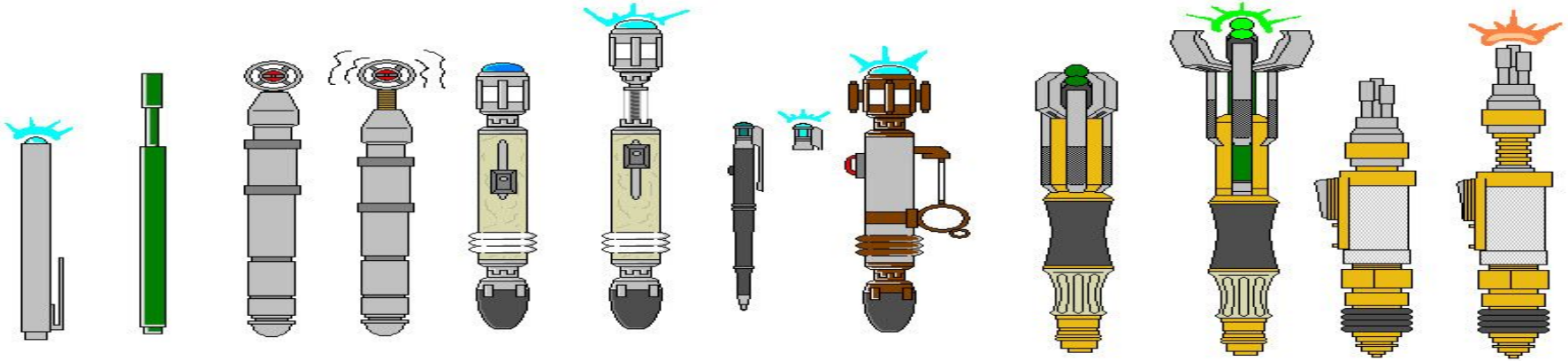


Automated Fault-Tolerance Testing

Ajay Vaddadi, Groupon

April 11th, 2016

About Me and My Team

- Lead Internal Tools Developer @ Groupon.
- Employed with Groupon for 4 years.
- Built various Internal tools ranging from Test Automation Frameworks to Continuous Performance and Deployment Tools (Story for another time …)
- My Awesome Team -
  - Adithya Nagarajan (Team Manager and Co-Author on the Paper)
  - Amrutha Badami
  - Kavin Arasu

Fault Tolerance - What is it and why is it important ?

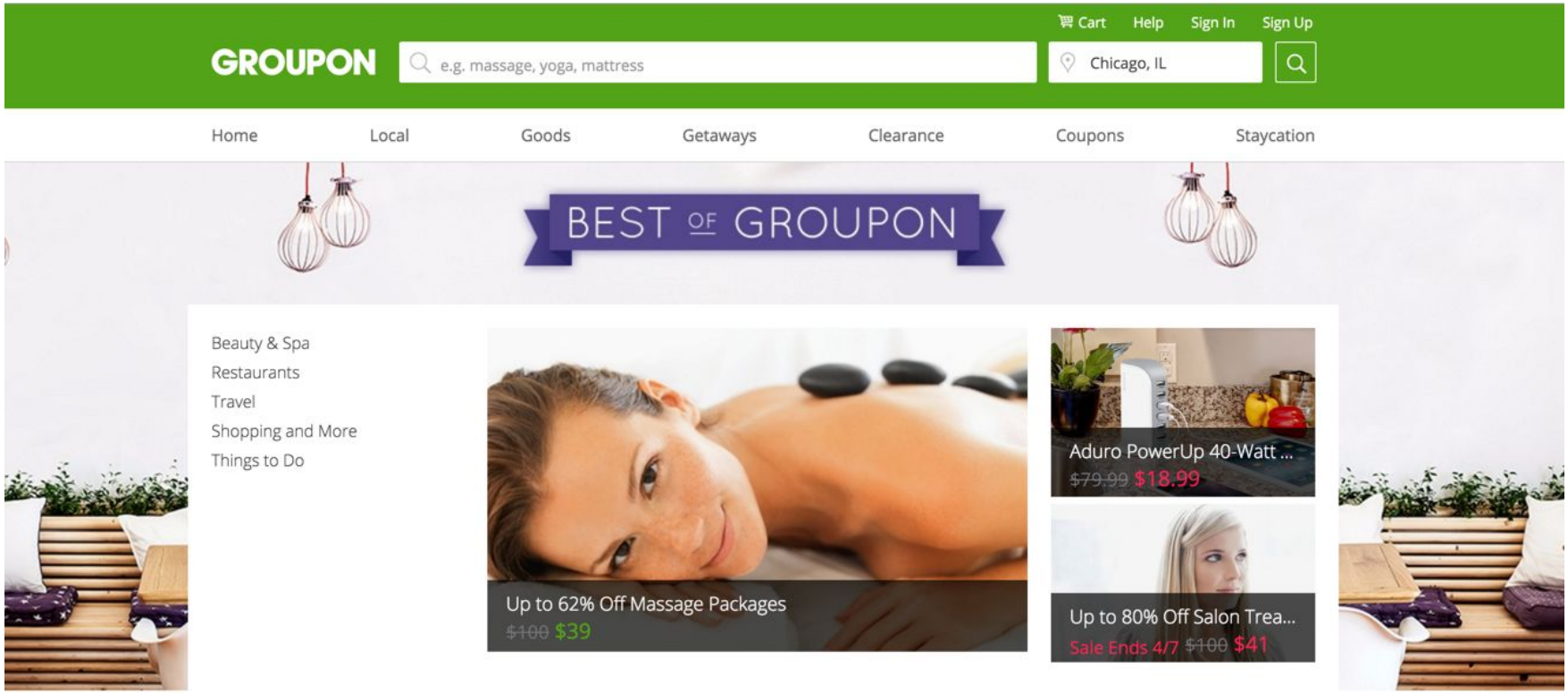
- Fault Tolerance is an ability of computer software to continue its normal operation despite the presence of system or hardware faults.

- Fault in Software terms can vary from high latency from dependent or sub-systems to complete failure of the System under test (SUT).

- Fault injection can easily be simulated and overall environment behaviour can be observed to make changes in design to be better tolerant to the failure.

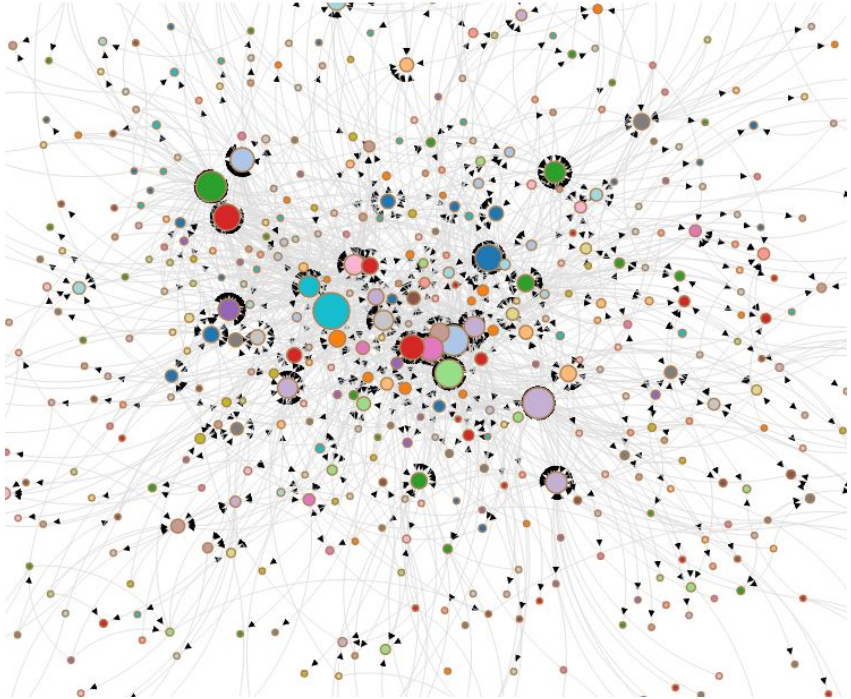
- Many real life tragedies would have been avoided if were tested for fault tolerance -
  - **Columbia Space Shuttle Disaster** - Small crack in heat shield
  - **ATT Complete Network Failure (1990)** - SPOF switch failure.

About Groupon ...



Engineering Scale @ Groupon

- **Microservices** Based Architecture
- **500+ Production Services**
- **In-house, Self-Managed Data Centres** spread across NA, Europe & APAC
- Complex Interdependencies between Services



What are the Requirements ?

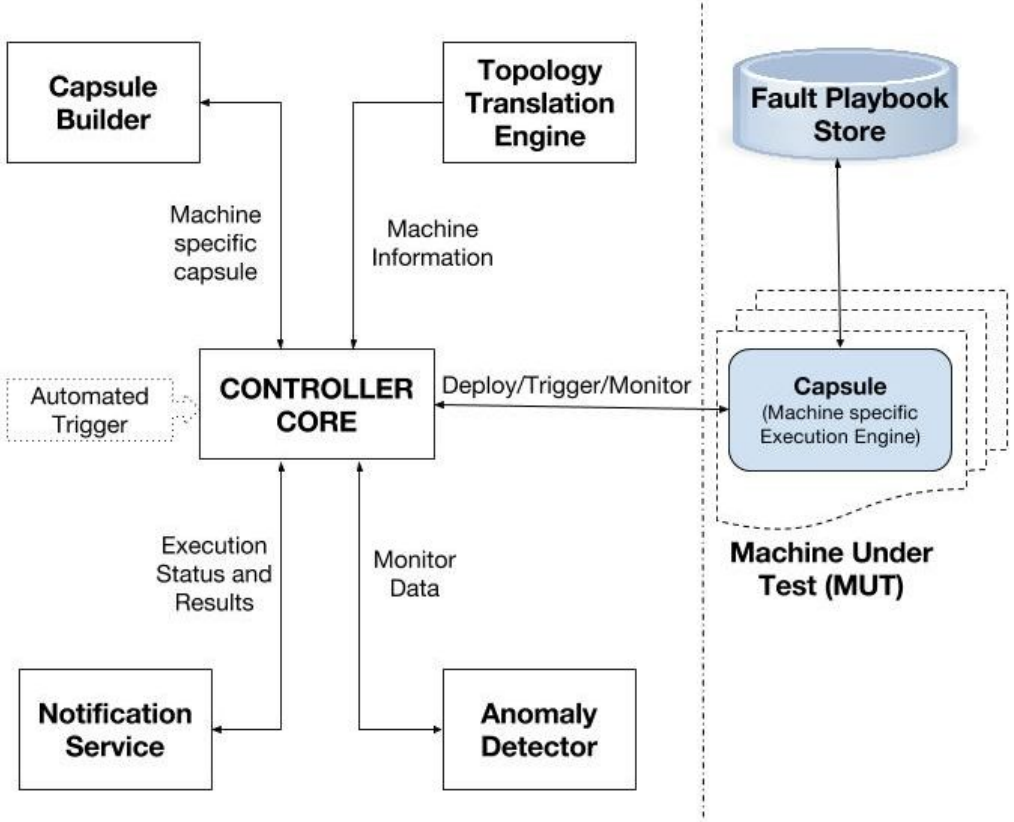
- Ability to Simulate Failures and Identify resiliency issues within the Groupon Architecture
- Ability to understand the Failure propagation pattern when a failure occurs.
- Ability to understand service architecture to better identify and group faults on a unique set of application servers/apps/machines.
- Ability to continuously monitor critical metrics during the faults on the SUT and dependent Services and terminate when anomaly is detected.
- Ability to recover the System under Test to pre-fault state.

Available Tools : *Are we reinventing the wheel ?*

What is already done ?

- Companies like Google, Amazon, Twitter and Netflix conduct fault tolerance testing on their platform.
- NETFLIX was one of the early players of in this Fault testing domain by developing widely popular CHAOS MONKEY / SIMIAN ARMY
- Other tools such as Varien, Pantera are mocked versions of SIMIAN ARMY developed in other ecosystems like python.
- Either very company specific and are not reusable OR highly geared towards ASG (Auto Scaling Groups) based Infra such as AWS etc.

ScrewDriver - Our Proposed Solution ..



Controller (and it's Friends)

Controller is a combination of 4 major modules focussing on 3 Key aspects - Building and Deploying Capsule, Injecting Fault(Core), Monitoring the Fault(Anomaly Detector).

How does it work?

- **Capsule Builder -** Generates time based expirable, machine specific capsule/container to be deployed on the machines specified by Topology service.
- **Controller Core** - Starts/Stops the fault/capsule with proper authentication. During fault execution works with Anomaly detector using Publisher/Subscriber model.
- **Anomaly Detector -** Interacts with Metric Client and identifies Anomalies. If found - aborts the current fault execution. Sometimes, Static Threshold checks are not enough to identify anomalies and need to delve into 2nd and 3rd derivatives.
- **Notification Module -** Communicates the status about the fault execution and post-execution report to the service owners.

Topology Translation Service

Service to parse and understand the building blocks (components/layers) of any service. Picks the machines and the corresponding faults to be executed and identifies the dependencies for the given service to monitor.

How does it work?

- Consumes topology YAML file for a given service and identifies underlying components/layers, machines, monitors and dependencies for the service.
- Information is pushed to GraphDB(Neo4j) for persistent storage.
- GraphDB allows us to identify dependency chain between systems without complex joins.
- Identifies list of machines for fault to be executed based on various selection criteria like - Traffic %, Rack, Colo, Random etc.

Capsule : *Self Contained Execution Engine.*

Self contained module which acts as an agent and executes faults on the Machine under Test.

How does it work?

- Deployed on demand to the machines under test. Self contained entity on its own, only dependent on Tower for basic instructions
- Bootstrap checks ensures that the capsule was not tampered with, only used on the machines it was signed for and also expires after a specific TTL.
- All the actions performed by Capsule will ensure no system will go into unresponsive state and will automatically trigger fault termination and system restore to bring back the system to healthy state as soon as possible.
- Executes the faults based on the Ansible styled - Fault Playbook (user defined steps).

Milestones and Roadmap

- Basic Prototype was completed in January 2016 and it showed promising results.
- Plan to run this tool extensively against Groupon services over next few months (Q2 2016).
- Plan to publish a follow up article with Detailed Implementation, Learnings and Issues Faced by Q3 2016.
- Long Term goal to Open Source the tool and evangelize for adaptation across industry.

Questions?