progress engine
we will develop it

#tceh

2016

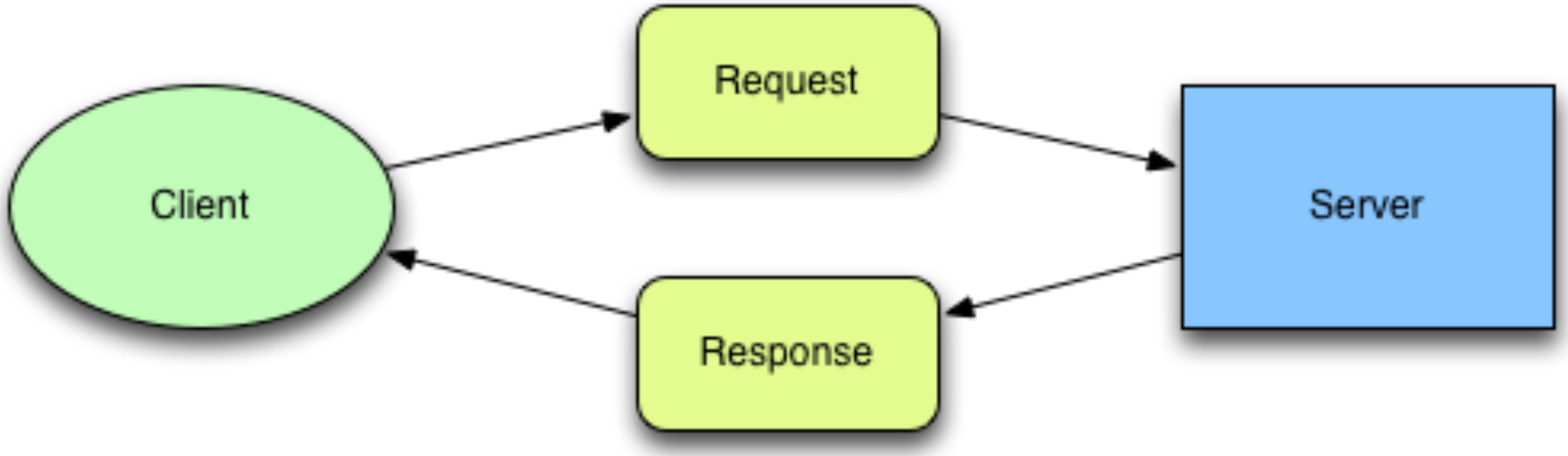
Врубиться в Ruby

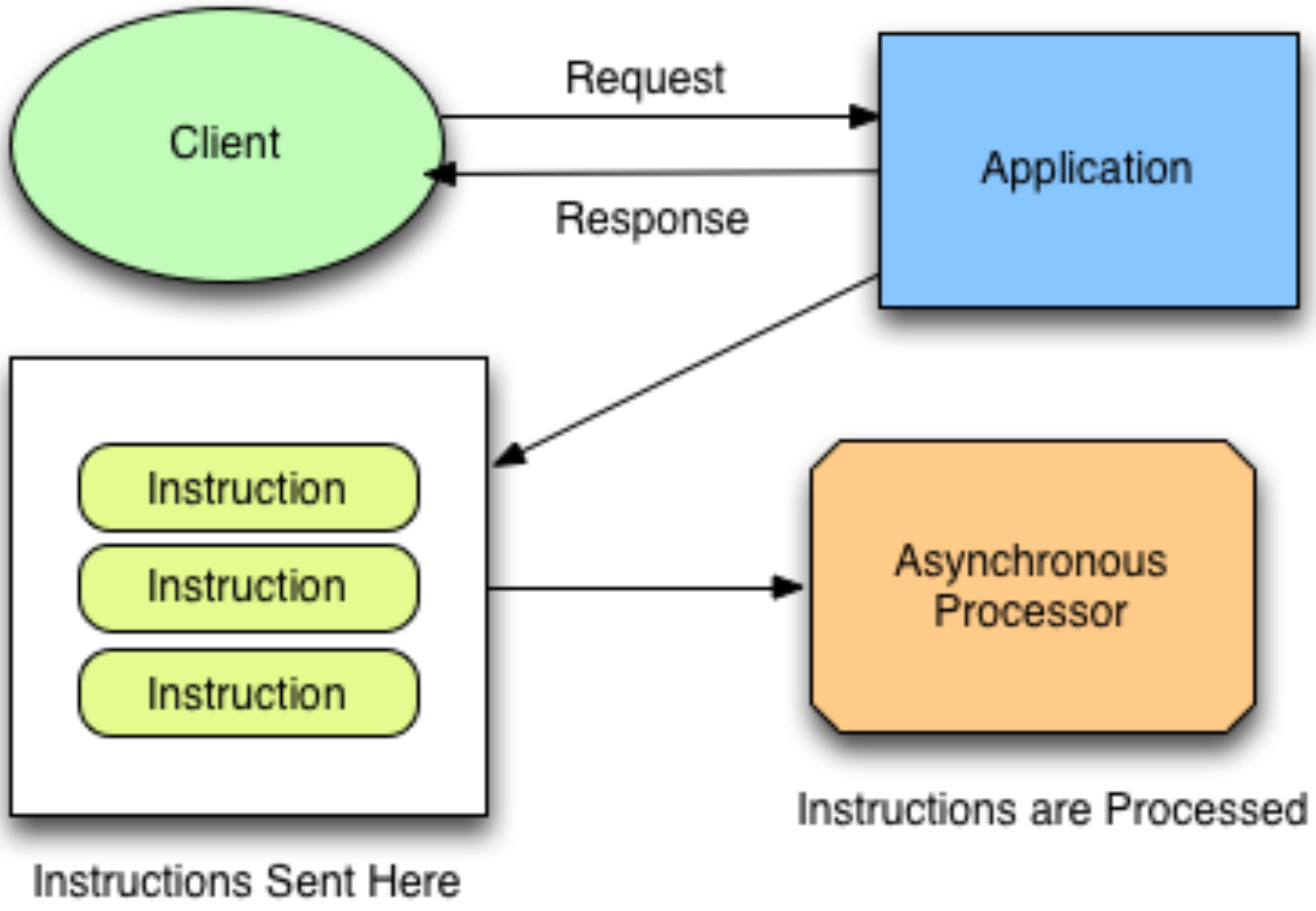
Лекция 15

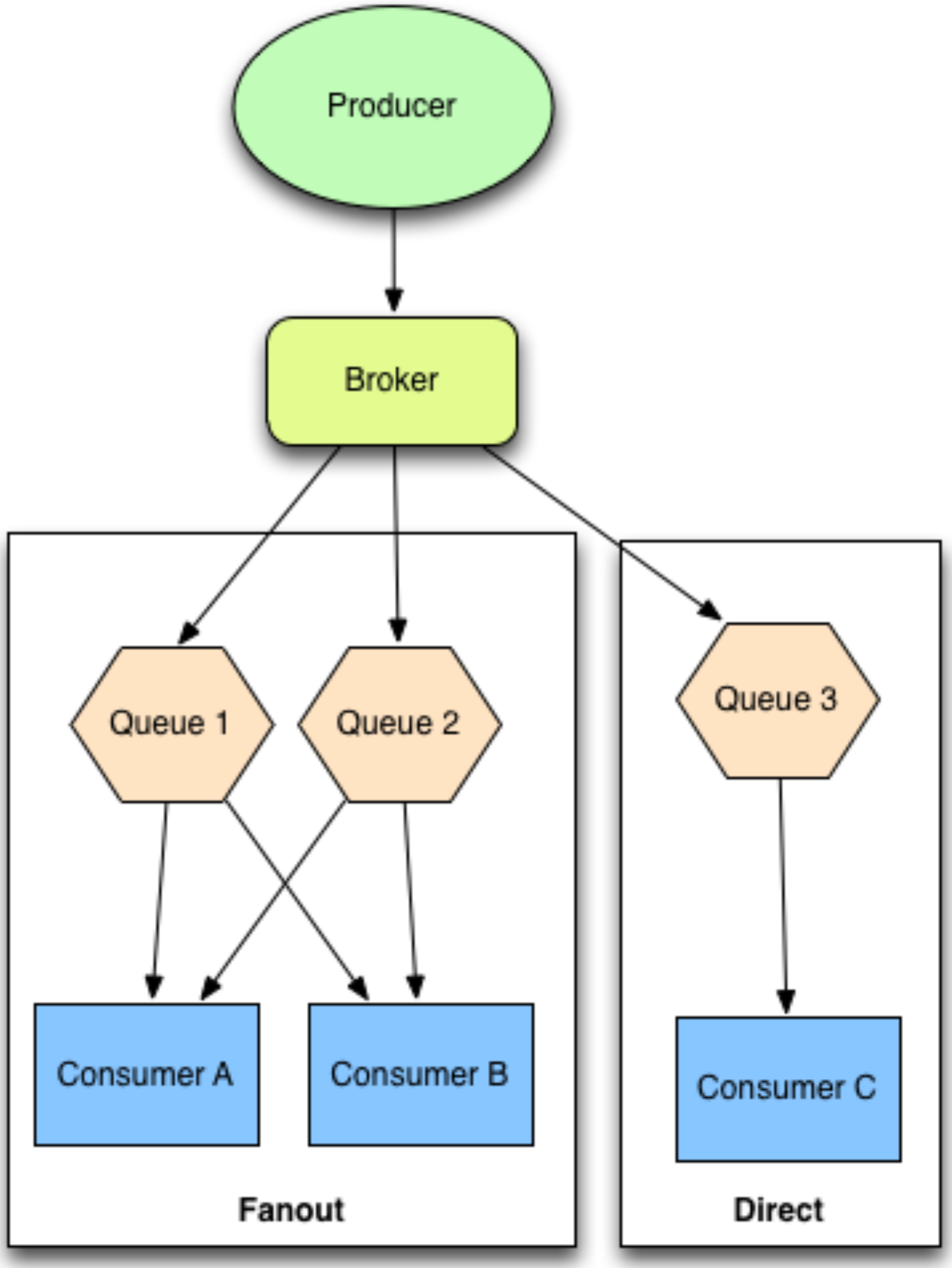
Лекция 15

- Очереди и для чего они нужны
- DelayedJob, Sidekiq и другие
- ActiveJob
- Whenever gem

Sync, async and queues


Client
Request
Response
Server


Client
Request
Response
Application
Instruction
Instruction
Instruction
Asynchronous Processor
Instructions Sent Here
Instructions are Processed


Producer
Broker
Queue 1
Queue 2
Queue 3
Consumer A
Consumer B
Consumer C
Fanout
Direct

Библиотеки

Delayed Job

https://github.com/collectiveidea/delayed_job

Delayed Job

```
# without delayed_job
@user.activate!(@device)

# with delayed_job
@user.delay.activate!(@device)

# with delayed_job running at a specific time
Notifier.delay(run_at: 5.minutes.from_now).signup(@user)
```

Delayed Job

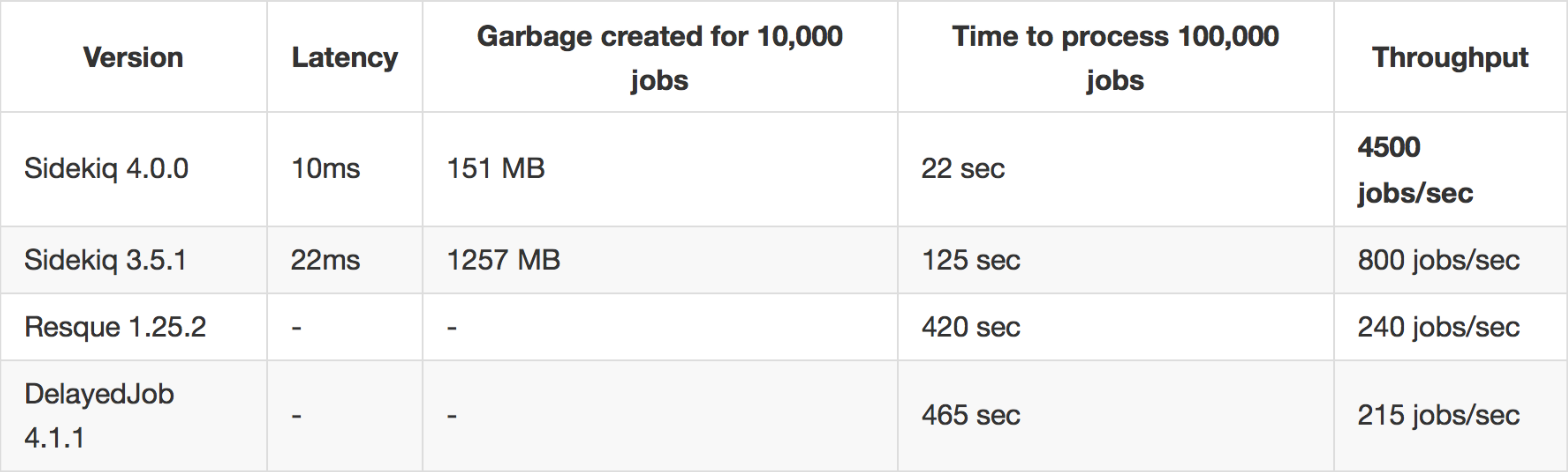
```
RAILS_ENV=production bin/delayed_job -n 2 start

RAILS_ENV=production bin/delayed_job stop
```

Sidekiq

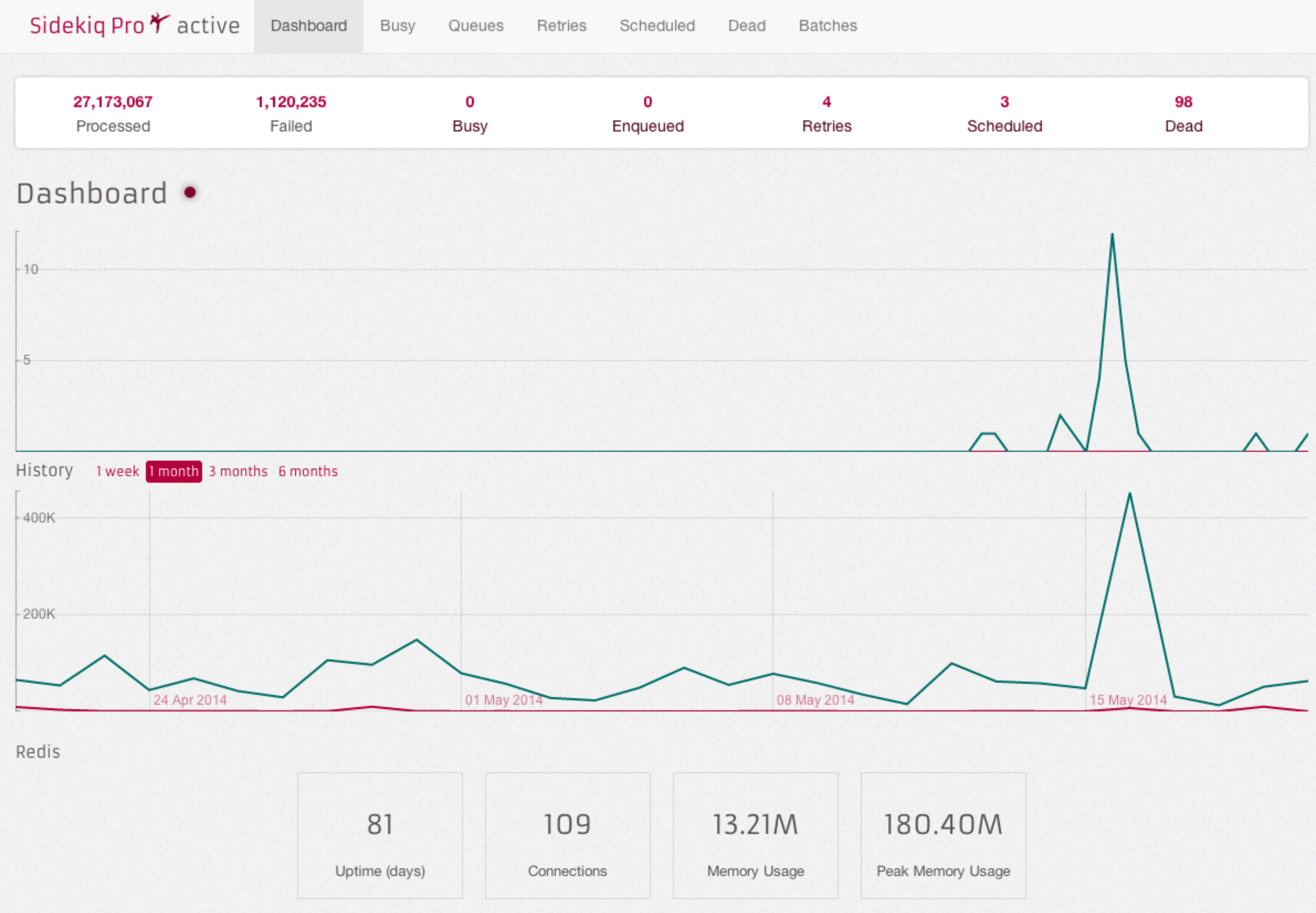
http://sidekiq.org

Sidekiq



Version	Latency	Garbage created for 10,000 jobs	Time to process 100,000 jobs	Throughput
Sidekiq 4.0.0	10ms	151 MB	22 sec	**4500 jobs/sec**
Sidekiq 3.5.1	22ms	1257 MB	125 sec	800 jobs/sec
Resque 1.25.2	-	-	420 sec	240 jobs/sec
DelayedJob 4.1.1	-	-	465 sec	215 jobs/sec

Sidekiq



Processed	Failed	Busy	Enqueued	Retries	Scheduled	Dead
27,173,067	1,120,235	0	0	4	3	98

Uptime (days)	Connections	Memory Usage	Peak Memory Usage
81	109	13.21M	180.40M

Sidekiq

```
class HardWorker
  include Sidekiq::Worker

  def perform(name, count)
    # do something
  end

end
```

Sidekiq

```
HardWorker.perform_async('bob', 5)
```

Sidekiq

```
bundle exec sidekiq
```

Resque

https://github.com/resque/resque

Active Job

```
$ rails g job HardWork
      invoke  test_unit
      create    test/jobs/hard_work_job_test.rb
      create  app/jobs/hard_work_job.rb
```

```
class HardWorkJob < ActiveJob::Base
  queue_as :default

  def perform(*args)
    # Do something later
  end
end
```

HardWorkJob.**perform_later** @something

```ruby
# config/application.rb
module YourApp
  class Application < Rails::Application
    # Be sure to have the adapter's gem in your Gemfile
    # and follow the adapter's specific installation
    # and deployment instructions.


    config.active_job.queue_adapter = :sidekiq
  end
end
```

Callbacks

- `before_enqueue`
- `around_enqueue`
- `after_enqueue`
- `before_perform`
- `around_perform`
- `after_perform`

Callbacks

```
class HardWorkJob < ApplicationJob
  queue_as :default

  before_enqueue do |job|
    # Do something with the job instance
  end

  around_perform do |job, block|
    # Do something before perform
    block.call
    # Do something after perform
  end

  def perform
    # Do something later
  end
end
```

Практика

Вопросы?

+7 (926) 889-16-32

alec@alec-c4.com

http://alec-c4.com/