Cognitive Interaction Toolkit - Support #323

set c/c++ compiler invocation to "-j2" to default?

2014-11-15 14:19 - Florian Lier

Status:	Closed		**Start date:**	2014-11-15
Priority:	Normal		**Due date:**	
Assignee:			**% Done:**	50%
Category:	Recipe		**Estimated time:**	0.00 hour
Target version:	Toolkit 0.1		**Spent time:**	0.00 hour

Description

We currently have 2 "executers" as the default in Jenkins so "make -j2" should be perfectly doable as most machines today have at least 4 threads available.

Shall we do this?

IMHO this would speed up the deployment process.

History

#1 - 2014-11-27 07:59 - Florian Lier

Does feedback mean it is implemented?

#2 - 2014-11-28 04:42 - Jan Moringen

Florian Lier wrote:

> Does feedback mean it is implemented?

No.

If we decide to do this, it can mostly be achieved on the template level, I think. Most of the work will be designing good defaults, allowing the defaults to be overwritten and not making the solution CMake/make-specific.

@Johannes: I have never seen parallelization wihtin Jenkins jobs. Do you know whether this is a supported use-case?

#3 - 2014-11-28 17:09 - Johannes Wienke

This somehow contradicts the idea of jenkins executors.

#4 - 2015-01-05 09:19 - Robert Haschke

For me, it's not clear, why jenkins executors should be single-threaded.
If there is only a single job to execute, it will greatly benefit from parallelization. Otherwise, i.e. having multiple parallel jobs that in turn are parallelized, will not harm much, wouldn't it?

Hence, I vote for job-internal parallelization too.

#5 - 2015-01-05 10:53 - Johannes Wienke

Using jenkins parallelization + job parallel builds will easily flood the available working memory on the computer.

#6 - 2015-02-20 18:21 - Florian Lier

I have the feeling at some point this has been implemented in the famula branch, is that correct? I would, again, vote for a -D j=int option with a comment "expert" mode.

Jenkins comes with #2 executers by default and most machines these days support at least 4 threats. I believe parallel C/C++ builds are more efficient than more executers. For example if a c/c++ job which has a lot of downstream dependencies (that are correctly blocked in this case) takes forever, parallel executers deliver no speed up.

So, if you know what you are doing, and you have 2 executers and let's say 8 threats available -j >2 won't hurt much.

#7 - 2015-02-20 22:10 - Robert Haschke

Yes, in the famula branch I added an appropriate variable make.threads for autotools and freestyle projects. However, I wasn't able to influence cmake builds in this manner.

Even having many executors AND many build threads typically doesn't harm - as long as enough memory is available.

#8 - 2015-08-26 09:25 - Florian Lier

Did this feature make it into the master?

#9 - 2015-08-26 10:17 - Jan Moringen

- Assignee deleted (Jan Moringen)

Florian Lier wrote:

> Did this feature make it into the master?

I don't know but this can be done entirely on the template level.

#10 - 2015-10-29 11:26 - Robert Haschke

As far as I could see (when I was searching for an appropriate option),
this cannot be handled on the template level.
There is no make command specified in the templates for cmake:

    grep -r [^c]make .

./toolkit/T600-freestyle.template: "make.command": "
./toolkit/T600-freestyle.template:make -j${make.threads|1}
./toolkit/T700-autotools.template:make -j${make.threads|1}
./ci/T600-freestyle.template: "make.command": "
./ci/T600-freestyle.template:make -j${make.threads|1}

./ci-deploy/T600-freestyle.template: "make.command": "
./ci-deploy/T600-freestyle.template:make -j${make.threads|1}
./ci-deploy/T700-autotools.template:make -j${make.threads|1}
./_common/T800-runnable.template: "make.command": "
./_common/T800-runnable.template:make -j${make.threads|1}

The corresponding jenkins commands are generated by the binary...

#11 - 2015-10-30 13:41 - Jan Moringen

- Status changed from Feedback to In Progress

- % Done changed from 0 to 50

build-generator master allows this. Not sure what good defaults are, though.

#12 - 2015-11-02 12:19 - Jan Moringen

- Category changed from build-generator to Recipe

#13 - 2017-03-03 10:09 - Florian Lier

- Status changed from In Progress to Closed