Distributed Ramsey Search

Team : Party of 4
Debjani, Sachin, Sahaj, Varun

Ramsey Search App Orchestration

- Fabric is a python library for streamlining the use of SSH deployment or sys admin tasks
- It makes execution of shell commands over SSH easy and pythonic.
- Our aim was to stop administering our environment and start developing it.

To start our cloud environment for ramsey_search now we run-

‘fab start_client --set f=client.json’

‘fab start_server --set f=server.json’

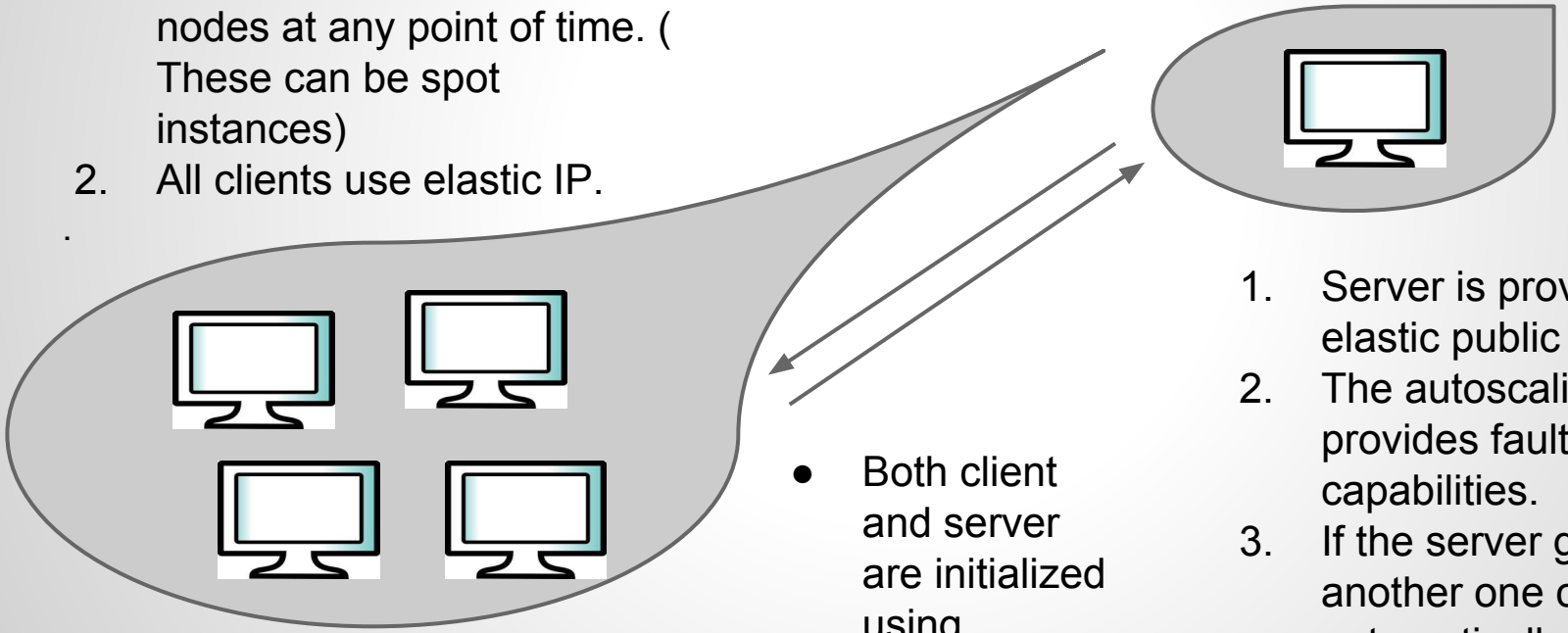
Architecture

- Each cluster/partition uses a different algorithm in its pursuit for the highest ramsey counter example

1. Autoscaling ensures that there are atleast 4 client nodes at any point of time. (These can be spot instances)
2. All clients use elastic IP.

.

Server autoscale group (Size=1)

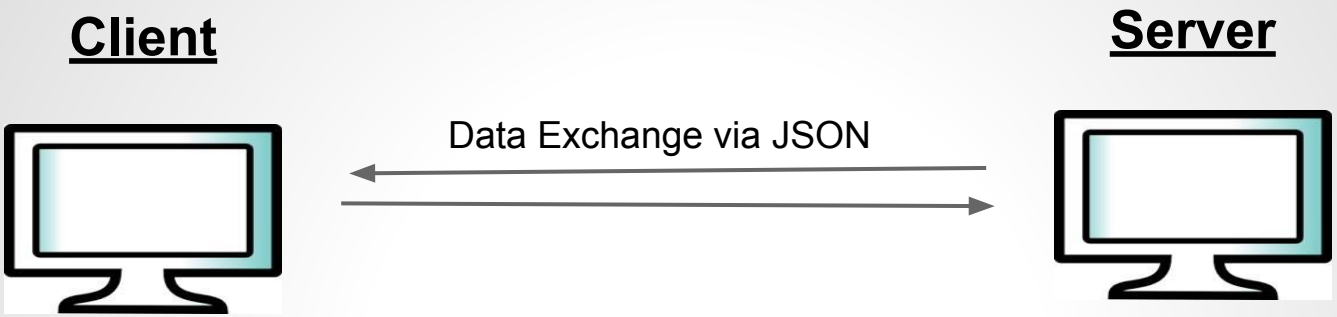


Client autoscale group (Size=4)

1. Server is provided with elastic public IP.
2. The autoscaling facility provides fault tolerant capabilities.
3. If the server goes down, another one comes up automatically.

- Both client and server are initialized using **cloud-init scripts.**

Multi-threaded Client-Server



1. Client starts the ramsey_search and spawns 3 threads.
2. Send Counterexamples to server.
3. Send updates to server.
4. Listens for broadcast messages from server and acts accordingly.

1.Server listens to multiple clients and processes different message types.
2. It intelligently decides the best graph/update and broadcasts them to all clients.

Languages Used: Python and C

The Big Picture

Internet Facing Load Balancer (Azure)

Simulated Annealing [Eucalyptus]

Dynamo DB

BOTO API

Application Servers in a load balanced set

Multi-flip (2 @ a time)

[Azure]

Multi-flip (3 @ a time)

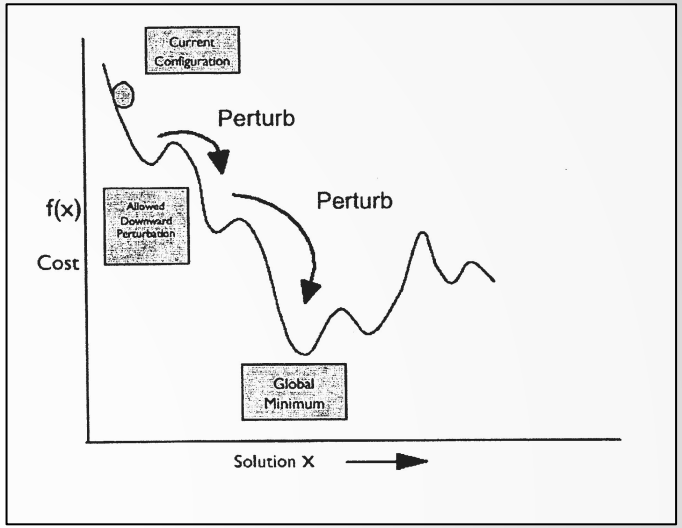
[AWS]

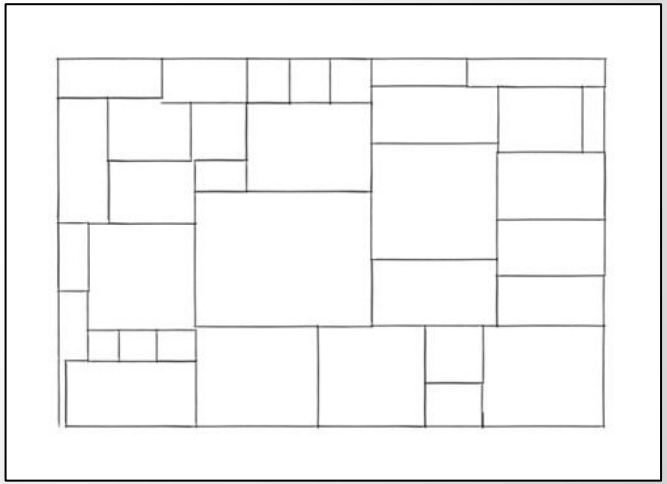
Genetic Algorithm [Cloud Lab]

**Not extensively used

Algorithms Used

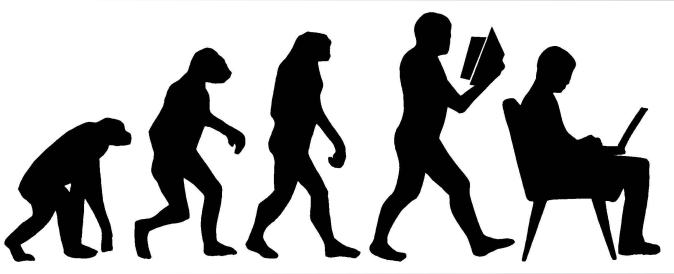
- Simulated Annealing
- Generating random seeds
- Paley Graph (101)
- Combining smaller CEs





Algorithms Used

- ➔ Multi-flip taboo search
- ➔ Genetic Algorithm



- ➔ Exploring only ‘vertex-symmetric’ graphs
- ➔ Combining paley graphs
- ➔ Shared Taboo list with pthreads.

Learning Experience

- Explored various cloud features such as High **Availability**, **Load balancing**, **Autoscaling**, **Availability sets**, **Virtual networking** across different IaaS providers.
- Multithreading in Python and Global Interpreter Lock
- Integrating **Condor** with cloud model was difficult because of frequent checkpointing reasons. We used it as an independent compute partition.
- **Xsede**, though powerful did not have a predictable job scheduling mechanism. Hence integration with cloud was not feasible.
- It is not hard to quickly exceed the cost in cloud environment!

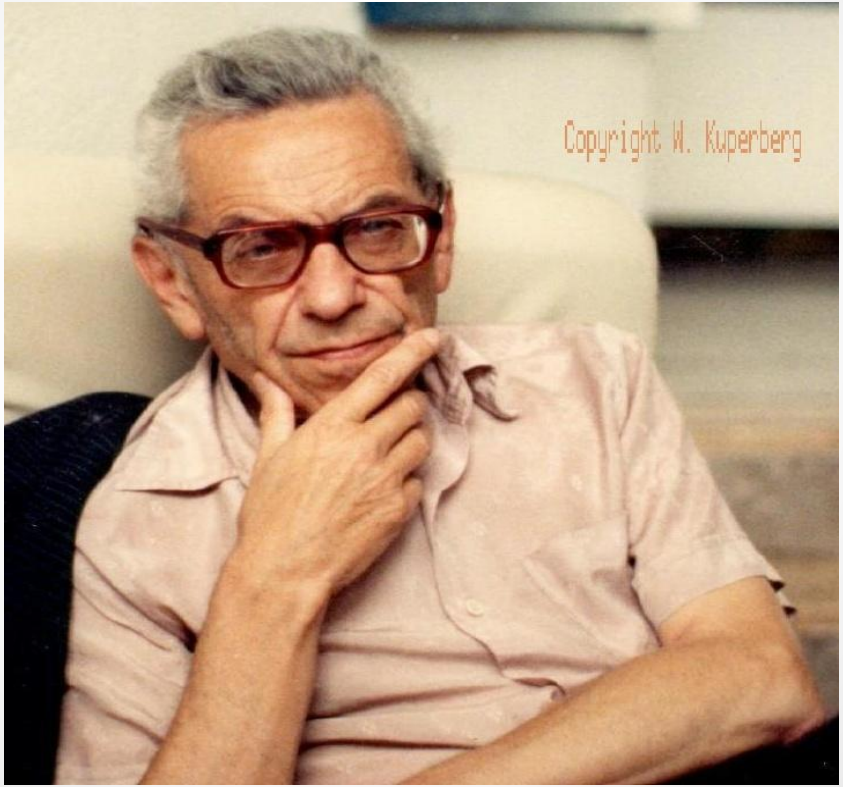
Platforms Used

- AWS [100 % of quota + beyond]
- Azure [90 % of quota]
- Eucalyptus [4 VM instances]
- Condor
- Xsede
- CloudLab (Sparingly)

Languages used:

- C
- Python
- Ruby on Rails

Ramsey Counter Example ...



204!!!

Thank You