Stupid Ideas for Many Computers

Aja Hammerly
No Good Ideas In This Talk
Aja Hammerly
http://github.com/thagomizer
@thagomizer_rb
http://www.thagomizer.com
All code is copyright Google and licensed Apache V2
Stupid
Tweets & Emoji
sentiment analysis (noun)
the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc., is positive, negative, or neutral.
Hard
"Sure, I'd love to"
<table>
<thead>
<tr>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>❤️</td>
<td>💩</td>
</tr>
<tr>
<td>👍</td>
<td>👿</td>
</tr>
<tr>
<td>😆</td>
<td>😈</td>
</tr>
</tbody>
</table>

@thagomizer_rb
Tweets
Code
SENTIMENTS = {
  ":-)" => 10,
  "😀" => 30,
  "❤" => 5,
  "(,:)" => 5,
  "👿" => -15,
  "::~D" => 15,
  "👍" => 7,
  "💩" => -30
}
def analyze tweet
    SENTIMENTS.each do |s, val|
        if tweet.include? s then
            sentiment += val
        end
    end
end
end
Many Computers!
linda
Tuple Space

[:sentiment, 1]

[:tweet, "text"]

[:cat, "Nick"]

Worker

Worker

Worker

Worker
Server
require 'rinda/tuplespace'

URI = ARGV[0] || "druby://0.0.0.0:61676"
DRb.start_service(URI, Rinda::TupleSpace.new)
DRb.thread.join
3 Worker Types
Server (Tuple Space)

Fetcher

Analyzer

Reducer
Server (Tuple Space)

[:tweet, "My :-)"]

Fetcher
Analyzer
Reducer
Server (Tuple Space)

[:tweet, "My :-)"]

Fetcher
Analyzer
Reducer
Fetcher  Analyzer  Reducer
[:tweet, "My :-)"]
Server (Tuple Space)

Fetcher

[:,sentiment, 10]

Analyzer

Reducer
Server (Tuple Space)

Fetcher

Analyzer

Reducer

[\{:sentiment, 10\}]

@thagomizer_rb
Server (Tuple Space)

Fetcher
Analyzer
Reducer

total_sentiment = 10
Making It Go
Kubernetes
Kubernetes
(Duh!)
Server
FROM ruby:latest
RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app
COPY . /usr/src/app
EXPOSE 3000
CMD ["./rinda_server.rb", "druby://0.0.0.0:3000"]
Replication Controller
apiVersion: v1
kind: ReplicationController
metadata:
  labels:
    name: server
    name: server-controller
spec:
  replicas: 1
  selector:
    name: server
  template:
    metadata:
      labels:
        name: server
    spec:
      containers:
      - image: gcr.io/stupid_ideas/tweet_server:latest
        name: server
        ports:
        - port: 3000
          containerPort: 3000
          targetPort: 3000
apiVersion: v1
kind: ReplicationController
metadata:
  labels:
    name: server
    name: server-controller
spec:
  replicas: 1
selector:
  name: server
template:
  metadata:
    labels:
      name: server
  spec:
    containers:
    - image: gcr.io/stupid_ideas/tweet_server:latest
      name: server
      ports:
      - port: 3000
        containerPort: 3000
        targetPort: 3000
Service
apiVersion: v1
class: Service
metadata:
  name: server
  labels:
    name: server
spec:
  type: LoadBalancer
  ports:
    - port: 3000
      targetPort: 3000
      protocol: TCP
  selector:
    name: server
Fetcher
FROM ruby:latest
RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app
COPY . /usr/src/app
EXPOSE 3000
CMD ["./rinda_fetcher.rb"]
Pod
apiVersion: v1
kind: Pod
metadata:
  labels:
    name: fetcher
    name: fetcher
spec:
  containers:
  - image: gcr.io/stupid_ideas/tweet_fetcher:latest
    name: fetcher
Analyzer
FROM ruby:latest
RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app

COPY . /usr/src/app

EXPOSE 3000

CMD ["./rinda_analyzer.rb"]
Replication Controller
apiVersion: v1
kind: ReplicationController
metadata:
  labels:
    name: analyzer
    name: analyzer-controller
spec:
  replicas: 5
  selector:
    name: analyzer
template:
  metadata:
    labels:
      name: analyzer
  spec:
    containers:
    - image: gcr.io/stupid-ideas/tweet_analyzer:latest
      name: analyzer
Reducer
FROM ruby:latest
RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app
COPY . /usr/src/app
EXPOSE 3000
CMD ["./rinda_reducer.rb"]
Replication Controller
apiVersion: v1
kind: ReplicationController
metadata:
  labels:
    name: reducer
    name: reducer-controller
spec:
  replicas: 1
  selector:
    name: reducer
template:
  metadata:
    labels:
      name: reducer
  spec:
    containers:
    - image: gcr.io/stupid-ideas/tweet_reducer:latest
      name: reducer
Demo
#KubeConEmoji
Latin Squares
Latin Square

In combinatorics and in experimental design, a Latin square is an $n \times n$ array filled with $n$ different symbols, each occurring exactly once in each row and exactly once in each column.
Euler
Simple
6 x 6
812,851,200
9 x 9
5,524,751,496,156,892,842,531,225,600
\sim 5 \times 10^{27}
Code
Sieve
SIZE = ARGV[0].to_i || 6

# generate all permutations of size n
permutates = (1..SIZE).to_a.permutation.map { |n| n }

# generate all size n permutations of the permutations
permutates.permutation(SIZE).each do |grid|
  pp grid
end
SIZE = ARGV[0].to_i || 6

# generate all permutations of size n
permutates = (1..SIZE).to_a.permutation.map { |n| n }

# generate all size n permutations of the permutations
permutates.permutation(SIZE).each do |grid|
  pp grid
end
SIZE = ARGV[0].to_i || 6

# generate all permutations of size n
permutes = (1..SIZE).to_a.permutation.map { |n| n }

# generate all size n permutations of the permutations
permutes.permutation(SIZE).each do |grid|
  pp grid
end
[[1, 2, 3, 4, 5],
[1, 2, 3, 5, 4],
[1, 5, 4, 2, 3],
[5, 3, 4, 2, 1],
[3, 2, 1, 5, 4]]
[[1, 2, 3, 4, 5],
[1, 2, 3, 5, 4],
[1, 5, 4, 2, 3],
[5, 3, 4, 2, 1],
[3, 2, 1, 5, 4]]
(n!)! / (n! - n)!
$O(n!!)$
\[ \sim 1 \times 10^{50} \]
size = solution[0].length
example = (1..size).to_a

correct = solution.all? { |row| row.sort == example } &&
          solution.transpose.all? { |col| col.sort == example }

if correct then
  pp solution
end
$10^{50}$
SCALE!!!!
Server
Generator
Checker
Deployment
Same As Before
Same Dockerfiles
Server
apiVersion: v1
kind: ReplicationController
metadata:
    labels:
        name: server
        name: server-controller
spec:
    replicas: 1
    selector:
        name: server
    template:
        metadata:
            labels:
                name: server
        spec:
            containers:
            - image: gcr.io/stupid_ideas/latin_server:latest
              name: server
              ports:
              - port: 3000
                containerPort: 3000
                targetPort: 3000
apiVersion: v1
kind: Service
metadata:
  name: server
  labels:
    name: server
spec:
  type: LoadBalancer
  ports:
    - port: 3000
      targetPort: 3000
      protocol: TCP
  selector:
    name: server
Generator
apiVersion: v1
kind: Pod
metadata:
    labels:
        name: generator
spec:
    containers:
        - image: gcr.io/stupid_ideas/
latin_generator:latest
        name: generator
        restartPolicy: OnFailure
Checker
apiVersion: v1
kind: ReplicationController
metadata:
  labels:
    name: checker
    name: checker-controller
spec:
  replicas: 15
  selector:
    name: checker
template:
  metadata:
    labels:
      name: checker
  spec:
    containers:
      - image: gcr.io/stupid-ideas/latin_checker:latest
        name: checker
10 Nodes
Profound Thoughts
Useful
Correct Tool
Correct Tool
Fun
Yes
Fast
Power
Computer?
Limits
Distributed Systems Are Fun & Easy
Fun
Thank You