Puissance de Clojure

Tom Marble

June 26, 2012

Contents

1 Overview

Puissance de Clojure
file:~/src/irill/Clojure-glyph.svg

1.1 Overview: Topics to cover

Press **f9** to see the list of topics
Merci: IRILL, Zacchiroli, Ledru

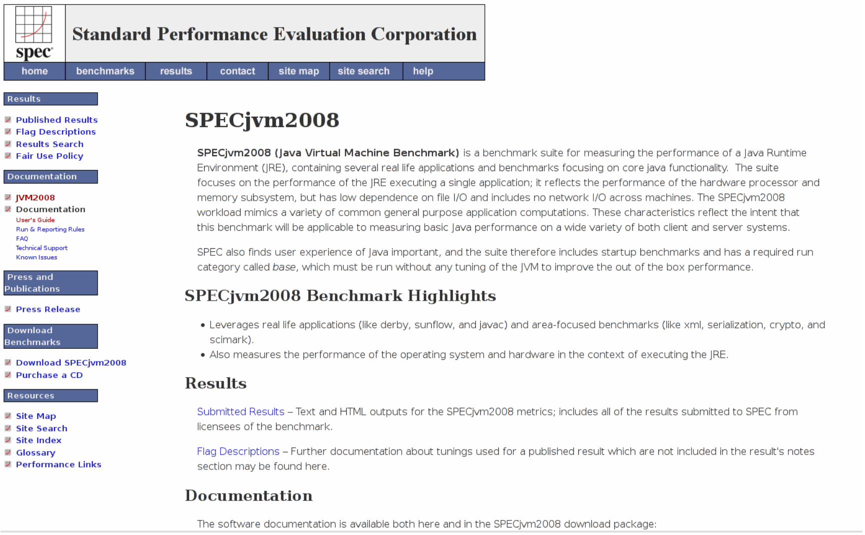
2 Background

2.1 About Tom

tmarble

2.1.1 Sun: technical presales during the dot.com era

2.1.2 Sun: Java Performance



SPECjvm2008

SPECjvm2008 (Java Virtual Machine Benchmark) is a benchmark suite for measuring the performance of a Java Runtime Environment (JRE), containing several real life applications and benchmarks focusing on core java functionality. The suite focuses on the performance of the JRE executing a single application; it reflects the performance of the hardware processor and memory subsystem, but has low dependence on file I/O and includes no network I/O across machines. The SPECjvm2008 workload mimics a variety of common general purpose application computations. These characteristics reflect the intent that this benchmark will be applicable to measuring basic Java performance on a wide variety of both client and server systems.

SPEC also finds user experience of Java important, and the suite therefore includes startup benchmarks and has a required run category called *base*, which must be run without any tuning of the JVM to improve the out of the box performance.

SPECjvm2008 Benchmark Highlights

- Leverages real life applications (like derby, sunflow, and javac) and area-focused benchmarks (like xml, serialization, crypto, and scimark).
- Also measures the performance of the operating system and hardware in the context of executing the JRE.

Results

Submitted Results – Text and HTML outputs for the SPECjvm2008 metrics; includes all of the results submitted to SPEC from licensees of the benchmark.

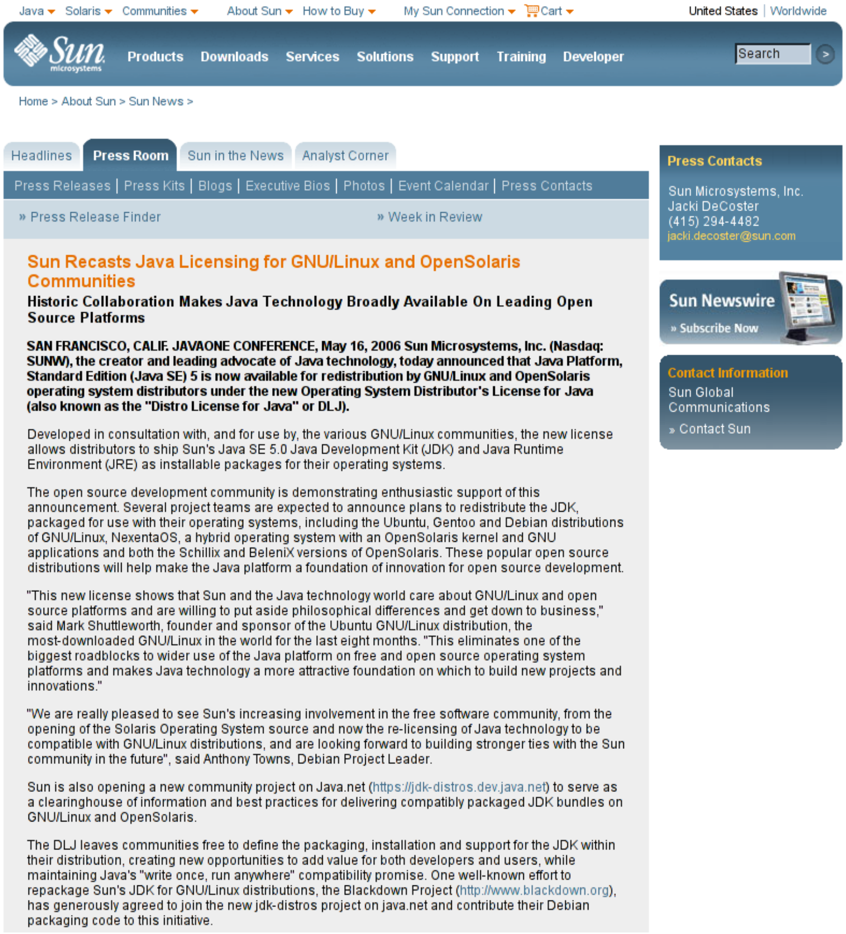
Flag Descriptions – Further documentation about tunings used for a published result which are not included in the result's notes section may be found here.

Documentation

The software documentation is available both here and in the SPECjvm2008 download package:

2.1.3 Sun: DLJ

Early 2006: DLJ with Debian and Canonical (Ubuntu)



Press Contacts

Sun Microsystems, Inc.
Jacki DeCoster
(415) 294-4482
jacki.decoster@sun.com

Sun Recasts Java Licensing for GNU/Linux and OpenSolaris Communities

Historic Collaboration Makes Java Technology Broadly Available On Leading Open Source Platforms

SAN FRANCISCO, CALIF. JAVAONE CONFERENCE, May 16, 2006 Sun Microsystems, Inc. (Nasdaq: SUNW), the creator and leading advocate of Java technology, today announced that Java Platform, Standard Edition (Java SE) 5 is now available for redistribution by GNU/Linux and OpenSolaris operating system distributors under the new Operating System Distributor's License for Java (also known as the "Distro License for Java" or DLJ).

Developed in consultation with, and for use by, the various GNU/Linux communities, the new license allows distributors to ship Sun's Java SE 5.0 Java Development Kit (JDK) and Java Runtime Environment (JRE) as installable packages for their operating systems.

The open source development community is demonstrating enthusiastic support of this announcement. Several project teams are expected to announce plans to redistribute the JDK, packaged for use with their operating systems, including the Ubuntu, Gentoo and Debian distributions of GNU/Linux, NexentaOS, a hybrid operating system with an OpenSolaris kernel and GNU applications and both the Schillix and BeleniX versions of OpenSolaris. These popular open source distributions will help make the Java platform a foundation of innovation for open source development.

"This new license shows that Sun and the Java technology world care about GNU/Linux and open source platforms and are willing to put aside philosophical differences and get down to business," said Mark Shuttleworth, founder and sponsor of the Ubuntu GNU/Linux distribution, the most-downloaded GNU/Linux in the world for the last eight months. "This eliminates one of the biggest roadblocks to wider use of the Java platform on free and open source operating system platforms and makes Java technology a more attractive foundation on which to build new projects and innovations."

"We are really pleased to see Sun's increasing involvement in the free software community, from the opening of the Solaris Operating System source and now the re-licensing of Java technology to be compatible with GNU/Linux distributions, and are looking forward to building stronger ties with the Sun community in the future", said Anthony Towns, Debian Project Leader.

Sun is also opening a new community project on Java.net (https://jdk-distros.dev.java.net) to serve as a clearinghouse of information and best practices for delivering compatibly packaged JDK bundles on GNU/Linux and OpenSolaris.

The DLJ leaves communities free to define the packaging, installation and support for the JDK within their distribution, creating new opportunities to add value for both developers and users, while maintaining Java's "write once, run anywhere" compatibility promise. One well-known effort to repackage Sun's JDK for GNU/Linux distributions, the Blackdown Project (http://www.blackdown.org), has generously agreed to join the new jdk-distros project on java.net and contribute their Debian packaging code to this initiative.

2.1.4 Sun: OpenJDK

JavaOne 2006: Rich Green announces that Sun will open source Java Core Strategy Team

- How organize community governance
- Copyright, Patent and Trademark licensing
- Infrastructure tools
- Pick license

First OpenJDK Ambassador (I went to a lot of conferences)

- FOSDEM

- FISL
- OSCON

ApacheCon 2006: Sun unBOF/Party

Copyright 2006 Ted Leung: https://secure.flickr.com/photos/twleung/268116213/

2.1.5 Consulting

Cybersecurity
Probabilistic Model Verification (Electrical Engineering)
Software for Smart Grid + Renewable Energy
Clojure

2.1.6 Tom and Debian

Using Linux since 1996
Using Debian since 2003
Helped Debian Java Packaging Team since DLJ in 2006
Working with Debian and Oracle on meshing Jigsaw with **apt**

2.1.7 FOSDEM

Java track

- Oracle
- Distros
- Developers
- Users

Legal Issues Track

- Organized by Karen Sandler, Bradley Kuhn, Richard Fontana and myself
- Check out the Wiki http://info9.net/wiki/fosdem/LegalIssuesDevRoom/
- Check out the oggcast http://faif.us/

2.2 My current work in Clojure

The client

- Developing a multiplayer game server hosting service

- Comprised of very young developers
- Is in stealth mode (sorry!)

I have been given the authority to

- Make significant choices about architecture
- Green light to open source generic bits

(this is why i like consulting :)

3 Pourquoi Common Lisp?

John McCarthy was old school: developed LISP in 1954

3.1 Avantages de Common Lisp

3.1.1 homoiconic

code is data

List

(def mylist (list 1 2 3))

Function

(def myadd (fn [a b] (+ a b)))

A lisp is defined in terms of the evaluation of data structures and not in terms of the syntax of files.

3.1.2 macros: code transformations at compile time

Macros offer hooks for syntactic abstraction and there is very little syntax.

```
(defmacro and ([] true) ([x] x) ([x & rest] ‘(let [and# ˜x] (if and# (and ˜@rest) and#))))
```

Allows code transformation **before** the reader does evaluation. In Clojure **defn** is a macro that makes defining functions a little simpler.

Code walkers are easy to write.

3.1.3 Great for Domain Specific Languages

LISP is the language of choice when writing Domain Specific Languages (DSL's).

Example from ILC ‘09 at MIT

- Alex Fukunaga (Tokyo University) spoke on The Satisfyability Problem
- A DSL for SAT algorithms
- Used a biological evolution inspired algorithm

3.1.4 REPL

The Read Eval Print Loop

Interactive code development

Instead of just dump a stack trace and die on an error... you can edit data and functions (they look the same) and continue your program!

3.1.5 Lisp successes

Artificial Intelligence

Travel Planning

Google's $700 M acquisition of ITA

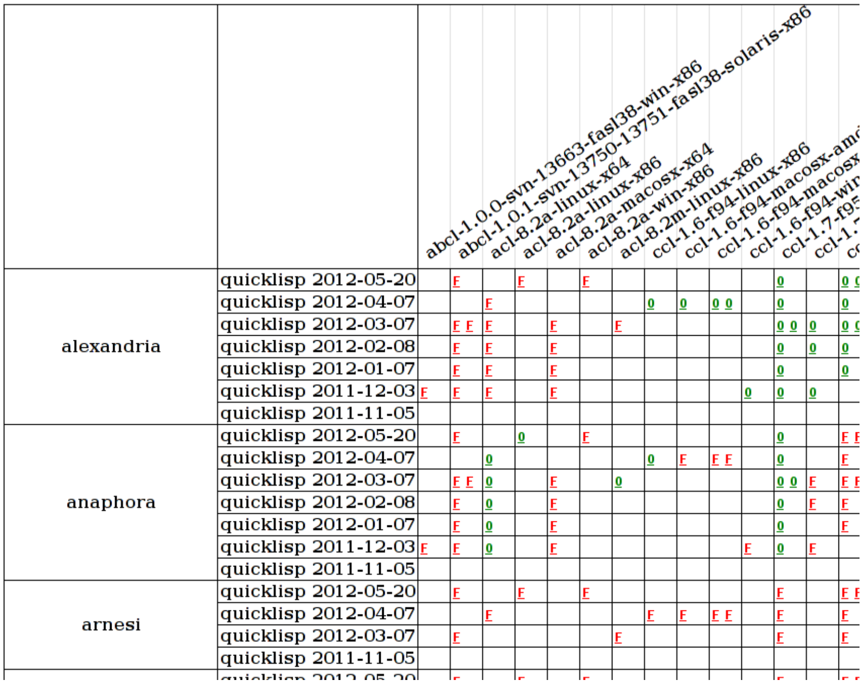
Scientific Computing Lisp

SciCL augments Common Lisp with an extensive library of aggregate-wise (“AG-wise”) operations on arrays, providing the essential functionality of languages such as APL, Fortran 90, IDL and Matlab.

http://www.siginf.com/

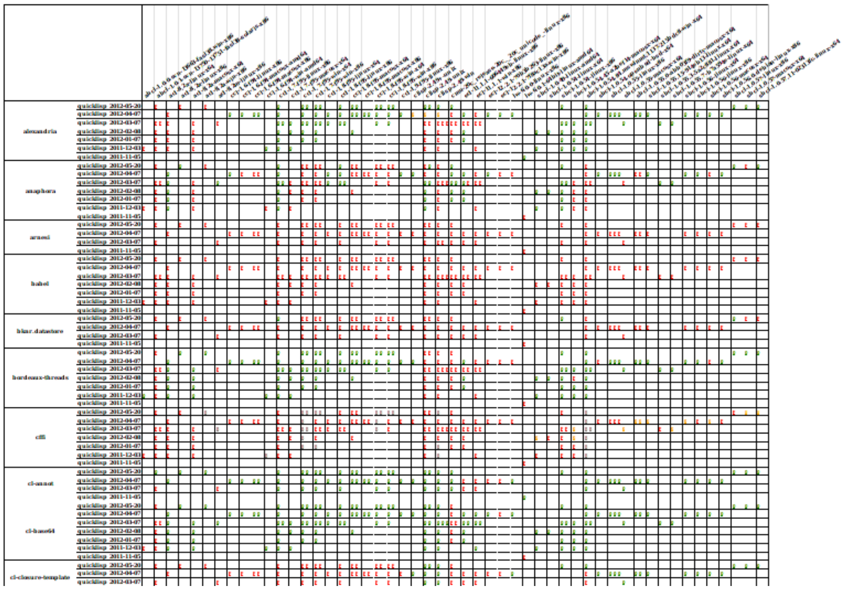
3.2 Inconvenients de Common Lisp

3.2.1 There is a standard, but no compatibility test kit (as for Java)



		abcl-1.0.0-svn-13663-fasl38-win-x86	abcl-1.0.1-svn-13750-13751-fasl38-solaris-x86	acl-8.2a-linux-x64	acl-8.2a-linux-x86	acl-8.2a-macosx-x64	acl-8.2a-win-x86	acl-8.2m-linux-x86	ccl-1.6-f94-linux-x86	ccl-1.6-f94-macosx-am	ccl-1.6-f94-macosx	ccl-1.6-f94-wir	ccl-1.7-f95	ccl-1.	cc
alexandria	quicklisp 2012-05-20		F		F		F						0		0 0
	quicklisp 2012-04-07			F					0	0	0 0		0		0
	quicklisp 2012-03-07		F F	F		F		F					0 0	0	0 0
	quicklisp 2012-02-08		F	F		F							0	0	0
	quicklisp 2012-01-07		F	F		F							0		0
	quicklisp 2011-12-03	F	F	F		F						0	0	0	
	quicklisp 2011-11-05														
anaphora	quicklisp 2012-05-20		F		0		F						0		F F
	quicklisp 2012-04-07			0					0	F	F F		0		F
	quicklisp 2012-03-07		F F	0		F		0					0 0	F	F F
	quicklisp 2012-02-08		F	0		F							0	F	F
	quicklisp 2012-01-07		F	0		F							0		F
	quicklisp 2011-12-03	F	F	0		F						F	0	F	
	quicklisp 2011-11-05														
arnesi	quicklisp 2012-05-20		F		F		F						F		F F
	quicklisp 2012-04-07			F					F	F	F F		F		F
	quicklisp 2012-03-07		F					F					F		F
	quicklisp 2011-11-05														
	quicklisp 2012-05-20		F		F		F						F		F F

3.2.2 Having a standard is no guarentee of compatibility


alexandria
anaphora
arnesi
babel
bknr.datastore
bordeaux-threads
cffi
cl-annot
cl-base64
cl-closure-template
quicklisp 2012-05-20
quicklisp 2012-04-07
quicklisp 2012-03-07
quicklisp 2012-02-08
quicklisp 2012-01-07
quicklisp 2011-12-03
quicklisp 2011-11-05

3.2.3 The "library" problem

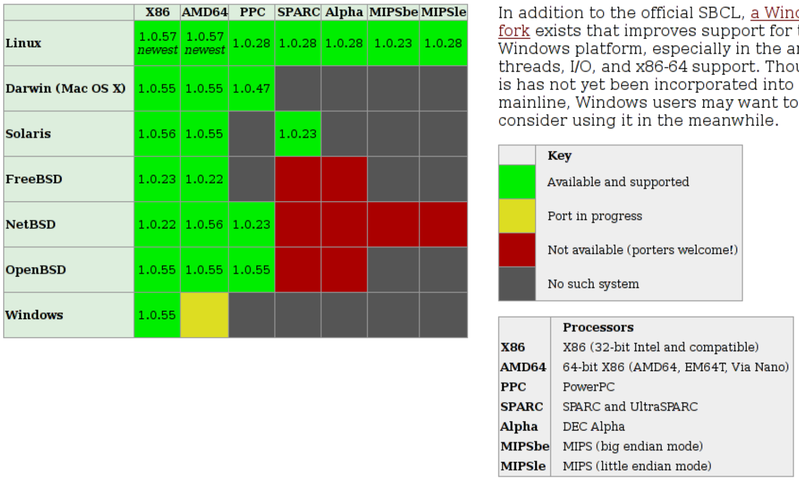
There isn't a consistent disclipline within the LISP community about how best to package and distribute libraries.

There is ASDF which is a low-level approach and QuickLisp (which uses ASDF) to create a CPAN-like high-level approach. But not all the libraries are Quicklisp enabled (nor work on all Lisp implementations).

Writing Lisp code means making a committment to a set of implementations and a packaging approach – and porting missing bits.

3.2.4 The porting problem

The current Lisp implementions often use some assembly langauge at a low level... which limits their portability. For example here is the picture for SBCL:



	X86	AMD64	PPC	SPARC	Alpha	MIPSbe	MIPSle
Linux	1.0.57 *newest*	1.0.57 *newest*	1.0.28	1.0.28	1.0.28	1.0.23	1.0.28
Darwin (Mac OS X)	1.0.55	1.0.55	1.0.47	⬛	⬛	⬛	⬛
Solaris	1.0.56	1.0.55	⬛	1.0.23	⬛	⬛	⬛
FreeBSD	1.0.23	1.0.22	⬛	🟥	🟥	⬛	⬛
NetBSD	1.0.22	1.0.56	1.0.23	🟥	🟥	🟥	🟥
OpenBSD	1.0.55	1.0.55	1.0.55	🟥	🟥	⬛	⬛
Windows	1.0.55	🟨	⬛	⬛	⬛	⬛	⬛

In addition to the official SBCL, a Windows fork exists that improves support for the Windows platform, especially in the area of threads, I/O, and x86-64 support. Though is has not yet been incorporated into mainline, Windows users may want to consider using it in the meanwhile.

	Key
🟩	Available and supported
🟨	Port in progress
🟥	Not available (porters welcome!)
⬛	No such system

	Processors
X86	X86 (32-bit Intel and compatible)
AMD64	64-bit X86 (AMD64, EM64T, Via Nano)
PPC	PowerPC
SPARC	SPARC and UltraSPARC
Alpha	DEC Alpha
MIPSbe	MIPS (big endian mode)
MIPSle	MIPS (little endian mode)

3.2.5 The concurrency problem

The tools for managing threads and concurrent operations are not part of the ANSI Specification and thus left as an “exercise for the reader” :(

4 Pourquoi Java?

4.1 Avantages Java

4.1.1 Cross Plaform

WORA $^{\text{RunDebug}}$ is still better

The assembly language coding has been done for you
Zero assembler JIT

4.1.2 Rich set of libraries

Many many libraries are available for Java

4.1.3 Enterprise adoption

Very popular

4.1.4 Performance: Dynamic code optimization

HotSpot Virtual Machine

- on the fly profiling,
- inlining, loop unrolling
- de-opt/reopt
- escape analysis
- dead code elimination

4.1.5 Performance: Garbage Collection

Several proven GC algorithms

- throughput
- pause time

4.2 Inconvenients Java

4.2.1 Mutation is (almost) required

Graph of mutable, stateful objects are a nightmare to manage with concurrency

Unconcious mutation is a source of bugs

- passing mutable objects to functions
- using mutable objects as keys

Coping mechanisms

- copy constructurs "freeze state" in a snapshot
- deep copy
- collections offer a weak facade

4.2.2 Single inheritence hierarchy

Object Oriented Programming is used for **everything** even when it doesn't make sense

- java.lang.Math has to gather up a bunch of static functions

Interfaces are a soft attempt at multiple inheritance

Aspect oriented programming is an attempt to avoid code duplication in the face of strong typing.

4.2.3 Complex

Java Fetishizes Complexity

4.2.4 Java EE Containers

file:~/src/irill/overview-architecture-cont.gif

4.2.5 Java EE APIs in the Web Container

file:~/src/irill/overview-architecture-web.gif

4.2.6 Java EE APIs in the EJB Container

file:~/src/irill/overview-architecture-ejb.gif

4.2.7 Java EE APIs in the Application Client Container

file:~/src/irill/overview-architecture-acc.gif

5 Pourquoi Clojure?

5.1 Avantages Clojure

5.1.1 Easy interoperation with Java

Embraces the power of the JVM

- Note: also runs on the CLR and on JavaScript (*)

Typing support without the burden of strong typing
(defn #$^{P}ropertiesas{-}properties“Convertanyseqofpairstoajava.utils.Propertiesinstance.Usesas$
$strtoconvertbothkeysandvaluesintostrings.''\{:tagProperties\}[m](let[p(Properties.)](doseq[[kv]m](.se$
$strk)(as-strv)))p))$

5.1.2 Leverages advantages of a Lisp

Clojure models its data structures as immutable objects represented by interfaces

Many functions defined on few primary data structures (seq, map, vector, set).

Clojure multimethods decouple polymorphism from OO and types

- Supports multiple taxonomies

- Dispatches via static, dynamic or external properties, metadata, etc

Clojure is a Lisp-1

5.1.3 Multimethods

Multimethod Examples…

file:~/src/irill/multi-1.clj

file:~/src/irill/multi-2.clj

5.1.4 Lazy sequences

All Clojure collection types are sequences (as are Java collections and Arrays)

A **lazy sequence** will only compute contents when they are consumed.

file:~/src/irill/lazy-seq-1.clj

file:~/src/irill/lazy-seq-2.clj

5.1.5 Functional Programming

Immutable data + first-class functions, supporting recursion

Dynamic polymorphism

Emphasizes recursive iteration instead of side-effect based looping

user> (let [my-vector [1 2 3 4] my-map {:fred “ethel”} my-list (list 4 3 2 1)] (list (conj my-vector 5) (assoc my-map :ricky “lucy”) (conj my-list 5) my-vector my-map my-list)) -> ([1 2 3 4 5] {:ricky “lucy”, :fred “ethel”} (5 4 3 2 1) [1 2 3 4] {:fred “ethel”} (4 3 2 1))

5.1.6 Software Transactional Memory

Core data structures are immutable and can easily be shared between threads

Mutation **is** possible using locks to avoid conflicts

- dosync, ref, set, alter, et al, supports sharing changing state between threads in a synchronous and coordinated manner.
- The agent system supports sharing changing state between threads in an asynchronous and independent manner.
- The atoms system supports sharing changing state between threads in a synchronous and independent manner.
- The dynamic var system supports isolating changing state within threads.

5.1.7 No spec, one implementation

Disadvantages: All eggs in one basket

Advantages: Clojure works **everywhere** Innovation happens quickly Core data structures are extensible abstractions Vibrant community

5.2 Inconvenients Clojure

5.2.1 The state of Clojure Contrib (is a challenge)

“Modularization of Contrib”

http://dev.clojure.org/display/doc/Clojure+Contrib

Wait, why isn’t there a project.clj (for lein)?

- officially must use mvn (!) (lein originally could not deploy to remote mvn repos)

The idea is that everything that hasn't been modularized yet is supposedly either low quality or in low demand

Using clojars: change groupID to highlight it's non-canonical

Also it's tricky to find out what the **real** disposition of stuff is.. I wanted java-utils (moved to clojure.java.io)

6 La programmation concurrente

6.1 Threads

(def long-calculation (future (apply + (range 1e8)))) (deref long-calculation) @long-calculation

(def bg (future (Thread/sleep 5000) (println “done”))) @bg

6.2 Threading harness (for examples)

file:~/src/irill/futures.clj

6.3 Types of operations

Coordinated: multiple actors must cooperate to produce correct results

Synchronous: caller blocks evaluation

Operations	`Coordinated`	`Uncoordinated`
`Synchronous`	Refs	Atoms
`Asynchronous`		Agents

NOTE: as the focus of Clojure is in-process concurrency the Coordinated - Asynchronous case is not implemented directly in the language (e.g. more for databases)

6.4 Refs

STM has ACID properties (except D):

- Atomic
- Consistent
- Isolated
- (Durability)

(dosync ;; the body is a transaction (alter myref f arg1 arg2)) ;; mutation of a reference

Show CPU usage

- xterm -fn terminus-24 -geometry 89x1+0+0 -n mpstat -e mpstat -P ALL 5
-

file:~/src/irill/refs.clj

6.5 Atoms

Safe mutation within a thread: compare and set

;; The function f will be retried of the value of myatom changed during the call (swap! myatom f)

file:~/src/irill/atoms.clj

6.6 Agents

Agent example: clojure-redis benchmarks

file:~/src/redis-examples/redis-bench/src/redis_bench/core.clj

7 L'avenir prometteur

7.1 Java

Moore's law in combination with new architectures makes Java very attractive from mobile to super computers.

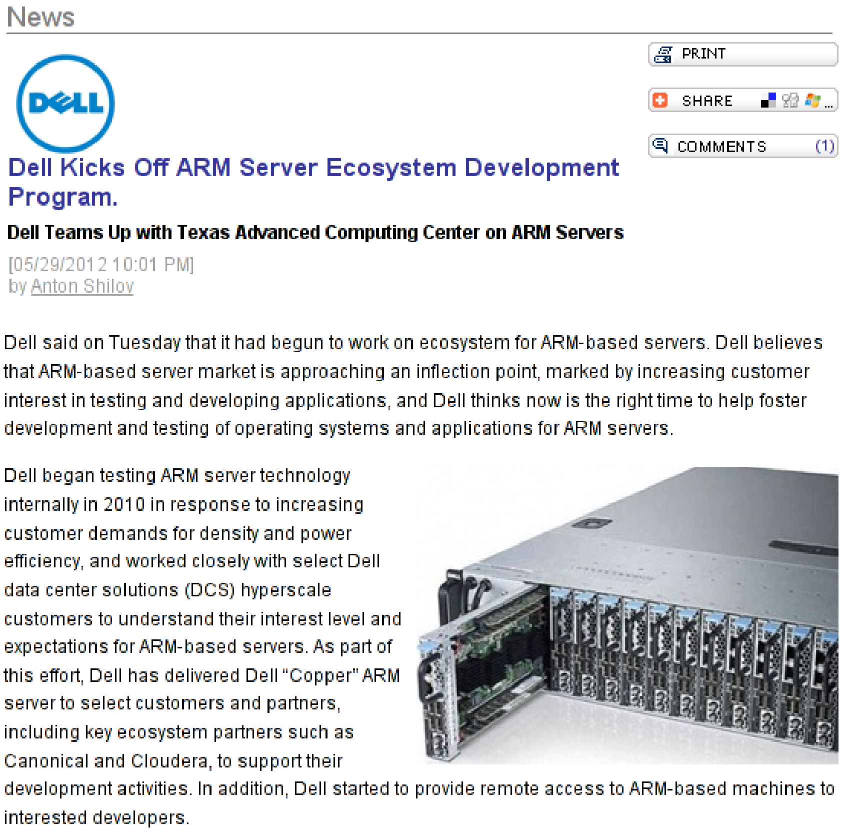
Sun originally wanted Java to enable customers to use SPARC

Today many Enterprises run on Intel architectures

But what about tomorrow?

7.1.1 ARM looks very good for size, cost, heat

Maybe we will see ARM in the data center? NOTE: Dell servers



Dell Kicks Off ARM Server Ecosystem Development Program.

Dell Teams Up with Texas Advanced Computing Center on ARM Servers

[05/29/2012 10:01 PM]
by Anton Shilov

Dell said on Tuesday that it had begun to work on ecosystem for ARM-based servers. Dell believes that ARM-based server market is approaching an inflection point, marked by increasing customer interest in testing and developing applications, and Dell thinks now is the right time to help foster development and testing of operating systems and applications for ARM servers.

Dell began testing ARM server technology internally in 2010 in response to increasing customer demands for density and power efficiency, and worked closely with select Dell data center solutions (DCS) hyperscale customers to understand their interest level and expectations for ARM-based servers. As part of this effort, Dell has delivered Dell "Copper" ARM server to select customers and partners, including key ecosystem partners such as Canonical and Cloudera, to support their development activities. In addition, Dell started to provide remote access to ARM-based machines to interested developers.

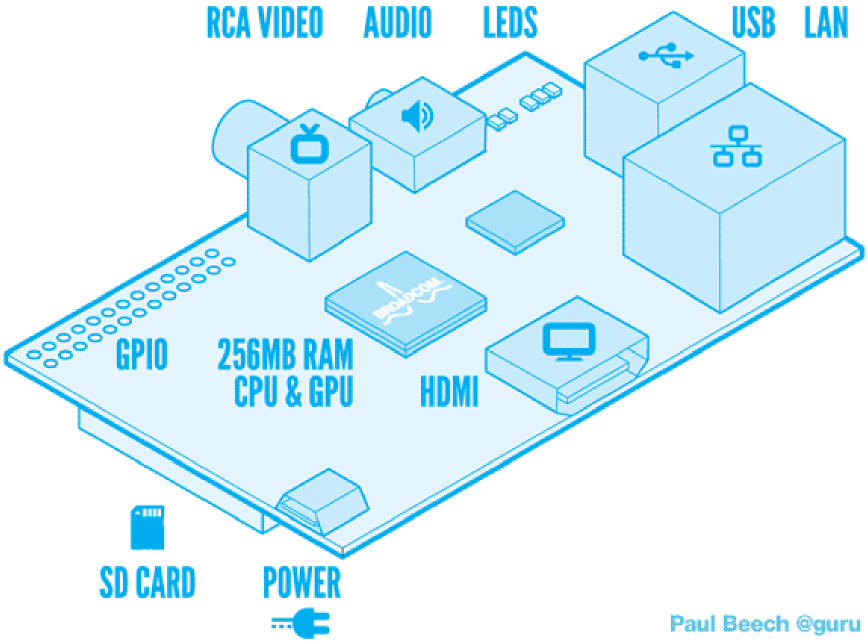
http://news.softpedia.com/news/Ubuntu-and-HP-Will-Power-ARM-Data-Centers-231827.shtml

7.1.2 We are seeing ARM everywhere in embedded devices

Raspberry Pi = $25

- SoC is a Broadcom BCM2835. This contains an ARM1176JZFS, with floating point, running at 700Mhz
- Videocore 4 GPU. The GPU is capable of BluRay quality playback, using H.264 at 40MBits/s.
- It has a fast 3D core accessed using the supplied OpenGL ES2.0 and OpenVG libraries.
- 256 MB RAM
- One USB port

- (Model B adds a 2nd USB port, Ethernet)



http://www.raspberrypi.org/

7.1.3 Java as assembly language

For these reasons Clojure is one of many vibrant, alternative languages on the JVM which include:

- JRuby
- Scala
- Jython
- IKVM.NET
- Gosu
- Smalltalk
- JavaScript

7.2 Bleeding Edge OpenJDK features

NOT yet truly being used by Clojure

7.2.1 Fork/Join

Bring Doug Lea's Fork/Join framework into Clojure

Primary example **pmap**

- using the shortest map/reduce tutorial ever

  user> (def mylist '(1 2 3 4 5 6)) #'user/mylist user> (map even? mylist) (false true false true false true) user> (reduce 'or (map even? mylist)) true

David Liebke: "From Concurrency to Parallelism" http://incanter.org/downloads/fjclj.pdf

7.2.2 Tail Call Optimization

Save space on the stack:

call factorial (3) call fact (3 1) call fact (2 3) call fact (1 6) call fact (0 6) return 6 return 6 return 6 return 6 return 6

call factorial (3) call fact (3 1) replace arguments with (2 3), jump to "fact" replace arguments with (1 6), jump to "fact" replace arguments with (0 6), jump to "fact" return 6 return 6

NOTE: Clojure does have **recur** and **trampoline** but the JVM itself lacks a generic optimization for TCO (but there is an older patch in the MVLM repo).

https://en.wikipedia.org/wiki/Tail_call

7.2.3 Invoke Dynamic

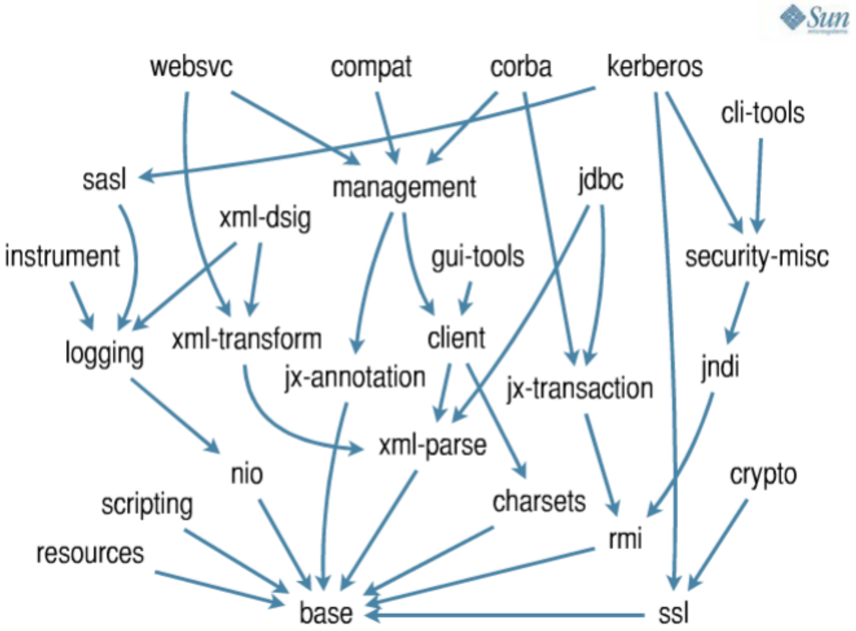
JSR 292

Enables the HotSpot VM to **see** into your "JVM Language" code and optimize it!

Why Clojure Doesn't Need Invokedynamic (Unless You Want It to be More Awesome) http://blog.headius.com/2011/10/why-clojure-doesnt-need-invokedynamic.html

7.2.4 Modularization (Jigsaw)

Better startup time Finer grained dependencies Smaller footprint (embedded)



- Note's on what I put in my .emacs.d/custom.el
  (require 'org-tree-slide)

  (global-set-key (kbd "<f1>") 'show-all) (global-set-key (kbd "<f5>") 'text-scale-decrease) (global-set-key (kbd "<f6>") 'text-scale-increase) (global-set-key (kbd "<f8>") 'org-tree-slide-mode) (global-set-key (kbd "<f9>") 'org-tree-slide-content) (global-set-key (kbd "<f10>") 'hide-sublevels)

  Printing to PDF: C-c C-e p (org-export-as-pdf)

8 Conclusion

LISP is incredibly powerful (don't be afraid of the parens)

Clojure is the best LISP now (because of the JVM)

Java means future proof for platforms in the cloud and the "Internet of Things".

There are **still** many optimizations waiting to be made

The #1 reason to use Clojure: productivity.

Clojure: Copyright 2008-2012 Rich Hickey http://clojure.org
More information on my blog: http://tmarble.info9.net

9 Q/A

Questions?

1. Mutability of Java
2. Skeptical of JVM
   - HotSpot
     - http://en.wikipedia.org/wiki/HotSpot
   - JSR 166
     - http://en.wikipedia.org/wiki/Java_concurrency
     - http://www.javacodegeeks.com/2012/04/fork-and-join-in-java-7-jsr-166.html
   - Libraries
3. Strong Typing?
   - http://clojure.org/rationale
4. Invoke Dynamic
   - http://openjdk.java.net/projects/mlvm/jvmlangsummit/
   - http://blog.headius.com/2008/09/first-taste-of-invokedynamic.html
   - http://java.dzone.com/articles/java-7-complete-invokedynamic

10 Live Hacking

file:~/src/irill

10.1 Command line processing and configuration files

tools.cli https://github.com/clojure/tools.cli awesome, right?

tmarble@noir 102$ lein search tools.cli == Results from central - Showing page 1 / 1 total [org.clojure/tools.cli "0.1.0"] [org.clojure/tools.cli "0.1.0"] tmarble@noir 103$

10.2 Pretty Print HTML and XML

I created a future-contrib package: file:~/src/maas/clojure/future-contrib/project.clj

See file:~/src/maas/clojure/future-contrib/src/future_contrib/core.clj

Demonstrate example with file:~/src/clojuremn/example.xml

10.3 redis2xml

Demonstrates command line processing and configuration files

see file:~/src/maas/clojure/redis2xml/project.clj

see: file:~/.redis2xml

also try command line:

```
redis-cli -a NoOneWillEverGuess -n 3
./bin/redis2xml -v -n 3 -f -i ~/src/clojuremn/example.xml
```

10.4 Example Noir site

See file:~/src/noir-examples/my-website

11 Extra

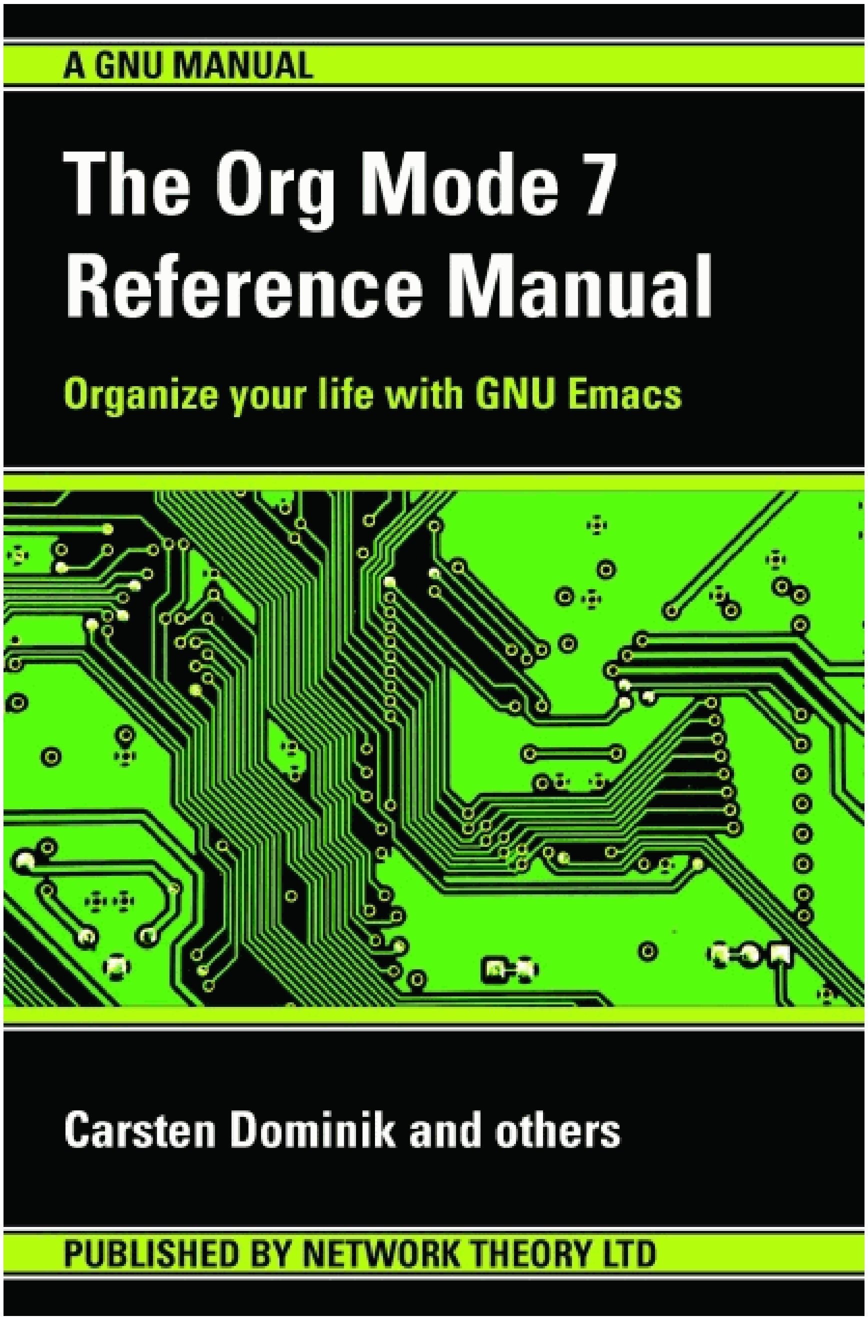
11.1 What is that presentation tool?

Emacs!

This is **org-tree-slide** from https://github.com/takaxp/org-tree-slide

For more on org mode see http://orgmode.org/org.html

Yes I will share my “slides” on my website http://tmarble.info9.net



A GNU MANUAL

The Org Mode 7 Reference Manual

Organize your life with GNU Emacs

Carsten Dominik and others

PUBLISHED BY NETWORK THEORY LTD

11.2 The Tools I am using

11.2.1 Maven

Finding dependencies: `mvn dependency:tree -DoutputFile=dependency.txt`

```
   my-website:my-website:jar:0.1.0-SNAPSHOT +- org.clojure:clojure:jar:1.3.0:compile
 noir:noir:jar:1.2.2-SNAPSHOT:compile  +-  compojure:compojure:jar:1.0.0-
RC2:compile

      +- org.clojure:core.incubator:jar:0.1.0:compile
      +- org.clojure:tools.macro:jar:0.1.0:compile
      +- clout:clout:jar:1.0.0:compile
       ring:ring-core:jar:1.0.1:compile
      +- commons-io:commons-io:jar:1.4:compile
      +- commons-fileupload:commons-fileupload:jar:1.2.1:compile
       javax.servlet:servlet-api:jar:2.5:compile

   +- org.clojure:tools.namespace:jar:0.1.0:compile

              org.clojure:java.classpath:jar:0.1.0:compile

   +- clj-json:clj-json:jar:0.4.3:compile

         org.codehaus.jackson:jackson-core-asl:jar:1.5.0:compile

   +- ring:ring:jar:1.0.1:compile


 +- ring:ring-devel:jar:1.0.1:compile
                                        ns-tracker:ns-tracker:jar:0.1.1:compile
 +- ring:ring-jetty-adapter:jar:1.0.1:compile
                                       +- org.mortbay.jetty:jetty:jar:6.1.25:compile
                                                                                org.mor
                                        org.mortbay.jetty:jetty-util:jar:6.1.25:compile
  ring:ring-servlet:jar:1.0.1:compile

   +- hiccup:hiccup:jar:0.3.7:compile +- clj-stacktrace:clj-stacktrace:jar:0.2.3:compile
+- ring-reload-modified:ring-reload-modified:jar:0.1.1:compile +- net.java.dev.jets3t:jets3t:jar:0.8.1:comp

       +- commons-codec:commons-codec:jar:1.3:compile
       +- commons-logging:commons-logging:jar:1.1.1:compile
       +- commons-httpclient:commons-httpclient:jar:3.1:compile
        com.jamesmurty.utils:java-xmlbuilder:jar:0.4:compile

    org.mindrot:jbcrypt:jar:0.3m:compile
```

11.2.2 Leiningen

Leiningen is awesome https://github.com/technomancy/leiningen

Use the REPL **swank-clojure** https://github.com/technomancy/swank-clojure

Get... $ lein plugin install swank-clojure 1.4.0 $ lein plugin install lein-localrepo 0.3 $ lein plugin install lein-noir 1.2.1

`lein localrepo help`

Public Repos: http://clojars.org/

Private Repos: https://github.com/technomancy/s3-wagon-private

Lein directly from git: https://github.com/tobyhede/lein-git-deps

11.2.3 Redis

Amazing NoSQL Database: http://redis.io

With a Clojure binding! https://github.com/mmcgrana/clj-redis

Redis utterly killed it in 2010 – check out the growth in share of developer conversation http://www.redmonk.com/jgovernor/2012/03/15/redis-utterly-killed-it-in-2010-check-out-the-growth-in-share-of-developer-conversation/

11.2.4 Jenkins

Continuous Integration Server: http://jenkins-ci.org/

Amazing Plugins: https://wiki.jenkins-ci.org/display/JENKINS/Plugins

The ones that I use:

- Trac Publisher
- Dependency Graph Viewer
- IM
- Pathignore (essential for big git repo)
- SSH Slaves
- Thin Backup
- Build Result Trigger

Fun ones

- Gravatar
- Emotional Jenkins

KK slides from February at MonkiGras in London http://www.slideshare.net/kohsuke/building-developer-community

11.2.5 Using Jenkins

Git push triggers Jenkins Updates the one (master) workspace Projects started based on updated paths

Java Client

- Builds on Linux
- Triggers native Mac OS X build on Mac slave
- Triggers native Windows build on Windows slave

Deploying Noir application

- shuts down dev website
- updates code
- restarts website

11.2.6 Trac

http://trac.edgewall.org/

- Tickets (bugs, tasks), Reports, Browse code, Timeline, Wiki
- Can now use git (yeah!)
- Integration with Jenkins http://trac-hacks.org/wiki/XmlRpcPlugin

11.2.7 Noir

Let's talk about Noir http://webnoir.org

11.3 Why Open Source Matters

Free as in Free Beer

Free as in Free Speech

Knowing the shape of the solutions: Ease of integration

No marketing: just code (extra credit: build in tests and Jenkins)

Fewer bugs (recent Coverity study)

Education, credentials and employment

- Employers **will** google you
- Many directly ask for pointers to FLOSS contributions

11.3.1 Where are you going to deploy that code?

The “cloud”.

Are you really going to deploy to Windows?

- you have to name your machines #FAIL
- you have to Remote Desktop in and click-to-admin #FAIL
- no anticipated downtime until 2016 :)

You can’t deploy to Mac OS X

- X Serve died a long time ago

You want to deploy to Linux

- Cost effective
- Legal
- More reliable
- More automatable

11.3.2 permissive vs. restrictive licensing

BSD (MIT AL2) vs. GPL (MPL)

Permissive is necessary, but sometimes not enough to hold a community together.

Jeremy Allison: Why Samba Switched to GPLv3 2011 Linux Collaboration Summit http://faif.us/cast/2011/may/10/0x0F/

NOTE: proprietary (dual) licensing with contributor license agreements is now considered harmful

11.3.3 Open Source and Web Services

What if you want to build a strong community around a web service?

In the “cloud” the GPL is just like BSD.

The answer? The AGPL (Affero General Public License)

From the FSF The GNU Affero General Public License is a modified version of the ordinary GNU GPL version 3. It has one added requirement: if you run the program on a server and let other users communicate with it there, your server must also allow them to download the source code corresponding to the program that it’s running.

What? I’m going to build a business on AGPL? Is that CRAZY?

It is being done now: http://status.net “Enterprise Social Software is OPEN for business.”

11.3.4 Where is the value?

Productivity!

Right Now

- Hardware is effectively free
- The best software in life is Free
- Savoir Faire (brainpower) is expensive
- Data are like diamonds: they vary in clarity, quality and value

New business models need to maximize productivity around managing and improving quality of data.

(NOTE: China doesn’t care about intellectual property anyway)

11.3.5 Why Debian

Commitment to quality and building everything from source

Package inter-dependencies are core to the system

- Windows needs Maven, Gems, cygwin, etc.
- Mac needs MacPorts, etc.

Very predictable, easy to administer & automate, secure, stable

One of the two major Linux families (.deb and .rpm) and the foundation of many derivatives (e.g. Ubuntu)

DebConf11
24.-30.07.2011. Banja Luka, BIH
Rule 1: United. We stand.
Rule 2: Divided ... we argue a bit on the lists and then - see Rule 1.

http://wiki.debconf.org/wiki/DebConf11/Pictures/GroupPhoto