

Readers-  
writers  
24/7

Konrad  
Kurdej

Outline

Readers-  
writers

Fast readers,  
slow writers

Some notes

# Readers-writers 24/7

Konrad Kurdej

18-12-2012

Readers-  
writers  
24/7

Konrad  
Kurdej

Outline

Readers-  
writers

Fast readers,  
slow writers

Some notes

## 1 Readers-writers

## 2 Fast readers, slow writers

## 3 Some notes

1 Readers-writers

2 Fast readers, slow writers

3 Some notes

Readers-  
writers  
24/7

Konrad  
Kurdej

Outline

Readers-  
writers

Fast readers,  
slow writers

Some notes

- definition (library/data)
- fairness / starvation

1 Readers-writers

2 Fast readers, slow writers

3 Some notes

Readers-  
writers  
24/7

Konrad  
Kurdej

Outline

Readers-  
writers

Fast readers,  
slow writers

Some notes

## Why bother?

Readers-  
writers  
24/7

Konrad  
Kurdej

Outline

Readers-  
writers

Fast readers,  
slow writers

Some notes

## Why bother?

It happens - **URLAnnotator** aka **buildaclassifier**.

## Why bother?

It happens - **URLAnnotator** aka **buildaclassifier**.

## Example

- reading takes 15 sec, very often (10 per minute)
- writing takes 30 min, once an hour

Outline

Readers-  
writers

Fast readers,  
slow writers

Some notes

## Why bother?

It happens - **URLAnnotator** aka **buildaclassifier**.

### Example

- reading takes 15 sec, very often (10 per minute)
- writing takes 30 min, once an hour

Readers would wait for 30 min ...

Outline

Readers-  
writers

Fast readers,  
slow writers

Some notes

## Why bother?

It happens - **URLAnnotator** aka **buildaclassifier**.

### Example

- reading takes 15 sec, very often (10 per minute)
- writing takes 30 min, once an hour

Readers would wait for 30 min ...

Lets do something about this!

# Lets work with classifier example

## Classifier operations

- $t$  - train
- $c$  - classify\_\*

Clearly this matches RW schema.

# Simple solutions

Outline

Readers-  
writers

Fast readers,  
slow writers

Some notes

- create new classifier for each train
- do the training
- swap newly trained one with *classifying* classifier

# Simple solutions

- create new classifier for each train
- do the training
- swap newly trained one with *classifying* classifier

Because we can do better - we won't go into details

## Only two classifiers

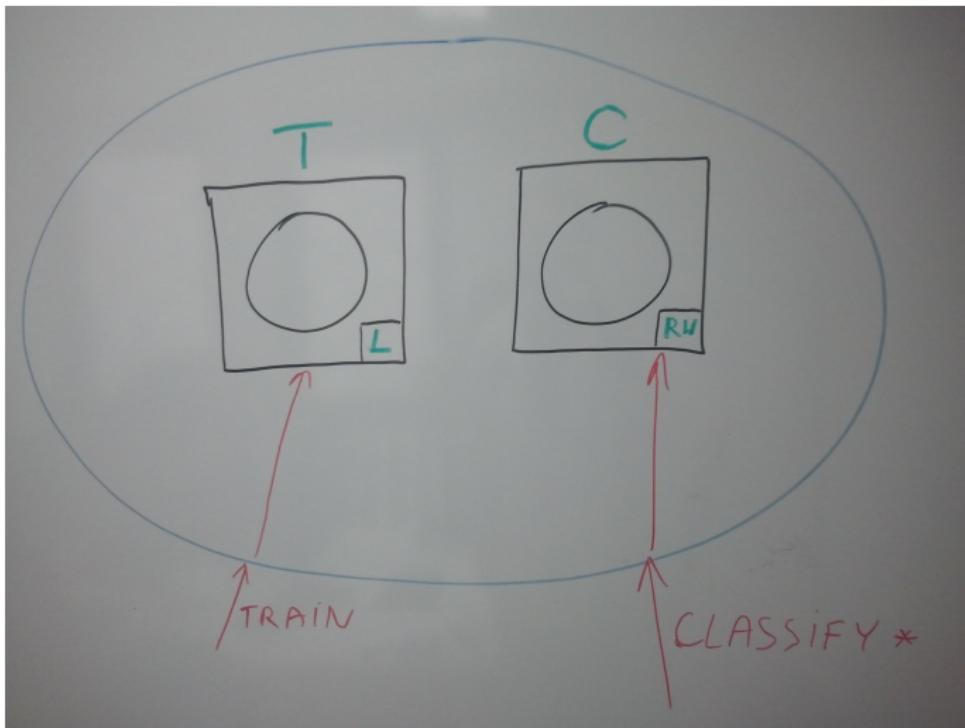
- two slots - **T**rain, **C**lassify
- protect **T** with normal lock/mutex etc.
- protect **C** with RW lock
- each classify-\* goes to **C**
- each train goes to **T**
- when we gain lock, we execute command on classifier found in slot
- after train, we don't release **T** lock, we are acquiring **C** lock and then do swap in slots

# Only two classifiers

- two slots - **T**rain, **C**lassify
- protect **T** with normal lock/mutex etc.
- protect **C** with RW lock
- each classify-\* goes to **C**
- each train goes to **T**
- when we gain lock, we execute command on classifier found in slot
- after train, we don't release **T** lock, we are acquiring **C** lock and then do swap in slots

Let see it in action

## Overall view



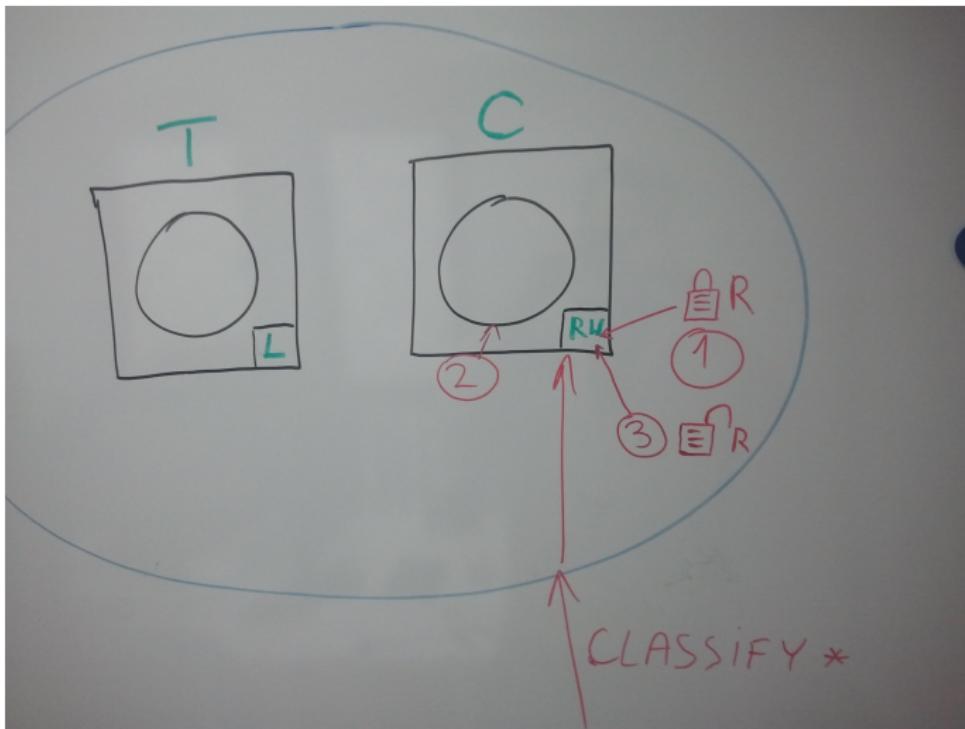
## Classify

## Outline

## Readers-writers

Fast readers,  
slow writers

Some notes



Readers-writers  
24/7

Konrad  
Kurdej

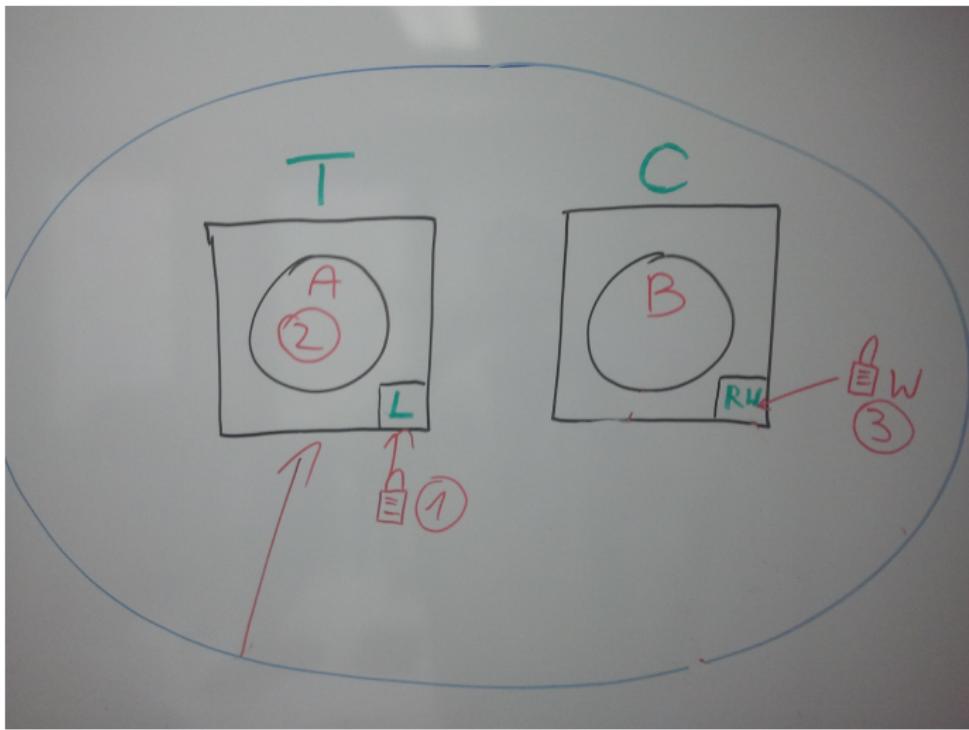
Outline

Readers-writers

Fast readers,  
slow writers

Some notes

# Train



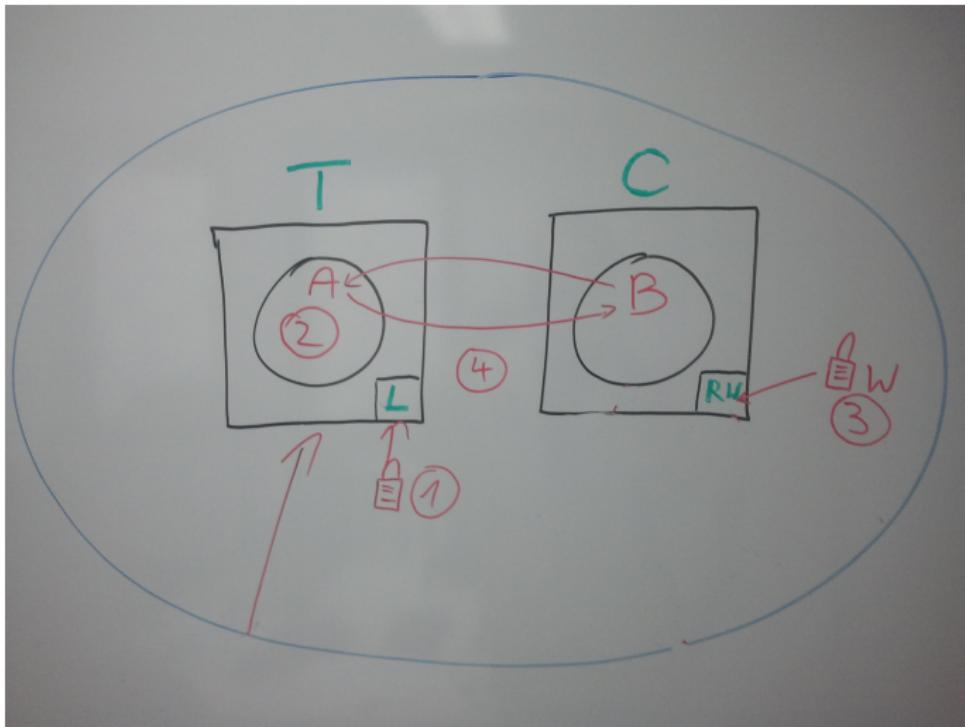
# Train

Outline

Readers-  
writers

Fast readers,  
slow writers

Some notes



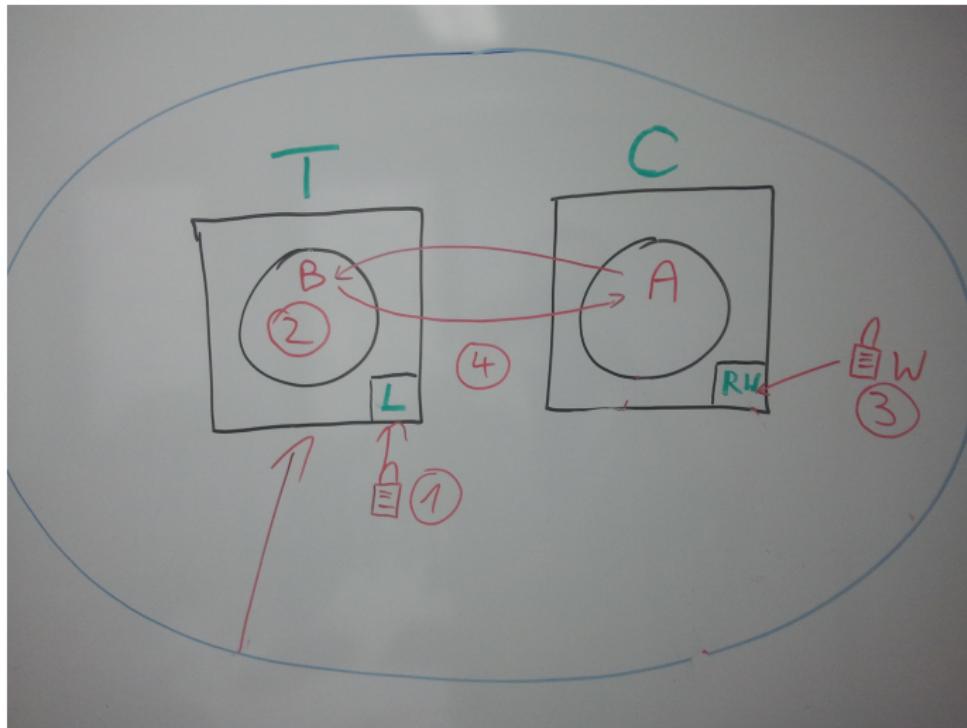
# Train

Outline

Readers-writers

Fast readers,  
slow writers

Some notes



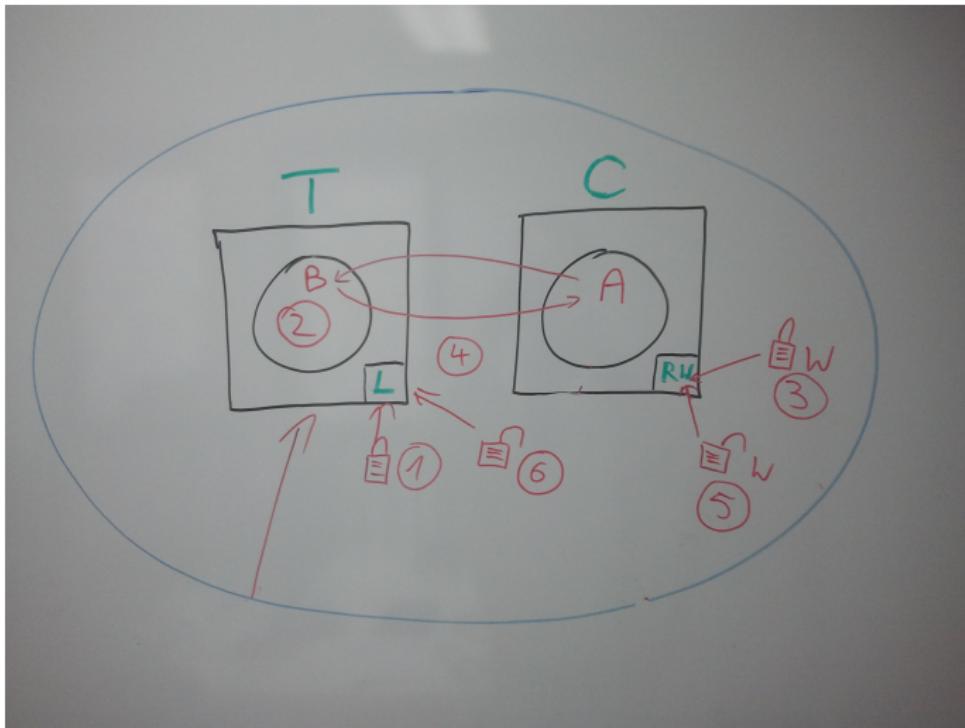
Outline

Readers-writers

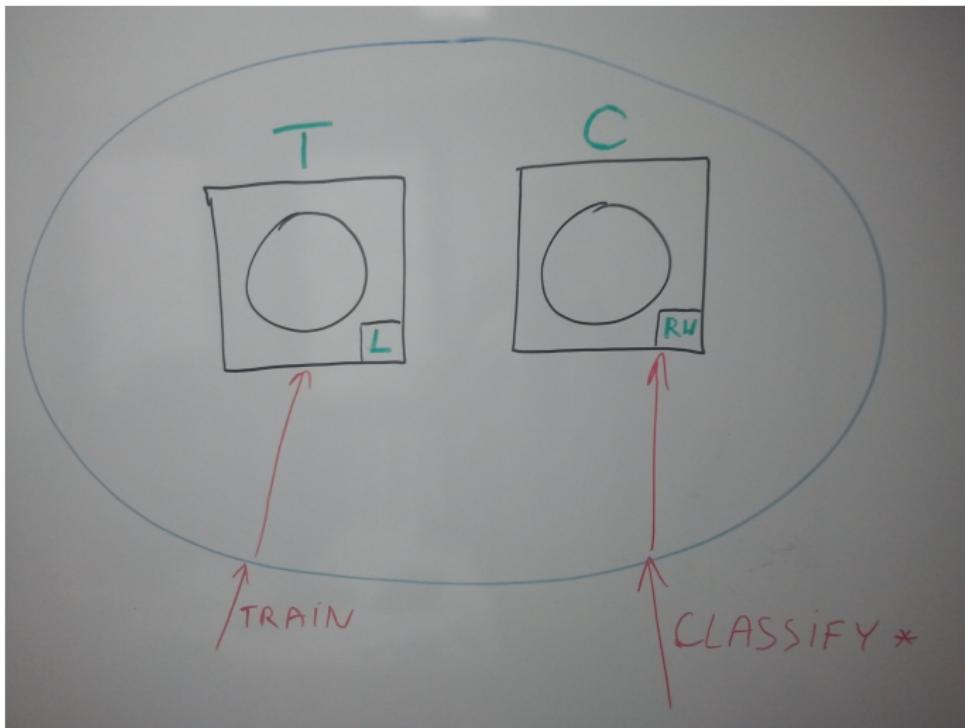
Fast readers,  
slow writers

Some notes

# Train



## Overall view



## Summary

- + no long reads
- + no need to know how to construct classifier - they are given at initialization
- - doubles memory/resource requirements

① Readers-writers

② Fast readers, slow writers

③ Some notes

- 24/7 matches *decorator pattern* schema
- Python stdlib lacks synchronization of not related processes
- Java stdlib implements RW synchronization only with locks

Readers-  
writers  
24/7

Konrad  
Kurdej

Outline

Readers-  
writers

Fast readers,  
slow writers

Some notes

# Q & m A



Outline

Readers-  
writers

Fast readers,  
slow writers

Some notes

Next time: slooooow readers ...

