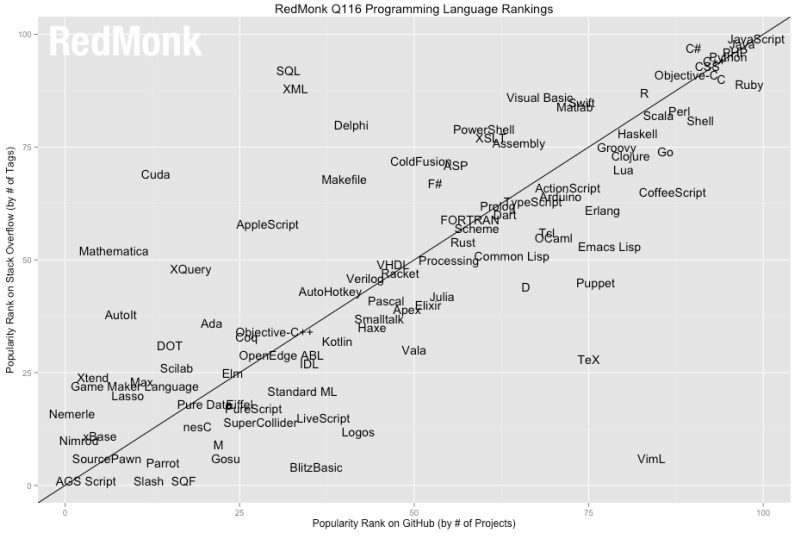
- All the above are seeing increased adoption

Programming languages

- Compiled languages
  - Go (seeing rapid adoption)
  - Rust (relatively new)
  - Apple Swift

- Prerequisites (not all are required in all cases)
  - LLVM backend
    - Plus language runtimes
  - Tweaks to C and C++ compilers
    - Support for latest language standards (C++)
    - Support for some GNU C/C++ extensions
  - Updates to C RTL and threads library

Programming languages



1. JavaScript
2. Java
3. PHP
4. Python
5. C#
6. C++
7. Ruby
8. CSS
9. C
10. Objective-C
11. Shell
12. Perl
13. R
14. Scala
15. Go
16. Haskell
17. Swift
18. Matlab
19. Clojure
20. Groovy
21. Visual Basic

See http://redmonk.com/sogrady/2016/

AGENDA

- Programming languages
- **Integration technologies**
- Databases
- Web
- Libraries/utilities
- Software development tools
- Cloud
- UNIX compatibility
- Analytics
- Add-ons
- Other considerations
- Summary/conclusions
- Questions

Integration technologies

- Data-level integration
  - FreeTDS (SQL Server, Sybase)
  - UnixODBC
    - Also need to implement drivers
  - Not Open Source, but don't forget about partners like Attunity
  - Various others

Integration technologies – message queuing

- Protocols:
  - AMQP (0.9.1 and 1.0)
  - MQTT
  - STOMP
  - CoAP (maybe)
- Products:
  - Kits are available for…
    - Mosquitto broker and Paho client
    - ZeroMQ
    - ActiveMQ
  - RabbitMQ (requires Erlang) and libRabbitMQ
  - Kafka
    - Requires updated Scala
    - Could provide a C/C++ client
  - …




11

Integration technologies

- Web services
  - gSOAP
    - Updated to gSOAP 2.8.22 and MOD_GSOAP updated for CSWS 2.4
  - AXIS2
  - RESTful services (libcURL is useful here)
  - …
- API-level integration
  - Various WSIT enhancements (see later)
  - …
- CIFS/Samba
  - Not exactly an integration technology, but might as well stick it in here
  - Bill Pedersen doing a great job here

12

AGENDA

- Programming languages
- Integration technologies
- **Databases**
- Web
- Libraries/utilities
- Software development tools
- Cloud
- UNIX compatibility
- Analytics
- Add-ons
- Other considerations
- Summary/conclusions
- Questions

Databases

- SQL/relational
  - PostgreSQL
  - MySQL and/or MariaDB
    - Existing ports need updating, hardening
  - Ingres
  - …
- NoSQL
  - Riak (requires Erlang; potentially a good match with OpenVMS and clustering)
  - MongoDB (needs updated C++)
  - Cassandra (Java)
  - Many, many other possibilities in this space
- Caching
  - Redis (certainly client APIs)
  - MemcacheD
    - Have an older version and client API

In addition to the databases and caches themselves, it is also important to provide client API's

AGENDA

- Programming languages
- Integration technologies
- Databases
- **Web**
- Libraries/utilities
- Software development tools
- Cloud
- UNIX compatibility
- Analytics
- Add-ons
- Other considerations
- Summary/conclusions
- Questions

Web (clients and servers)

- Web servers:
  - Apache HTTPD
    - New version in Maynard release (CSWS 2.4, includes OpenSSL 1.0.2 support)
  - Mongoose
    - Version 6.4 built on OpenVMS
  - Tomcat
    - Need Java 7 or 8 in order to update
  - Nginx

- HTTP clients:
  - cURL, libcURL
    - Kit available from VSI (7.49.0, includes OpenSSL 1.0.2 support)
  - Other

- Also need to be thinking about HTTP/2
  - Mark Daniel has added some of this to the latest WASD

AGENDA

- Programming languages
- Integration technologies
- Databases
- Web
- **Libraries/utilities**
- Software development tools
- Cloud
- UNIX compatibility
- Analytics
- Add-ons
- Other considerations
- Summary/conclusions
- Questions

Libraries/utilities – building blocks

- YAML, JSON, XML, …
- Google Protocol Buffers (and several similar such technologies)
- OpenSSL
  - Updated in Hudson release to OpenSSL 1.0.2
- Gearman

- Libffi (Foreign Function Interface)
  - See https://sourceware.org/libffi/
  - Has been ported before
  - Used with the likes of Python to interface with 3GL code

- Numerous other API's and utilities

AGENDA

- Programming languages
- Integration technologies
- Databases
- Web
- Libraries/utilities
- **Software development tools**
- Cloud
- UNIX compatibility
- Analytics
- Add-ons
- Other considerations
- Summary/conclusions
- Questions

Software development tools

- IDE's
  - eCube (Eclipse-based)
  - Other options
    - Possibly do more with NetBeans and Distributed NetBeans
    - Something like CodeBlocks and Uniwin perhaps
- Source code control
  - Git
    - Partial implementation available (supports most commonly used functions)
  - Subversion (beta kit available)
  - Mercurial
  - CVS (old, but simple to use, still a viable option)
- Testing tools
- Continuous integration
  - NXTware Remote for Jenkins from eCube
- Open Source package management (along lines of tools provided by Cygwin or Ubuntu)
- …

AGENDA

- Programming languages
- Integration technologies
- Databases
- Web
- Libraries/utilities
- Software development tools
- **Cloud**
- UNIX compatibility
- Analytics
- Add-ons
- Other considerations
- Summary/conclusions
- Questions

Cloud

- Client APIs and CLIs to facilitate interaction with cloud-based services
  - Amazon EC2/AWS
  - Google
  - OpenStack (HPE Helion, Rackspace, …)
- API support for services such as:
  - Iron.MQ
  - Amazon SQS, SNS, …
  - Dweet.io
  - Xively
  - …
- Containers
  - Longer-term (x86) maybe

It is not really Open Source, but being able to hook into cloud-based services is something that is of considerable interest, and technically it is not too difficult to do.

AGENDA

- Programming languages
- Integration technologies
- Databases
- Web
- Libraries/utilities
- Software development tools
- Cloud
- **UNIX compatibility**
- Analytics
- Add-ons
- Other considerations
- Summary/conclusions
- Questions

UNIX compatibility (GNV)

- Needs updating and expanding
  - Can leverage good work done by the community
  - Needs to be properly supported
  - Currently need to install base package and then apply a significant list of updates/patches…
- Probably a separate discussion…

AGENDA

- Programming languages
- Integration technologies
- Databases
- Web
- Libraries/utilities
- Software development tools
- Cloud
- UNIX compatibility
- **Analytics**
- Add-ons
- Other considerations
- Summary/conclusions
- Questions

Analytics

- R (extensible programming language for statistical computing)
  - Part-way through a port
  - Looking promising
- Apache Spark, Apache Flink, Kafka, …
  - Many of these Java-based technologies require higher versions of the JVM
- The likes of Big Data and Internet of Things need to be key focus areas

AGENDA

- Programming languages
- Integration technologies
- Databases
- Web
- Libraries/utilities
- Software development tools
- Cloud
- UNIX compatibility
- Analytics
- **Add-ons**
- Other considerations
- Summary/conclusions
- Questions

Add-ons (value-adds)

- More OpenVMS-friendly APIs for some Open Source products

Many OpenVMS users do not have developers that are able to incorporate C-based API's into their legacy application code, which may be written in languages such as COBOL, Fortran, or BASIC. It is therefore important to provide wrapper APIs that can be more readily used with these languages in a more OpenVMS-like way.

Add-ons (value-adds)

- OpenVMS-specific extensions for languages such as Python, Ruby, Lua, PHP, Erlang, …
- Integration with other OpenVMS-specific technologies
  - ACMS
    - Better access to audit trail data
    - Mechanism(s) for passing objects greater than 64K (have a working solution)
  - WSIT
    - Support for protocols other than ICC
      - AMQP
      - …
    - Support for other languages such as C#.NET (requires additional protocol support)
  - RTR (maybe)
  - UAF-based authentication (Mosquitto, RabbitMQ, …)
  - …
- Monitoring as a service

AGENDA

- Programming languages
- Integration technologies
- Databases
- Web
- Libraries/utilities
- Software development tools
- Cloud
- UNIX compatibility
- Analytics
- Add-ons
- **Other considerations**
- Summary/conclusions
- Questions

Other considerations

- Prioritization
- Value to customers (usefulness)
- Resources
- Prerequisites (updates to the C RTL, C/C++ compilers, …)
- Community involvement
  - Which Open Source packages do VSI port, maintain, and support; which packages are left to the community; collaboration?
- Input from community and customers
- IP considerations
- Open Source licensing considerations
- Support
- Consulting services and training
- Documentation
- …

Value considerations

What Open Source software is likely to be of greatest use to current OpenVMS users?

- Short term
- Medium term
- Long term

Need to take a strategic approach and consider industry trends

- Internet of Things
- Big Data
- Containers
- Cloud
- …
- Where does OpenVMS provide advantages?

Many OpenVMS users have very old application environments

- Need to provide software (and potentially services) that will help these users

Value considerations

What Open Source software will help to solidify OpenVMS' position with existing users?

- Integration technologies
  - Some users with legacy applications have very little perception of what can be done from an integration perspective
- Technologies that present opportunities to reduce 3rd party license and/or support costs
  - Open Source replacements for expensive, poorly supported, or unsupported software
  - …
- Users on Alpha looking to move to Integrity who need replacement options for technologies not available on Integrity

Value considerations

What Open Source software is likely to attract developers to OpenVMS?

- Modern language environments such as Ruby, Python, Go, Erlang, Rust, Node.js, Scala, Clojure, latest Java versions, …

Modern toolsets

- IDE's and related development tools
- Source code management
- Testing tools
- …

What Open Source software is likely to encourage developers to port applications to OpenVMS?

- As per the above
- The option of a good UNIX shell and related utilities (enhancements to GNV)

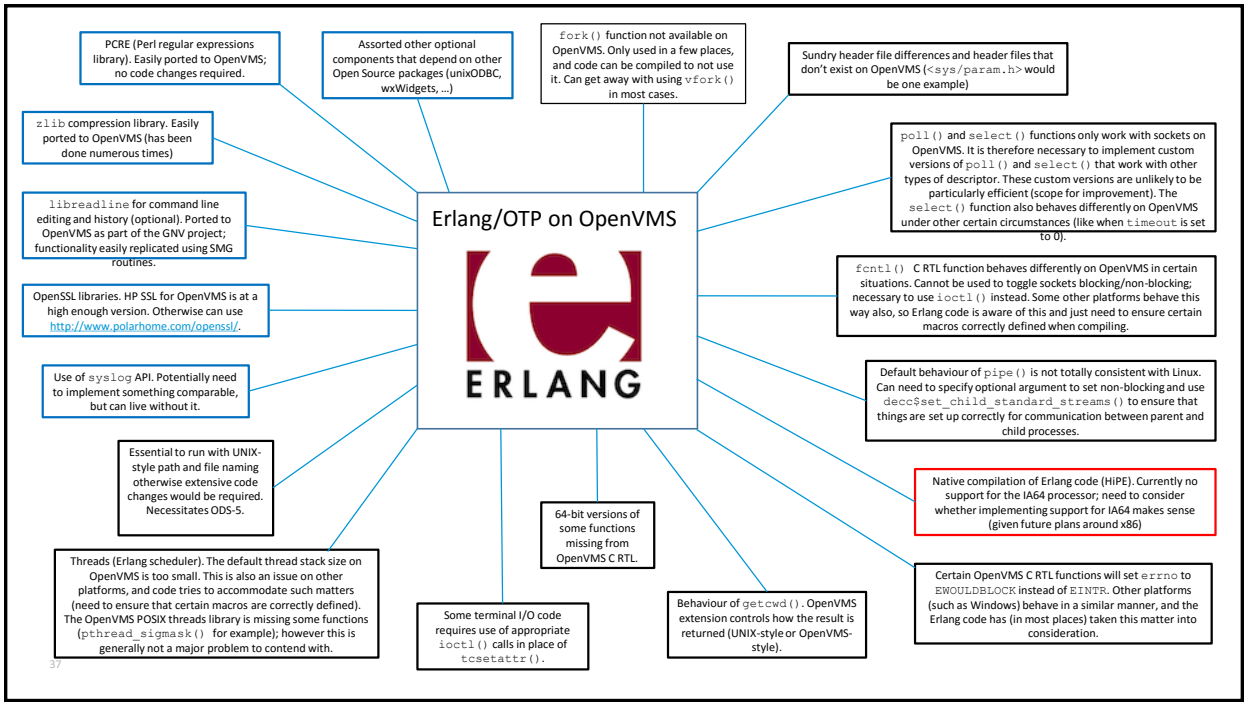
Dependencies and related matters

- Many Open Source products depend on other Open Source products
  - Libraries/API's
  - These are often fundamental building blocks
- C RTL issues
  - Missing functions
  - Differences in header files
  - Behavioural differences
  - We're working on it!
- C/C++ language standards
- …

But don't forget, OpenVMS is OpenVMS; it is not Linux, and there will always be differences that need to be contended with, just as there are when porting Open Source code to Windows, or even between Linux/UNIX variants.

Dependencies and related matters

- Some positioning for the next slide…
  - Illustrates some (but not all) dependencies and issues associated with building Erlang on OpenVMS
    - See my talk "*Porting Erlang to OpenVMS and getting it to do something useful*"
  - Blue boxes are dependencies
  - Red boxes are problematical but optional aspects
  - Black boxes are just the usual sorts of things that have to be dealt with when porting complex Open Source application to OpenVMS



AGENDA

- Programming languages
- Integration technologies
- Databases
- Web
- Libraries/utilities
- Software development tools
- Cloud
- UNIX compatibility
- Analytics
- Add-ons
- Other considerations
- **Summary/conclusions**
- Questions

Summary

- There's a lot to do and a lot that can be done…
  - New stuff as well as incremental updates to existing packages
  - But things are happening and quite a bit has been done!
- Languages, databases, and integration technologies are arguably some of the big-ticket items
  - Both for existing OpenVMS users and to attract new users
  - Development tools can be added to this list (IDE's, source code control, testing, CI, …)
  - Package management is another important consideration
- There are some significant dependencies on Java
  - We're working to get this sorted as soon as possible… work is progressing well; Camiel will update you shortly!
- Need to take a multi-pronged approach, involving some small but strategic items and larger pieces of work being done in parallel
- Identify, prioritise, and systematically address C RTL and compiler-related issues (it's happening)
- Model for community and partner engagement needs to be defined
- …

And now over to Camiel…

AGENDA

- Programming languages
- Integration technologies
- Databases
- Web
- Libraries/utilities
- Software development tools
- Cloud
- UNIX compatibility
- Analytics
- Add-ons
- Other considerations
- Summary/conclusions
- **Questions**


Questions