

# Parallelizing Transducers

Clojure Exchange 2017 - Workshops

# Going parallel

- Transducers, especially on of large datasets, could run in parallel for additional performances.
  - **Approach 1.** Divide and conquer: split the input and work in parallel (only stateless transducers).
  - **Approach 2.** Master-workers. Workers fight to take the next item from the dataset.

# Parallelization with Reducers

- Reducers is a namespace part of Clojure: `clojure.core.reducers`
- They wrap the Java fork-join framework, a divide and conquer parallel strategy with work-stealing.
- They assume the input collection can be split into chunks (no laziness).
- They also assume combining the chunks back is commutative.
- Transducer state is accessed concurrently: no stateful transducers.

initial collection  
split into chunks

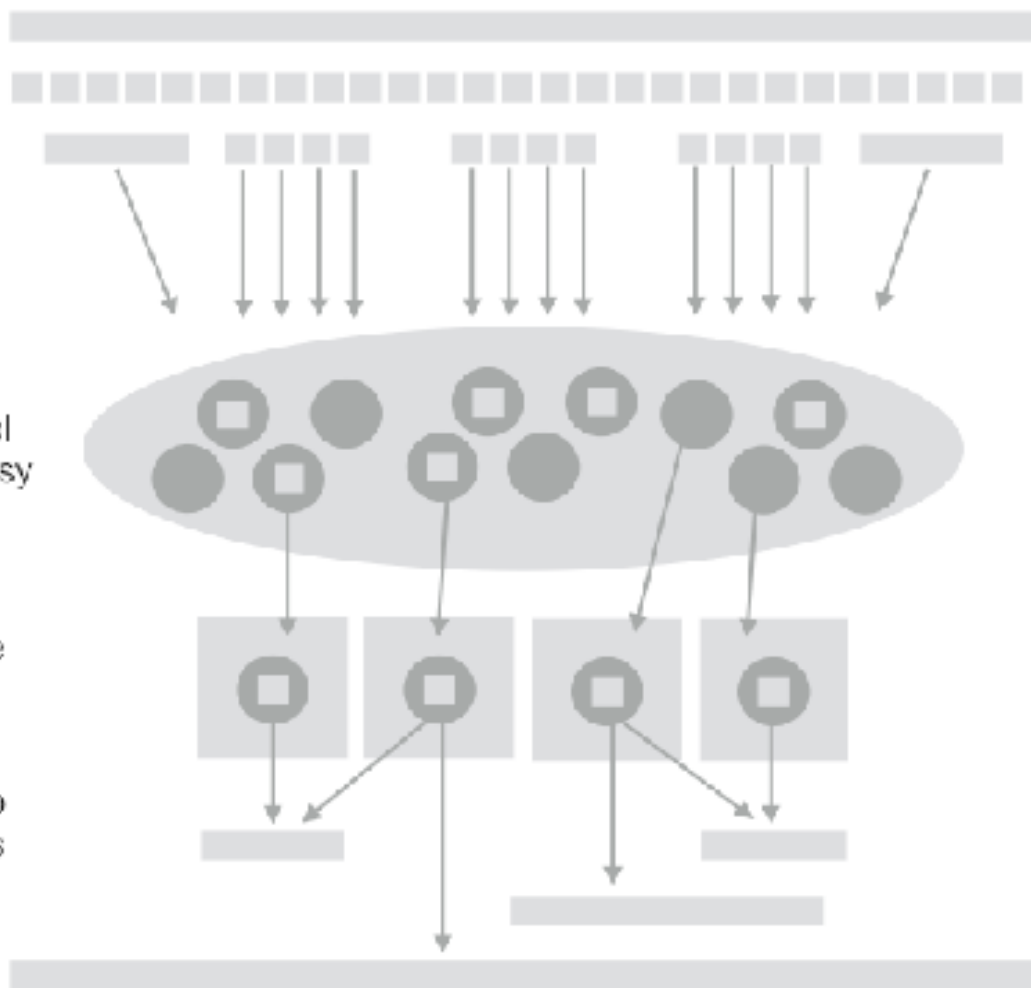
chunks assigned  
to available threads

fork-join thread-pool  
keeps CPU cores busy

reduce-fns execute  
on available cores

combine-fns also  
execute on cores

final results  
assembled



## To use Reducers

- Require the namespace
- `r/fold` is the main entry point
- They need a "reducef" reducing function (at the leaf).
- "combinef" function is used to join chunks back together.
- Apply the transducer chain (the one with `comp`) to the reducing function (such as "x" or "conj") to obtain "reducef"

# Parallelization with `core.async`

- `core.async` provides a "pipeline" construct to associate multiple parallel threads to a channel
- Pipelines can be further "piped" together and apply a different transducer each
- Input data can be streamed or loaded from a collection in one go.

# Resources

- ["A Java fork-join framework"](#) paper by Doug Lea
- [Clojure Applied](#) book contains chapters dedicated to Transducers with core.async pipes.
- [Standard Library book, Chapter 32: Reducers and Transducers](#)

# **Lab 04**

# **Parallelizing Transducers**



# Goal of Lab4

- **Task1:** parallelise the xform with reducers.
- **Task2:** parallelise the xform with pipelines.
- **Task3:** different pipelines for different transducers.

Open `transducers-workshop.lab04` to start.