



Erlang on OpenVMS and in hpcloud – an overview of what HP is wittingly and perhaps unwittingly doing with Erlang

Brett Cameron

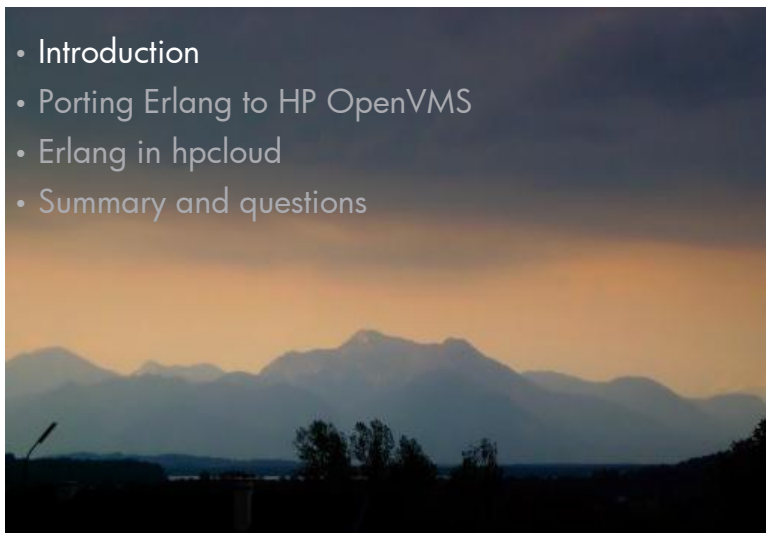
© 2012 Hewlett-Packard Development Company, L.P.
The information contained herein is subject to change without notice



Agenda



- Introduction
- Porting Erlang to HP OpenVMS
- Erlang in hpcloud
- Summary and questions



Abstract



The speaker has made something of a hobby in recent years of porting Open Source software to HP's heritage OpenVMS operating system. One of the most challenging and interesting porting exercises was porting Erlang/OTP. In this talk, the speaker will discuss the motivations behind this work, some of the challenges faced, progress to date, and future plans. In addition, the speaker has recently joined HP's corporate Cloud Services team, and in this capacity he will provide an overview of where and how Erlang is currently being used within the HP Cloud, and will outline potential future opportunities to further leverage Erlang technology in this space.

3

Some stuff about me



Brett Cameron currently works as a senior architect with HP's corporate Cloud Services group, focusing on the design and implementation of message queuing and related integration services for customers and internal use. Brett lives in Christchurch, New Zealand, and has worked in the software industry for some 19 years. In that time he has gained experience in a wide range of technologies, many of which have long since been retired to the software scrapheap of dubious ideas. In recent years Brett has specialized in systems integration, and the design and implementation of large distributed systems for HP's enterprise customers. This work has seen Brett get involved in the research and development of low-latency and highly scalable messaging solutions for the Financial Services sector running on HP platforms, and as a consequence of this work, Brett has been involved in several interesting Open Source projects, and he is responsible (or should that be irresponsible) for porting various Open Source solutions (including Erlang) to HP's "legacy" OpenVMS operating system platform. Brett holds a doctorate in chemical physics from the University of Canterbury, and still maintains close links with the University, working as a part time lecturer in the Computer Science and Electronic and Computer Engineering departments. In his spare time, Brett enjoys listening to music, playing the guitar, and drinking beer (preferably cheap Australian lager).

4

My introduction to Erlang



- We'll talk more about this later, but...
- It basically all started in November 2008 here in San Francisco as a consequence of a discussion with my friend Chris Brown in the BA lounge at SFO
 - Chris was at the time Director of Strategy for HP OpenVMS
 - He now runs his own consulting business and has a very successful cafe in Newcastle UK, <http://www.theurbancoffeehouse.co.uk/>
 - I had “encountered” Erlang several weeks earlier (see subsequent slides) and I was telling Chris about it
 - He basically challenged me to get it working on OpenVMS
 - Without Chris’ “input”, I probably would not have bothered
 - Actually, there are many stories from that particular trip...
 - *Chris has a lot to answer for!*



5

Objectives of this talk



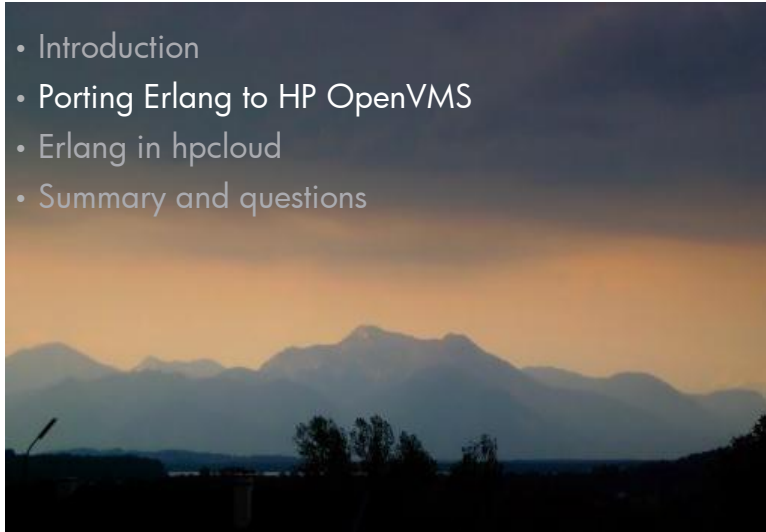
- To describe the work done to port Erlang to HP’s “legacy” OpenVMS operating system
 - Approach
 - Problems
 - Some of the problems encountered are likely to be relevant to ports involving other “exotic” operating systems
 - Results
- Provide an overview of how Erlang is currently being used within HP’s public cloud infrastructure, and future plans
- Provide 45 minutes of hopefully interesting entertainment

6

Agenda



- Introduction
- Porting Erlang to HP OpenVMS
- Erlang in hpcloud
- Summary and questions



7

What is OpenVMS?



- A server operating system that runs on VAX, Alpha, and Intel Itanium processors (VAX no longer supported; limited support for Alpha)
- Initially released by Digital Equipment Corporation (DEC) in 1977
 - DEC purchased by Compaq in 1998
 - Compaq purchased by HP in 2002
 - Latest major version of OpenVMS is 8.4; released in 2010 (Alpha and Itanium)
- Multi-user, multiprocessing, virtual memory-based
- Proprietary...
 - Not at all UNIX-like
 - GNV (GNU's Not VMS) is an attempt to provide a UNIX-like user environment
 - Can be quite an effort to port open source code to OpenVMS
- Designed for use in time sharing, batch processing, real-time, and transaction processing environments



8

What is OpenVMS?



- Offers extremely high levels of availability, scalability, and fault tolerance through clustering
 - First cluster-capable version released 1984
- Extremely secure – considered almost hacker-proof
- Large and very loyal installed base of over 300,000 systems supporting millions of users
- Supports most modern software development technologies
- Used for numerous purposes
 - Mail and other network services
 - Manufacturing and transportation control and monitoring
 - Banking and financial services
 - Most large stock exchanges run OpenVMS
 - Health care
 - Telco billing
 - Government
 - Military
 - Large scale industrial manufacturing
 - A number of chip manufacturers use OpenVMS



9

Motivations



- The corporate-speak:
 - To make available on HP OpenVMS software technologies that help customers extend their computing environments to better meet business needs
 - Demonstrate that OpenVMS can participate in the “modern” world
 - Prove the viability of Open Source solutions on HP OpenVMS and show that OpenVMS is an excellent platform for many Open Source technologies
 - Erlang is a programming language that has many features more commonly associated with an operating system than with a programming language:
 - Concurrent processes
 - Scheduling
 - Memory management
 - Distribution
 - Networking
 - ...

Why port Erlang to OpenVMS?

What better platform than OpenVMS is there to run Erlang on?

10

Motivations



- The reality:
 - Remember Chris from slide #5?
 - Interest in AMQP (Advanced Message Queuing Protocol, see <http://www.amqp.org>)
 - Wanted to get an AMQP implementation on OpenVMS
 - Likely to be of interest to financial services customers
 - Also of interest to customers looking to replace legacy messaging technologies like MQSeries and DEC/BEA/Oracle MessageQ
 - Might get me a few interesting consulting engagements in exotic locations!
 - Initially ported OpenAMQ (<http://www.openamq.org>), but creators (iMatix) stopped developing it
 - Pure (and fairly standard) C code, so not too hard to port
 - Found RabbitMQ (<http://www.rabbitmq.com>)
 - Liked the name, looked like pretty good software, but was written in something called Erlang
 - Cool, a new language to learn...
 - It's a hobby... it keeps me off the streets
 - Hmm, I wonder if I can get Erlang working on OpenVMS...
 - A challenge!

11

Basic approach to porting



- Erlang `beam` and associated components are written in C
 - Appreciable codebase
 - Approximately 400,000 lines of code (*.c, *.h) in total
- When porting Open Source C code to OpenVMS various approaches are possible
 - To some degree a matter of personal taste
 - Often depends on size and complexity of the software being ported
 - Erlang rates as complex
- A good knowledge of OpenVMS, UNIX, C, and the C runtime library is important
 - Understanding platform and C runtime library differences is critical
- Patience is useful
- A few beers often come in very handy
- If at first you don't succeed listen to some loud music or sleep on it for a while and then have another go

For Erlang, I adopted the following approach (one I often use)...

12

Outline of the basic approach



- The basic approach was to first build Erlang on a UNIX system (Solaris as it happens) and to capture a log of the output of the build process
 - To keep things reasonably simple, I did not go for the SMP build
 - I also deliberately excluded a few other optional components from the build that I did not want or need on OpenVMS
 - ... or that I knew would make the port even more complex
 - These other components could be dealt with later

Working on the port whilst
on holiday in Majorca



13

Compiling and linking



- The log of commands for the Solaris build was incrementally converted this into equivalent OpenVMS DCL (OpenVMS command shell) build commands
 - Sorting out initial compiler options
 - For example, `/names=(as_is,shortened)` to preserve case and deal with long (>31 characters) function names
 - Sorting out include-paths can be fun
 - Linker commands
 - Object libraries, shareable images
 - ...
 - Probably possible to use GNV (GNU's Not VMS), but it is (in my opinion) not really good enough yet to be a huge help (but it is getting better), and frankly I'd rather do things in a "pure" OpenVMS environment so as not to mask anything in any way
- At this time I also looked at the "config.h" file created during the UNIX build and modified it as appropriate for OpenVMS
 - Need to have a reasonably good knowledge of the OpenVMS C runtime library when doing this (missing functions, differences in header files, ...)
- Note also that I used simple DCL command procedures to compile the code, as opposed to attempting to use `make`-like utilities available for OpenVMS
 - This is a personal preference, but also in the initial stages of the port I wanted to be sure to recompile everything to make sure that I had not missed anything or messed anything up with a particular change

14

Adjusting for OpenVMS specifics



- Once an initial build procedure was set up, it was a case of systematically fixing up compiler errors, sorting out include paths, eliminating compiler warnings, and so on
 - At this point most of the trivial porting issues come into play
 - Differences in header files between OpenVMS and UNIX/Linux
 - Missing C runtime library functions
 - Choice of compiler flags (qualifiers)
 - ...
 - It is a case of working through these issues and addressing them as appropriate
 - Having done such things quite a few times in the past (and wishing I'd made notes), it did not take too long to at least get most of the code to at least compile and link
 - But this is just the start of the battle!
- As a next pass, I searched through the code for known issues such as trying to use `fcntl()` to set sockets to non-blocking and incrementally addressed these
 - Getting pipes to work correctly and dealing with `fork()/exec()` sequences required a little more thought (and luck)

15

Some final bits and pieces



- Adapting to the OpenVMS file system
 - The last hurdles to overcome in getting Erlang working were file-system related
 - It was reasonably easy to determine where the problems were in the C code
 - But when problems were at the Erlang level, they took me a little more time to resolve (as at this time I was not particularly familiar with Erlang)
- Making OpenVMS “look” like UNIX
 - In order to avoid having to make excessive changes to any of the standard libraries and OTP code, I “made” OpenVMS look like a variant of UNIX
 - This means that wherever the Erlang checks the operating system type it thinks it's a variant of UNIX
 - And it doesn't much care (except in a small number of cases) what the specific variant is

```
$ erl
Eshell V5.7 (abort with ^Z)
1> os:type().
{unix, openvms}
2>
```

16

Summary of main technical issues



- There is no UNIX-like `fork()` function on OpenVMS
 - Luckily only a small number of `fork()` calls in the Erlang code
 - Basically just `fork()/exec()` sequences that start other processes like `inet_gethost.exe` and set up a pipe for communication with parent
 - Could work around on OpenVMS using `vfork()/exec()` sequences
- `fcntl()/ioctl()`
 - You can't use `fcntl()` on OpenVMS to toggle sockets blocking/non-blocking
 - Simple case of replacing any such calls with appropriate `ioctl()` calls
- `poll()/select()`
 - Only work with sockets on OpenVMS so had to implement special versions to handle other types of file descriptor
 - These wrappers probably have performance implications...
- File system differences
 - The OpenVMS file system is quite different to UNIX (more like Windows... common lineage)
 - OpenVMS On Disk Structure (ODS) levels
 - ODS2 – no good for Erlang (short file names, no fancy characters, one ".", uppercase)
 - ODS5 – allows UNIX-like file names, special characters, mixed-case, ...
 - To avoid extensive modification of the Erlang file I/O module, went with ODS5 as prerequisite and translate to/from UNIX path/filename format when necessary

17

Testing the port



- Most of the testing of the port was performed using RabbitMQ
 - Getting RabbitMQ going was after all the primary objective!
 - Significant piece of code
 - Around 35,000 lines of quite complex Erlang code (*.erl, *.hrl)
 - Uses numerous bits of OTP
 - One minor RabbitMQ code change required to correctly determine the number of file descriptors the process could use
 - No such thing as `ulimit` on OpenVMS
- Performed additional testing with Yaws, CouchDB, and assorted simple Erlang test programs
- ... but let's look at RabbitMQ
 - Getting it going
 - Results
 - Problems

18

But first, what is RabbitMQ?



- A powerful Open Source message broker (message-oriented middleware)
 - The leading (and arguably the most popular) implementation of the Advanced Message Queuing Protocol (AMQP, <http://www.amqp.org>)
 - Provides a robust and flexible messaging platform designed to interoperate with other messaging systems
 - See <http://www.rabbitmq.com>
- RabbitMQ essentially comprises the following components:
 - The RabbitMQ broker
 - Gateways for HTTP, XMPP, STOMP, and other protocols
 - AMQP client libraries for Java, .NET, and C/C++
 - AMQP clients for other languages are available from other vendors and/or the Open Source community
 - Assorted useful plug-ins
- The broker is written entirely in Erlang and makes extensive use of the OTP framework for networking, clustering, failover, ...
 - The high availability and functional characteristics of Erlang/OTP make it ideal for a product such as RabbitMQ (an inspired move)
- Rabbit Technologies Ltd. was acquired in April 2010 by VMware
 - ... and have incorporated the software into their Cloud stack (possibly more on that later)...



19

Starting RabbitMQ



Here's part of the command procedure (shell script) to start RabbitMQ on OpenVMS (adapted from the corresponding UNIX script):

```
$ RABBITMQ_MNESIA_DIR = "/rabbitmq$root/mnesia"
$ RABBITMQ_LOGS = "/rabbitmq$root/log"
$ RABBITMQ_SASL_LOGS = "/rabbitmq$root/log"
$ RABBITMQ_NODE_NAME = "rabbit"
$ RABBITMQ_NODE_PORT = 5672
$ RABBITMQ_NODE_IP_ADDRESS = "16.156.32.108"
$
$ define decc$fd_locking 1
$
$ erl :=- $erlang$root:[bin]erlexec.exe
$ erl -
-pa "/rabbitmq$root/ebin" -
-pa "/erlang$root/lib/mnesia/ebin" -
-pa "/erlang$root/lib/os_mon/ebin" -
-pa "/erlang$root/lib/sasl/ebin" -
-pa "/erlang$root/lib/kernel/ebin" -
-pa "/erlang$root/lib/ssl/ebin" -
-pa "/erlang$root/lib/stdlib/ebin" -
-noinput -
-emu_args -
-boot "start_sasl" -
-s "rabbit" -
-config "/rabbitmq$root/sbin/rabbitmq.config" -
-name "'RABBITMQ_NODE_NAME'" -
-pp "pp" -
+A30 -
-rabbit "tcp_listeners" [{"RABBITMQ_NODE_IP_ADDRESS",RABBITMQ_NODE_PORT}] -
-kernel "error_logger" "{file,RABBITMQ_LOGS/rabbit.log}" -
-sasl "error_logger" "error" -
-sasl "sasl_error_logger" "{file,RABBITMQ_SASL_LOGS/sasl.log}" -
-mnesia "dir" "'RABBITMQ_MNESIA_DIR'"
$
$
$ exit
```

Assorted symbol definitions (basically the same as environment variables)

You don't really want to know what this is for

Normally you run something on OpenVMS by typing "run" followed by the name of the program. To make things work more like UNIX, we need to define a foreign command.

And this is our command line to start the RabbitMQ broker on OpenVMS. Not all of the double quotes are strictly necessary, but we do need to be careful to preserve case.

20

Running processes (1)



After successfully starting RabbitMQ, this is what we see if we list the running processes on the system:

```
$ show system
OpenVMS V8.3-1H1 on node CCIN02 29-FEB-2012 13:20:01.44 Uptime 214 21:54:49
Pid Process Name State Pri I/O CPU Page files Pages
00000401 SWAPPER HIB 16 0 0 00:09:21.55 0 4
00000404 USBSUCM_SERVER HIB 6 331 0 00:00:00.09 253 362
00000405 LANACP HIB 14 86 0 00:00:00.00 168 214
00000407 FASTPATH_SERVER HIB 10 8 0 00:00:00.00 108 134
00000408 IFCACP HIB 10 8 0 00:00:00.00 78 109
00000409 ERREMT HIB 8 1132537 0 00:00:39.30 167 203
0000040B OPCOM HIB 8 26408 0 00:00:01.69 5305 93
0000040C AUDIT_SERVER HIB 10 403 0 00:00:00.07 170 215
0000040D JOB_CONTROL HIB 9 4309528 0 00:03:08.45 124 189
00000411 QUEUE_MANAGER HIB 9 17591 0 00:00:02.72 201 275
00000412 SECURITY_SERVER HIB 10 74163 0 00:00:00.31 488 643
00000413 ACME_SERVER HIB 10 76 0 00:00:01.56 411 550 M
00000415 DNS$ADVER LEF 5 3824980 0 00:02:48.65 842 954
00000416 LESSACP_V30 HIB 8 132 0 00:00:00.01 123 146
00000417 NETSACP HIB 6 978 0 00:00:00.08 235 275
00000418 REMACP HIB 10 40 0 00:00:00.00 77 79
00000419 NETSEVD HIB 6 44 0 00:00:03.02 262 517
0000041A TP_SERVER HIB 10 13 0 00:00:00.00 151 182
00000421 TCP$P$INETACP HIB 10 29160 0 00:05:04.19 460 439
00000422 TCP$P$FTP_1 LEF 10 32560 0 00:00:01.00 1045 743 N
00000423 SYMBIONT_1 HIB 6 104 0 00:00:00.10 763 138
00000424 SYMBIONT_2 HIB 6 43938 0 00:00:00.17 1025 209
00000428 RMS_MONITOR72 LEF 15 32147 0 00:00:03.13 16704 212
0000042A ACMS_SWL HIB 9 184 0 00:00:00.03 123 164
0000042B WISMANAGER HIB 8 22602508 0 00:35:28.14 20751 7567 M
00000434 SMHANDLER HIB 8 52 0 00:00:00.00 247 246
00018086 EMPD$SERVER LEF 6 1262813 0 00:00:00.33 1125 375
001BE53C MEMCACHE$SERVER LEF 5 6735644 0 00:00:18.55 8496 6704
001C314F CAMERON CUR 0 4 1861 0 00:00:00.23 1638 310
00001978 RabbitMQ HIB 6 2148 0 00:00:00.13 1039 381
000D2177 CAMERON_41642 HIB 4 98136883 0 00:23:58.72 3816 1649 MS
000CB17A CAMERON_53419 HIB 6 850 0 00:00:00.03 300 287 S
000D197B CAMERON_49049 LEF 6 515 0 00:00:00.02 440 421 S
000059DF APACHE$SWS LEF 6 9968 0 00:00:00.93 990 1198
000059E0 APACHE$SWS0000 LEF 6 3707 0 00:00:00.44 989 1196
000059E1 APACHE$SWS0001 LEF 6 9522 0 00:00:01.19 979 1206
000059E2 APACHE$SWS0002 LEF 6 5218 0 00:00:00.57 1021 1095
000059E3 APACHE$SWS0003 LEF 6 10279 0 00:00:01.26 1014 1117
000059E4 APACHE$SWS0004 LEF 6 9715 0 00:00:01.23 972 1199
```

The empd processes had been started previously

What are these four processes? Let's have a closer look...

Running processes (2)



```
$ show proc/cont/id=000D8175
Process RabbitMQ 13:24:52
State HIB Working set 381
Cur/base priority 9/4 Virtual pages 10888
Current PC FFFFFFFF.805FC8B0 CPU time 0 00:00:00.13
Current PSI 00000000 Direct I/O 490
Current user SP 7AC29280 Buffered I/O 1658
PID 000D8175 Page faults 1039
UIC [USER,CAMERON] Event flags C0000001
E0000000
```

On UNIX this process would be "replaced" by beam.

```
$ show proc/cont/id=000D2177
Process CAMERON_41642 13:25:24
State HIB Working set 1650
Cur/base priority 5/4 Virtual pages 13315
Current PC FFFFFFFF.805FC8B0 CPU time 0 00:23:59.09
Current PSI 00000000 Direct I/O 74261234
Current user SP 0041FA30 Buffered I/O 23899700
PID 000D2177 Page faults 3817
UIC [USER,CAMERON] Event flags C0000001
E0000000
```

Beam ends up running as a sub-process of erlexec.exe (arguably not ideal, but okay for now)

```
$!1SDKA80:[USER,CAMERON.eal.] [bin]BEAM.EXE;300
```

Note the high version numbers. It took a few goes to get things right.

Running processes (3)



```
$ show proc/cont/id=000CB17A
Process CAMERON_53419          13:26:08
State                          HIB                Working set      287
Cur/base priority             9/4                Virtual pages    10996
Current PC FFFFFFFF.805FCEB0    CPU time          0 00:00:00.03
Current PSL 00000000            Direct I/O        20
Current user SP 7ACCB520        Buffered I/O      830
PID 000CB17A                   Page faults       300
UIC (USER,CAMERON)             Event flags       C0000001
                                40000000
```

Started by beam to manage start-up of child processes. Communicates with beam via a pipe.

```
$!1$DKA800:[USER.CAMERON.erl.] [bin]child_setup.exe:243
```

```
$ show proc/cont/id=000D197B
Process CAMERON_49049          13:26:39
State                          LEF                Working set      421
Cur/base priority             9/4                Virtual pages    11165
Current PC FFFFFFFF.8062A710    CPU time          0 00:00:00.02
Current PSL 00000000            Direct I/O        55
Current user SP 7ACCBAA0        Buffered I/O      460
PID 000D197B                   Page faults       440
UIC (USER,CAMERON)             Event flags       C0000001
                                80000000
```

Started by beam to do name service lookups. This functionality could arguably be incorporated into the beam process without too much bother.

```
$!1$DKA800:[USER.CAMERON.erl.] [bin]inet_gethost.EXE:89
```

23

Some results



- Performance on OpenVMS is (currently) not great but is acceptable
 - Message rates of around 6000 messages per second for a single publisher and single consumer on a low-end OpenVMS Integrity server (256-byte messages)
 - Could expect to see rates of above 20,000 messages per second on a comparable Linux system
 - Multiple reasons for this disparity
 - Performance of `poll()`/`select()` a major contributor
 - Probably need to do a bit of tuning
- SSL works perfectly (after I installed the correct version of OpenSSL)
- Clustering – no problem!
- Mnesia seems to work fine (after addressing some OpenVMS/UNIX file system differences)
- Memory
 - Currently constrained to beam using < 1GB of memory (default OpenVMS memory model... long story)
 - Need to build beam using 64-bit pointers to remove this limitation
- Very stable... so long as memory usage is constrained

24

Where to from here?



- Interest to date:
 - There are a number of customers wanting to use RabbitMQ to replace existing messaging queuing technologies
 - Two chip manufacturers and several other manufacturing organisations
 - A European rail company
 - Trying to develop interest in several key HP accounts
 - A certain security agency has exhibited strong interest in using Erlang on OpenVMS
 - Not quite sure why – they wouldn't say
 - Several OpenVMS hobbyists

25

Where to from here?



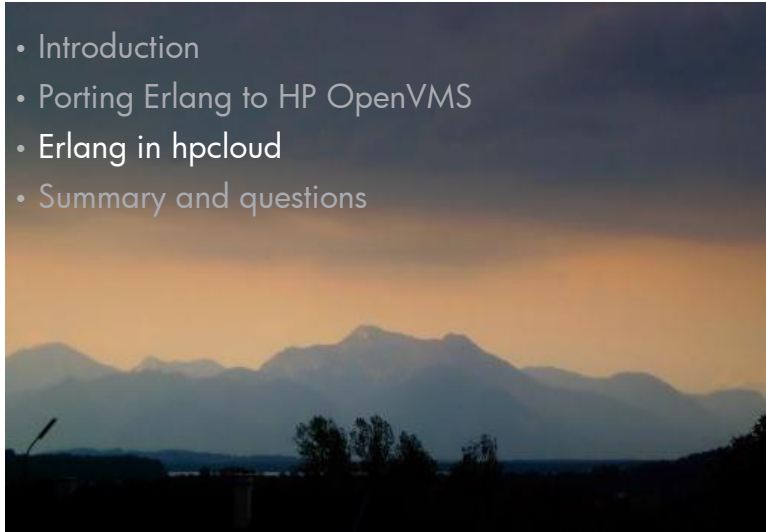
- Future plans:
 - Build a version using 64-bit pointers
 - Initial work indicates that this will be possible (a bit of effort required)
 - Will allow `beam` to use more than 1GB memory
 - Sort out remaining I/O issues
 - Do the job properly/better with Erlang 15B
 - Longer term goal... will probably stick with 13B for a little while longer
 - Maybe try to get HP OpenVMS engineering involved
 - Generate more interest amongst the OpenVMS community
 - Get more Erlang applications working on OpenVMS
 - My friend, colleague, and partner in crime, John Apps, and I probably need to launch one or two major promotional campaigns!
 - See <http://erlangonopenvms.blogspot.com> for any new developments

26

Agenda



- Introduction
- Porting Erlang to HP OpenVMS
- Erlang in hpcloud
- Summary and questions



27

HP Cloud Services



- See <http://www.hpcloud.com>
- Technology is now in private beta
 - Public beta not too far away!
 - Will provide public cloud infrastructure that is business grade, open source based, and developer-focused
 - Initial Infrastructure as a Service (IaaS) offerings include HP Cloud Compute and HP Cloud Object Storage
 - Built on HP's hardware and software
 - Underpinned by OpenStack (<http://www.openstack.org/>)
- Rapidly growing HP business unit... and making very rapid progress
 - Teams in various locations
 - UK, Ireland
 - US (Seattle, Cupertino, Colorado, and elsewhere)
 - Australia
 - New Zealand (just me)

28

H.P. Attempts to Take On Amazon's Cloud Service New York Times (blog), 03/9, Quentin Hardy



Within two months, Hewlett-Packard will offer a large and powerful cloud computing service similar to Amazon Web Services, but with more business-oriented features, according to the head of the project.

"We're not just building a cloud for infrastructure," said Zorawar "Biri" Singh, senior vice president and general manager of H.P.'s cloud services. "Amazon has the lead there. We have to build a platform layer, with a lot of third-party services." Among the first software applications available as part of the Hewlett-Packard cloud, he said, will be both structured and unstructured databases, and data analytics as a service.

"We won't pull (Amazon's) customers out by the horns," he said, "but we already have customers in beta who see us as a great alternative." He did not say how much the computing services would cost, but said "we are not coming at this at '8 cents a virtual computing hour, going to 5 cents," which even at the initial price could undercut Amazon. While Amazon tends largely to have a self-service model, Hewlett-Packard's cloud will also offer more personalized sales and service, he said.

H.P. also plans to offer a number of tools for developers to use popular online software languages, like Ruby, Java, and PHP, as well as ways for customers to provision and manage their workloads remotely. The service will also include an online store where people can offer or rent software for use in the Hewlett-Packard public cloud. Mr. Singh said the company would take precautions to ensure the quality and security of these software offerings from third parties by providing services like user authentication and billing.

Hewlett-Packard's alternative to A.W.S. has been underway for over a year, and is likely to be the most ambitious project yet under Meg Whitman, who became chief executive of the Palo Alto, Calif., technology company last September. While seemingly focused on Amazon, the company is also looking at the project as a new way to compete with its traditional rivals.

"We want to make it hard for an I.B.M. or an Oracle or anyone to come in," he said. By offering a lot of tools for developers and business-ready software to corporations, H.P. could find ways to undercut existing enterprise offerings, while surviving against Amazon, a notoriously low-margin competitor.

Though the data centers presently supporting H.P.'s cloud are located only on the East and West Coasts of the United States, H.P. plans to scale the program by installing small data centers across the globe. This small and dispersed approach is a break from the goliath data centers run by cloud companies like Amazon and Google. The project will run almost entirely on Hewlett-Packard technology.

As ambitious as the program sounds, Mr. Singh said the revenue from the public cloud business will have little initial impact on H.P.'s annual revenue, which are in excess of \$100 billion. His project will be judged, he said, as much on how well it helps other parts of Hewlett-Packard's business as it is on its own revenue. "We do everything from laptops to cloud computing," he said. "This will leverage our whole sales channel."

The analytics Hewlett-Packard will offer will be derived from its earlier purchases of Vertica and Autonomy. H.P. has previously talked about offering these pattern-finding capabilities in its computer servers. In addition, it hopes to use the public cloud, which like other clouds will interoperate with the computing resources inside companies, as a way to showcase its latest servers, which Mr. Singh said outperformed cheaper commercial offerings in areas like power usage and computing capabilities when H.P. was designing the system.

29

OpenStack



- Open source software for building private and public clouds
- The project was created with the goal of being the software choice for building cloud infrastructures
 - Began through joint work between NASA and Rackspace Hosting
 - Decided release as open source their internal cloud compute and cloud storage solutions
 - In little over a year it has become arguably the most talked about Open Source project
 - Currently some 155 companies involved
- Initial focus is on Infrastructure as a Service (IaaS)
 - Currently three major components:
 - OpenStack Compute (a.k.a. "Nova") – software to orchestrate, manage, and offer virtual machines
 - OpenStack Object Store (a.k.a. "Swift") – software for the redundant storage of static objects
 - OpenStack Image Service (a.k.a. "Glance") – provides query and storage services for virtual disk images
- Provides support for numerous hypervisors



30

So where does Erlang fit in?



- While we generally don't even think about it, Erlang is absolutely critical to the operation of HP Cloud
 - And indeed to the operation of any other cloud environment that's using OpenStack

Erlang runs the central nervous system for OpenStack Nova

Responsible for managing 10's of thousands of inter-process communication links across hundreds (eventually to be thousands) of servers

(AMQP-based communication between cooperating processes using our good friend RabbitMQ)

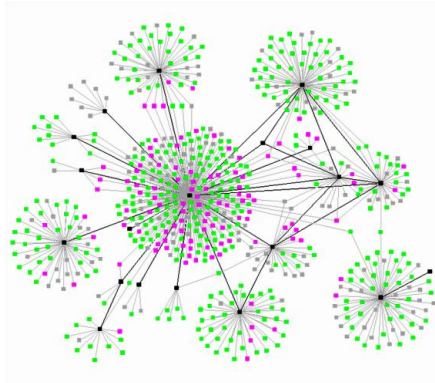


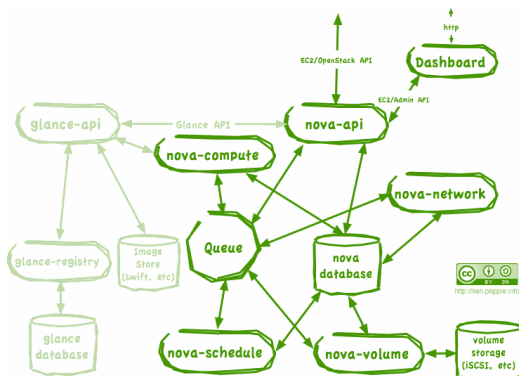
Image adapted with permission from a RabbitMQ presentation given by Tom McCuch (tmccuch@vmware.com)

31

And it's spreading



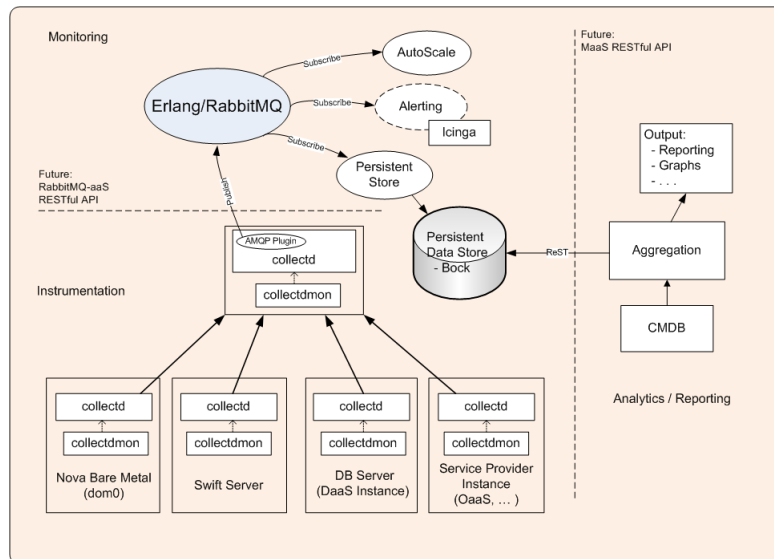
- Our architects seem to have embraced RabbitMQ and are doing (or are looking to do) all sorts of interesting things with it...
 - Message Queuing as a Service
 - Monitoring as a Service
 - This is starting to look really nice...
 - Bock (block storage)
 - Already up and running
 - Some seriously good stuff here
 - ... all underpinned by Erlang



- As a consequence of these services Erlang will become even more important to HP Cloud
 - ... which seems perfectly reasonable to me!

32

Monitoring as a Service

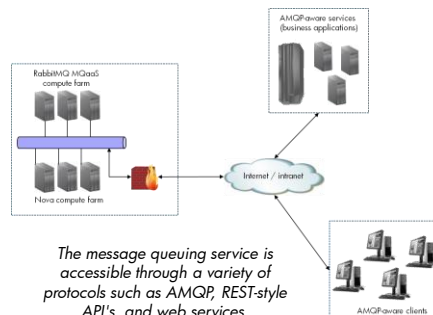


33

Message Queuing as a Service



- Message-oriented middleware deployed in a compute cloud using the software as a service model
- Service subscribers access queues and or topics to exchange data using point-to-point or publish and subscribe patterns
- Aims to eliminate traditional overheads associated with operating in-house messaging infrastructures
 - Unused capacity
 - Hardware
 - Licenses
 - Infrastructure maintenance and support
 - Idle time waiting for resource provisioning
 - Need to isolate messaging resources
- Internet-scale messaging platform



The message queuing service is accessible through a variety of protocols such as AMQP, REST-style API's, and web services.

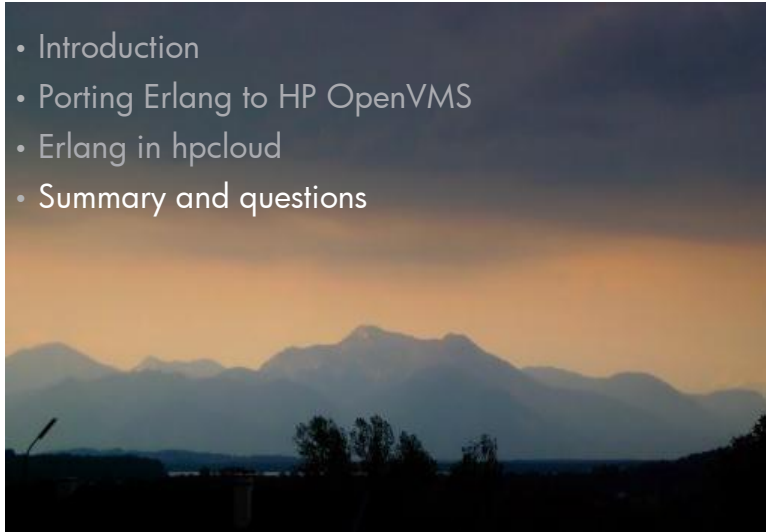
VMware recently launched a RabbitMQ service for Cloud Foundry. We're looking at doing something similar...

34

Agenda



- Introduction
- Porting Erlang to HP OpenVMS
- Erlang in hpcloud
- Summary and questions



35

Summary



- Successfully ported Erlang to HP OpenVMS
 - Still some issues to sort out, but generally things are working okay
 - Discussed some of the issues faced
 - Many of these issues and considerations would be applicable to other such porting exercises involving proprietary operating systems
- Erlang is managing to find its way into some interesting places
 - Ideally suited to cloud environments
 - Hopefully others will realise this and the Erlang cloud footprint will continue to grow!

36

Questions?

