

Parallelizing Quickref

~ ELS 2019 ~

Didier Verna
EPITA / LRDE

didier@lrde.epita.fr



[lrde/~didier](#)



[@didierverna](#)



[didier.verna](#)



[in/didierverna](#)



- ▶ Quickref parallelized
- ▶ Performance improvement: 4x

Thank you! Any questions?



Introduction



Jump to: # A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Quickref

Reference manuals for Quicklisp libraries

Downloaded from [quickref.com](#) on 2015-01-01.
Documentation generated with [quickref](#) 1.0.7. The "quickref" package is a "lib" package. All rights reserved.

Index Entry

#		
1am	3bmd	3d-vectors
3b-paf	3d-matrices	cl-6502
3dgl-shader		
A		
a-cl-logger	cl-annot	asdf
able	cl-annot-prove	asdf-dependency-grovel
abnf	cl-annotfun	asdf-encodings
abstract-datas	cl-ansi-term	asdf-finalizers
access	cl-ansi-text	asdf-linguist
acclimation	antk	asdf-manager
cl-acronymia	cl-apple-plist	asdf-package-system
cl-actors	apply-argv	asdf-system-connections
advanced-readable	apri	asdf-vis
active-charting	arc-compat	aserve
agnostic-board	architecture-builder-protocol	assoc-utils
egutil	architecture-service-provider	cl-association-rules
atosa	archive	asternids
alexandria	cl-art-parser	cl-async
algebraic-data-library	amesi	cl-async-future
cl-algebraic-data-type	array-operations	atdoc
cl-all	array-utils	authenticated-encryption
also-also	arrow-macros	cl-autorepo
amazon-ecs	arrows	cl-autowrap
cl-amqp	cl-arrows	avstar-api
cl-ans	cl-ansi-api	aws-foundation

- ▶ Reference manuals for CL (Quicklisp) libraries
- ▶ `quickref.common-lisp.net` or local builds
- ▶ Originally: sequential loop over 1700+ libraries
- ▶ 1h30 – 7h depending on the conditions
- ▶ *Parallelism worth investigating*

Plan

Toolchain

Experiments

Parallel Solution

Discussion and Perspectives

Toolchain

Experiments

Parallel Solution

Discussion and Perspectives

Toolchain



Important remarks:

- ▶ Dectl works by introspection
 - ▶ Compilation / loading (of dependencies) may be required
 - ▶ Avoid loading 1700+ libraries in the same Lisp image!
 - ▶ Run Dectl in external processes
- ▶ Makeinfo is a Perl/C script
 - ▶ *Ditto*

Plan

Toolchain

Experiments

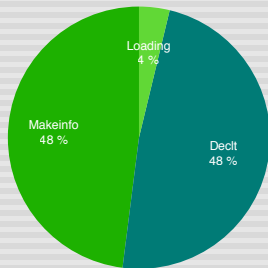
Parallel Solution

Discussion and Perspectives

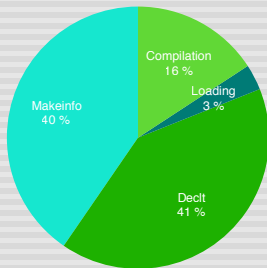
Scenarios

- ▶ Quickref Options
 - ▶ Libraries and Update
 - ▶ full / installed-only
 - ▶ download / compilation may occur
 - ▶ Compilation cache Policy
 - ▶ global / local
 - ▶ global cache may cause problems
- ▶ Scenarios
 1. All libraries already compiled (1h 27m)
 2. Nothing compiled, global compilation cache (1h 51m)
 3. Nothing compiled, local compilation cache (7h 01m)

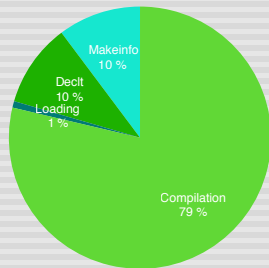
Time Distribution



1



2



3

Scenario	Texinfo	HTML
1	52%	48%
2	60%	40%
3	90%	10%

Plan

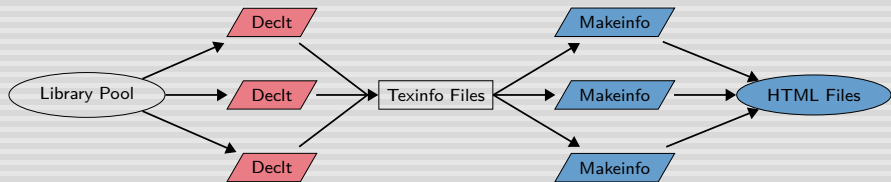
Toolchain

Experiments

Parallel Solution

Discussion and Perspectives

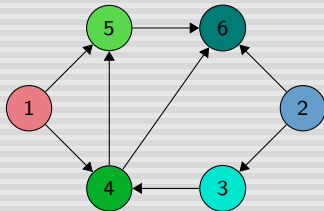
Parallel Solution



- ▶ Usable in all scenarios
- ▶ Dependency graph management cheap
- ▶ Scenario 2
 - ▶ Best results: 4 Declt threads / 4 Makeinfo threads
 - ▶ Total time: 28m 17s (25% of sequential time)

Library Pool Management

Dependency Graph

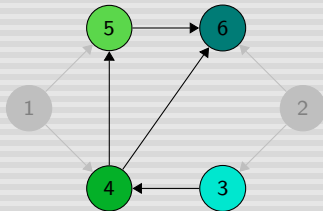


Library Pool

Done

Library Pool Management

Dependency Graph



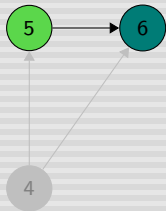
Library Pool



Done

Library Pool Management

Dependency Graph



Library Pool



Done



Library Pool Management

Dependency Graph



Library Pool



Done



Library Pool Management

Dependency Graph

6

Library Pool



Done



Library Pool Management

Dependency Graph

Library Pool

Done



Plan

Toolchain

Experiments

Parallel Solution

Discussion and Perspectives

Discussion and Perspectives

- ▶ Alternative Algorithms
- ▶ Dependency Management Issues
 - ▶ Based on static information provided by Quicklisp
 - ▶ Fragile, not always correct
- ▶ CPU vs. I/O Consumption
 - ▶ 4x is a bit disappointing
 - ▶ Open the Declt and Makeinfo black boxes
- ▶ SJF-like Scheduling
 - ▶ Very difficult to figure out where complexity comes from
 - ▶ Collect timings and use them in next run
- ▶ SSD!

Acknowledgments

- ▶ Initial code base: Antoine Martin
- ▶ Author index and parallel algorithm #4: Antoine Hacquard
- ▶ Hosting (code & website): CLF / `common-lisp.net`