

Dissecting Reducers

A code tour of a powerful functional abstraction



WGSN

About me

- @reborg
- Clojure Weekly, <http://reborg.net>
- Lead-Dev WGSN.com

Polite warning



**Clojure knowledge
assumed**

Just enough background

- ***pmap*** simple approach with limitation
- ***preduce***, deprecated fork-join wrapper
- ***pv-map***, experimental fork-join
- ***reducer***, native composable fork-join

What you will see

- Functional problem decomposition
- Some functional idioms
- A few refactoring moves

Feel free to interrupt anytime

Our goal

1. Function with similar interface to reduce
2. Accept composable transformations
3. Easy and efficient parallel execution

let's get started

live code...

For offline readers

Please download the annotated live code session from:

<https://github.com/reborg/talk-strangeloop2013/blob/master/src/strloop2013/core.clj>

Play with reducers?

- `brew tap reborg/taps`
- `brew update`
- `brew install pwc`
- `pwc -f some-large-file`

Resources

- <http://tinyurl.com/01-rich> the original blog
- <http://tinyurl.com/02-article> my SDJ article
- <http://tinyurl.com/03-reducers-clj> source
- <https://github.com/reborg/pwc> parallel wc