gridcontrol

Networked Process Managers

Who am I?

- CEO & Founder at Keymetrics.io
- Author of PM2
- Drone Racer
- Technologist

Let’s talk about Process Manager


EPM2

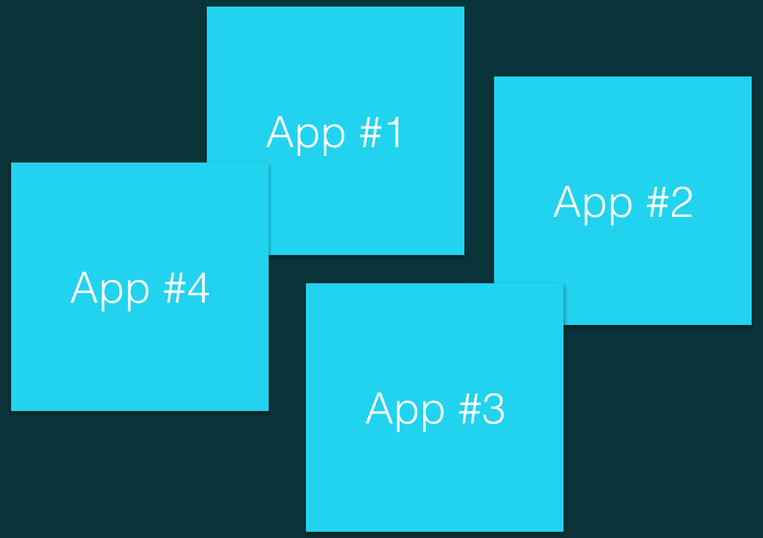
Process Manager

with a **Built-in load balancer**

that **Guarantees Uptime**

and **help you manage your processes**

Before



After

Disorganized application & workflow

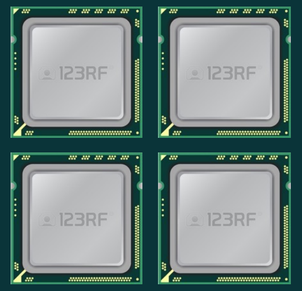
To

Structured application & workflow

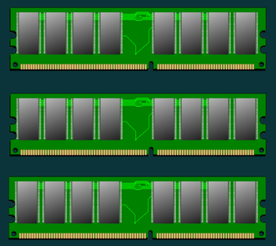
Ressources management and optimization

Extend capabilities

CPU



Memory



Modules

App #1

CL1 CL2 CL3 CL4

App #3

App #4

PM2

Usability

Log

Deployment

Easier software management

PM2: Some figures

- **14.000+** stars on GitHub
- **12.000.000+** downloads
- **84th** most popular JS project*
- **120+** contributors
- **1000+** tests

* **http://stats.js.org/**

EPM2


Frankfurt Am Main
Omsk
Albany
Mountain View
San Antonio
Gold
Central District
Dhaka
Jubail
San Pedro
Chennai
Medellín
Singapore
Male
Jakarta
Lima
Nairobi
Port Louis
São Paulo
Buenos Aires
Cape Town

PM2 CLI Overview

Install PM2:

```
$ npm install pm2 -g
```

Start an app:

```
$ pm2 start [app.js|process.json]
```

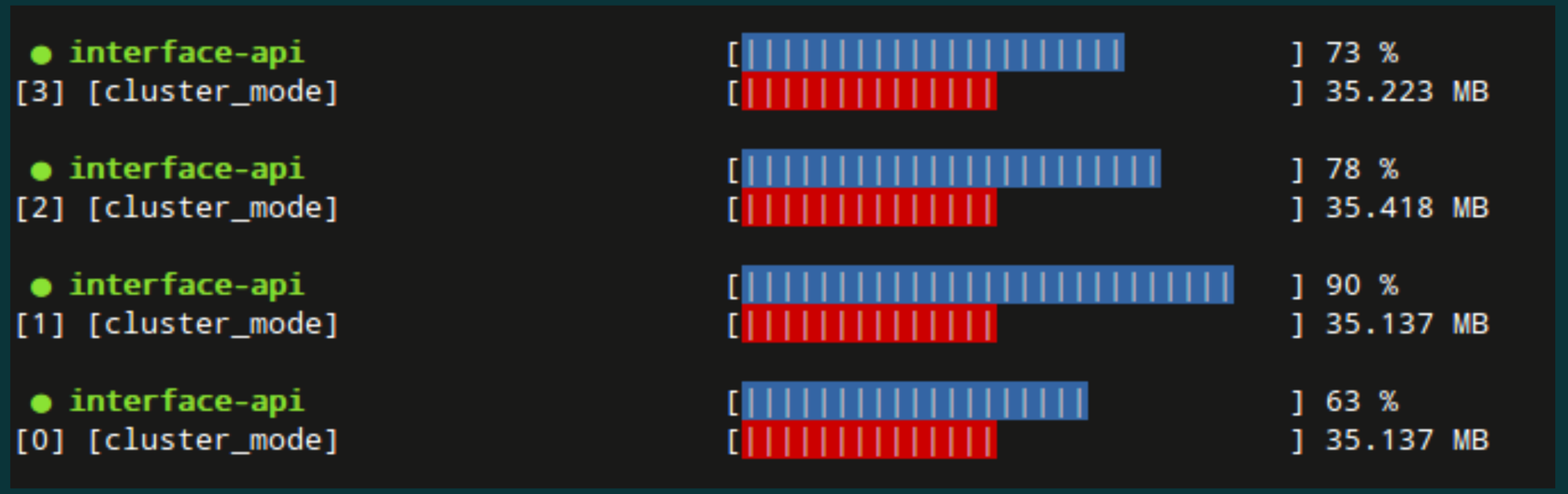
Start a Node.js app in cluster mode:

```
$ pm2 start app.js -i max
```

Starting an application cluster mode

```
$ pm2 start app.js -i <instances>
```

```
$ pm2 reload app
```



http://pm2.keymetrics.io/docs/usage/cluster-mode/

JSON Configuration file

ecosystem.json

```
{
  "apps" : [{
    "name"                : "Application 1 - API",
    "script"              : "api.js",
    "exec_interpreter"    : "node",
    "exec_mode"           : "cluster_mode",
    "instances"           : 10,
    "log_date_format"     : "YYYY-MM-DD HH:mm Z",
    "max_memory_restart"  : "160",
    "node_args"           : "--harmony",
    "ignore_watch"         : ["[\\/\\\\]\\./", "log"],
    "watch"               : true,
    "env" : {
      PORT : 3020
    }
  },{
    "name"                : "Application 2 - Worker",
    "script"              : "worker.js",
    "exec_mode"           : "fork_mode",
    "log_date_format"     : "YYYY-MM-DD HH:mm Z"
  },{
    "name"                : "Application 3 - Crawler",
    "script"              : "crawler.js",
    "exec_mode"           : "fork_mode",
    "log_date_format"     : "YYYY-MM-DD HH:mm Z"
  }]
}
```

Application 1 (API)

Application 2 (Worker)

Application 3 (Crawler)

PM2 V2 in mid September

- **4 months of testing**
- Optimized CPU/Memory usage
- Better integration with Docker
- PM2 API
- New logs commands (JSON/Formated)
- Windows Support

Official Website

http://pm2.io

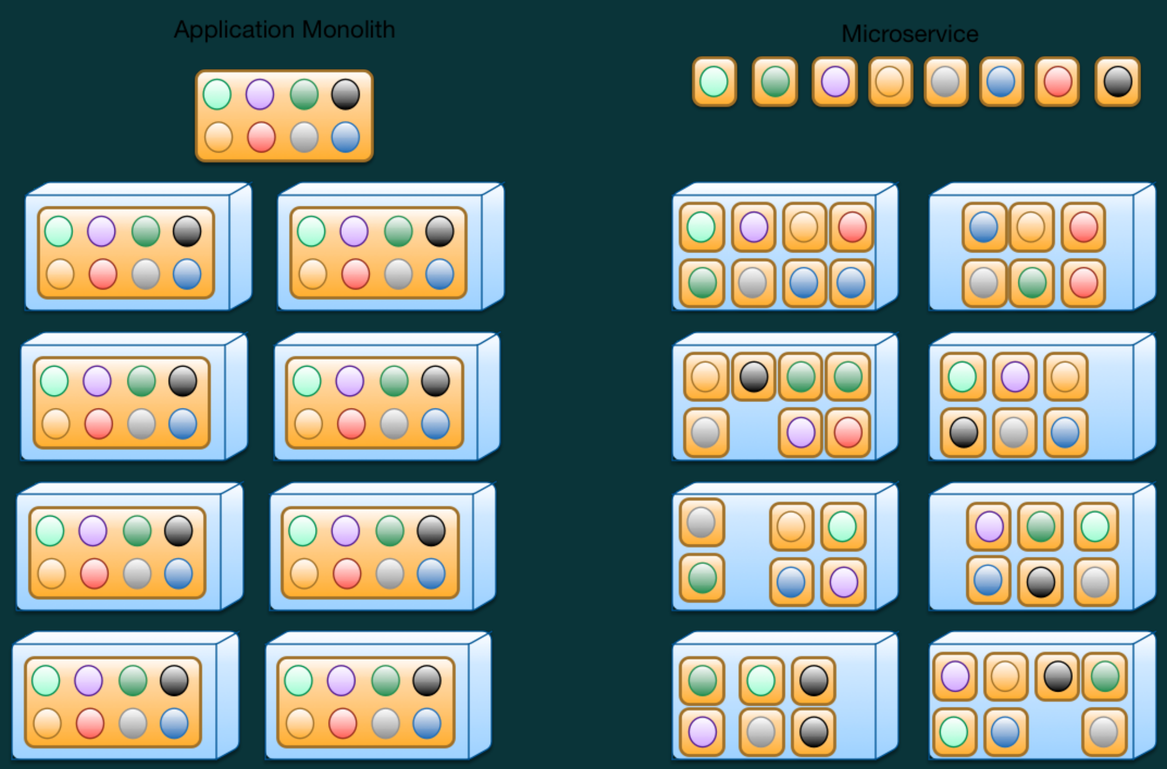
Github

http://github.com/Unitech/pm2

Micro services, Serverless and Networked Process Manager

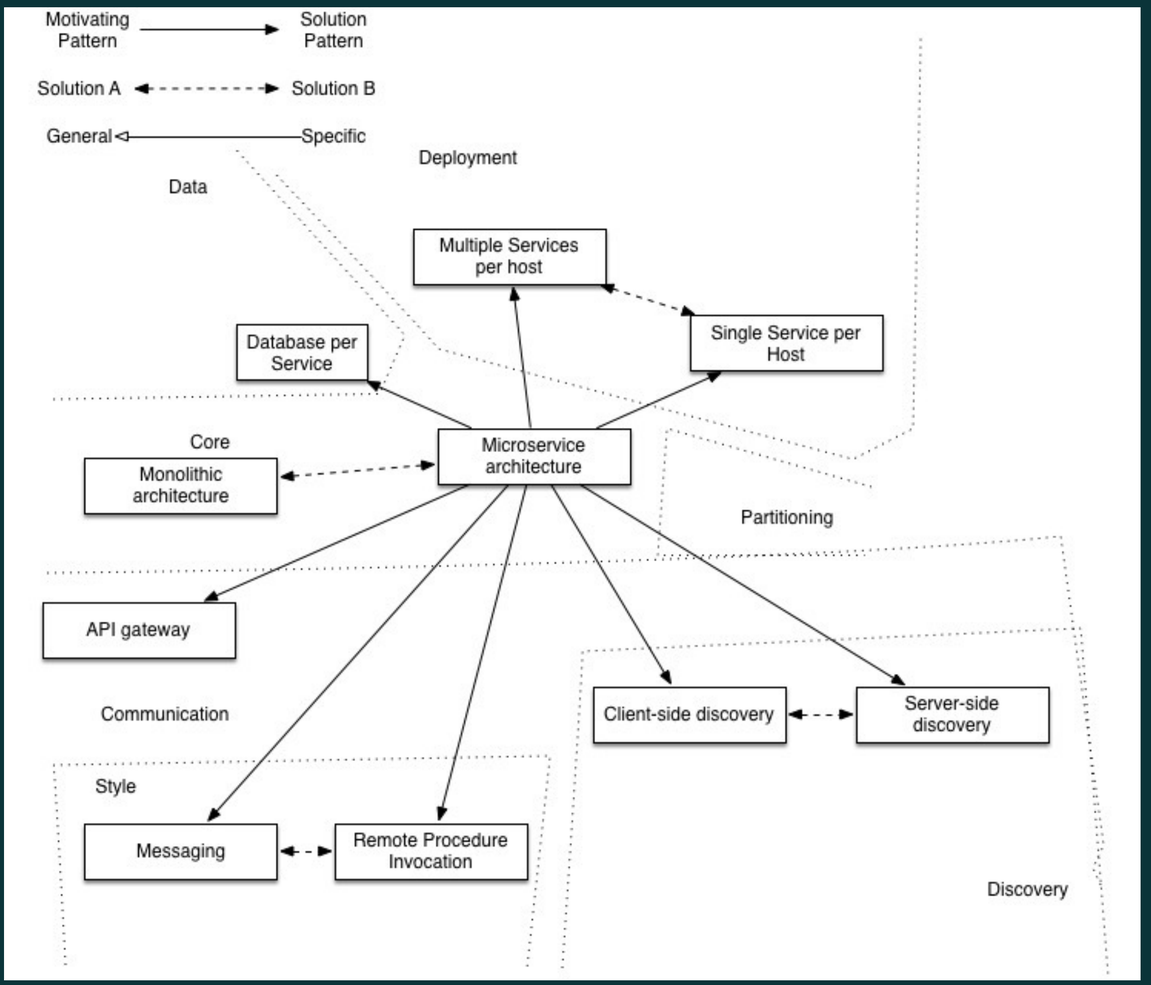
Theory

MicroService is HOT



- **Breaking Monolith into**
  - **Small, Independent, loosely coupled modules**
- **Better maintainability, monitoring and scalability**
- **Polyglot / Multi Language System**

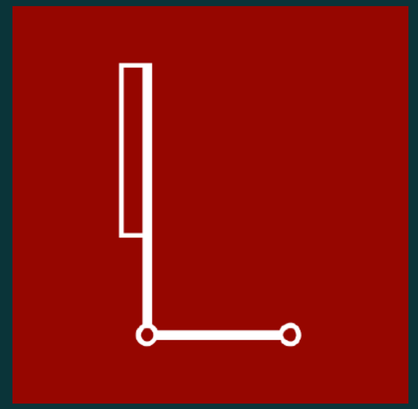
MicroService is HARD



- **Too many choices to do**
- **Difficulties for OSS framework to get attraction**
- **Lack of best-practices**
- **Deployment is complex**

MicroService is Disperse

- **Open Source Attempts**
  - LeverOS
  - Seneca
  - Effe
  - Klyng…
- **Commercial**
  - Amazon AWS Lambda
  - Google Cloud Functions
  - Microsoft Azure Functions



Not any OSS solutions get Adoption…

Infrastructure Provisioning, Management and Monitoring is the black whole on these systems

Commercial Solutions get Adoption

Commercial Solve the Problem of Infrastructure management

The case of Amazon Lambda: API

Expose a task in the "cloud":

```
exports.myHandler = (event, context, cb) => {
  // Use cb() and return information to the caller
};
```

Execute the task from your script:

```
lambda.invoke(params, function(err, data) {
  // Invoke the function
});
```

Where is the Ruby on Rails of MicroService?

Open Source, Conventions over Configuration,
Ease of Use, Human Friendly

Here is my Tentative

Making Process Managers connected togethers

Process Manager

Process Manager

Networked Process Manager

Process Manager

Network Layer

Service Discovery

Distributed File System

Distributed Messaging


gridcontrol

Decentralized Grid System

that **Provision** and **Link Server** to create a Compute Grid,

Replicate Task Files across all servers

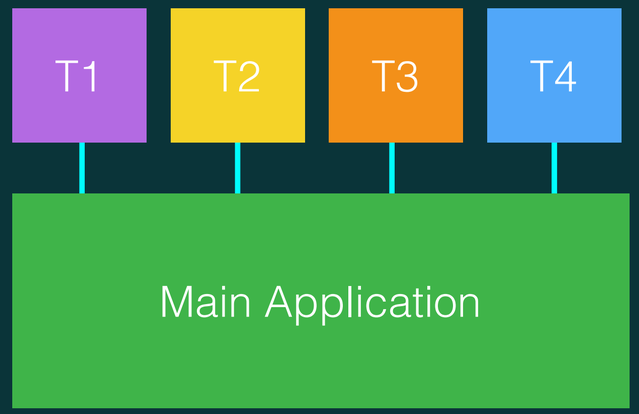
And Allow you to **Execute Tasks Remotely**

gridcontrol

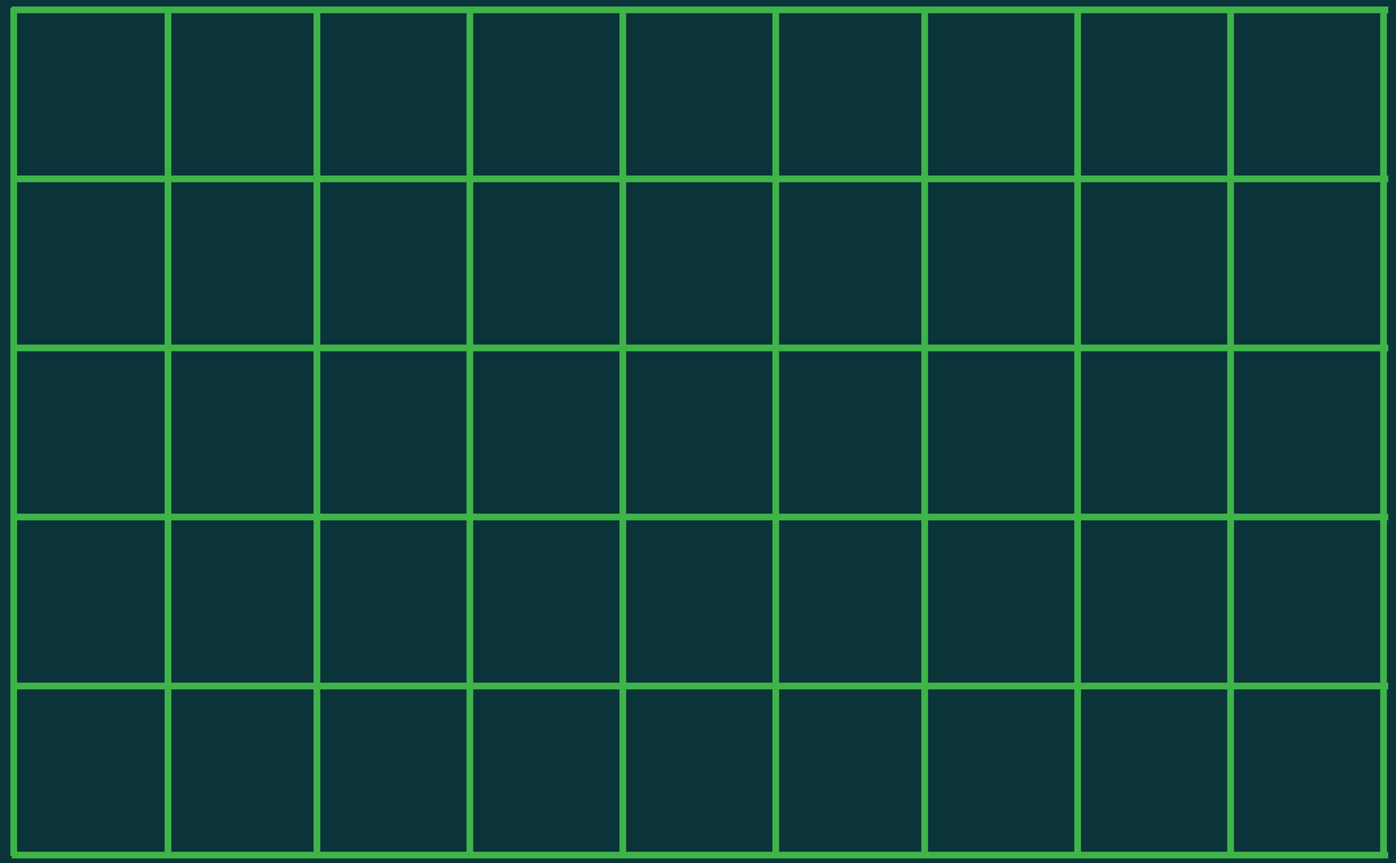
Local development

Tasks File

Main App

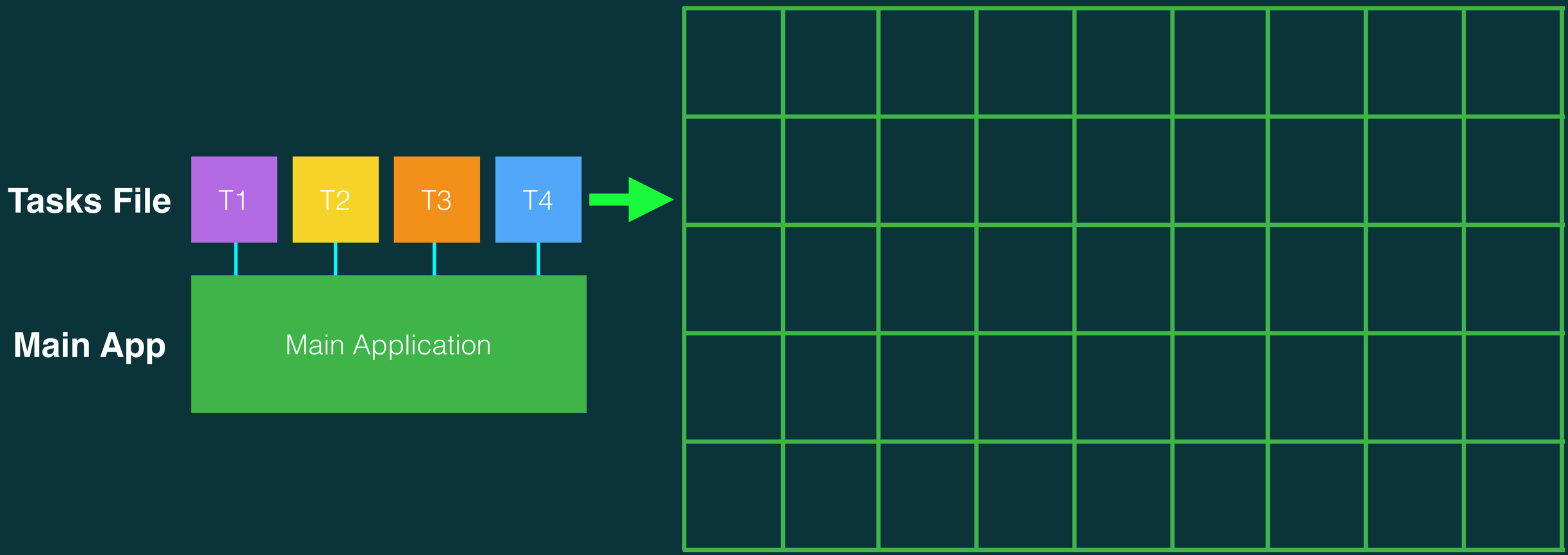


Compute Grid



gridcontrol

Deploy Local Tasks to the Grid



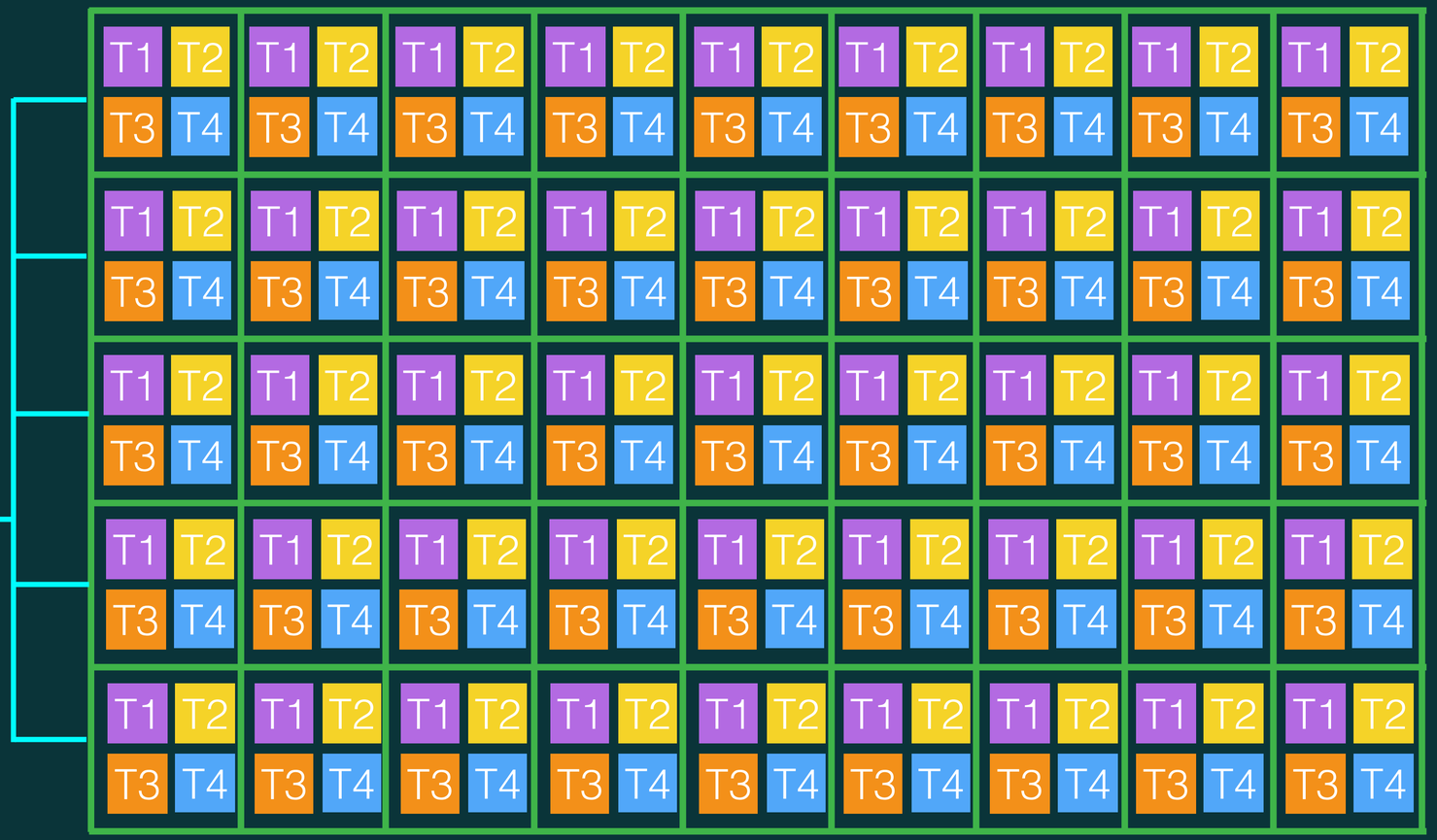
gridcontrol

Tasks Executed on Grid

Now Trigger Tasks in the Compute Grid

Main App

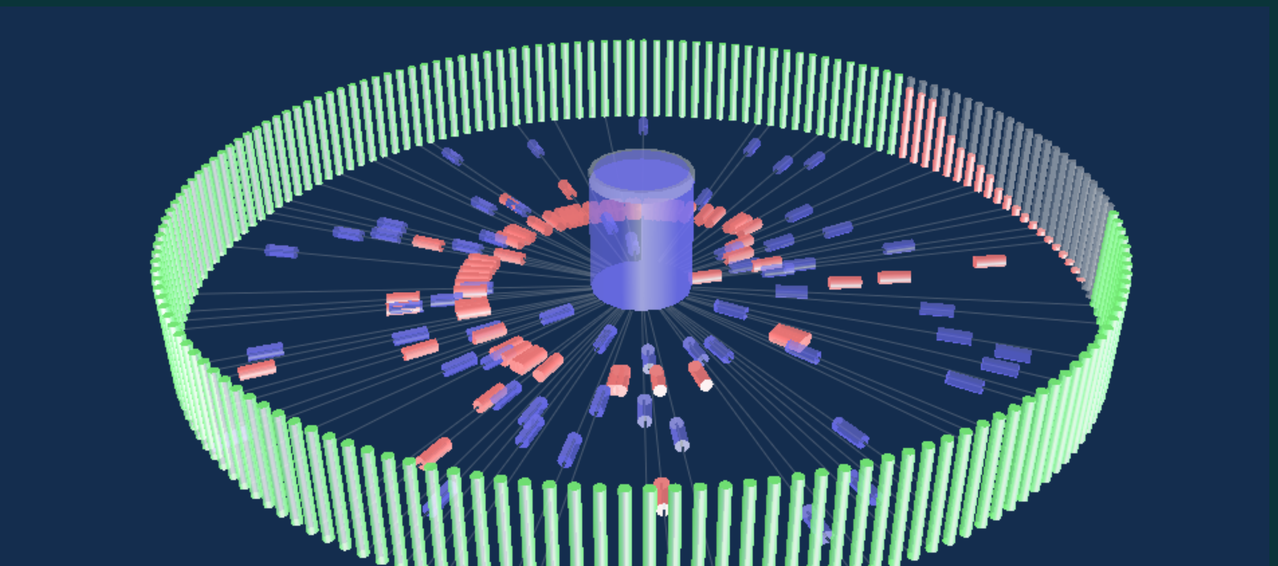
Main Application



Gridcontrol: Under the Hood

Service Discovery & Interconnectivity

- Multi DNS resolution + DHT
- Wide Area Discovery (across subnet)



Service Discovery

Distributed File System

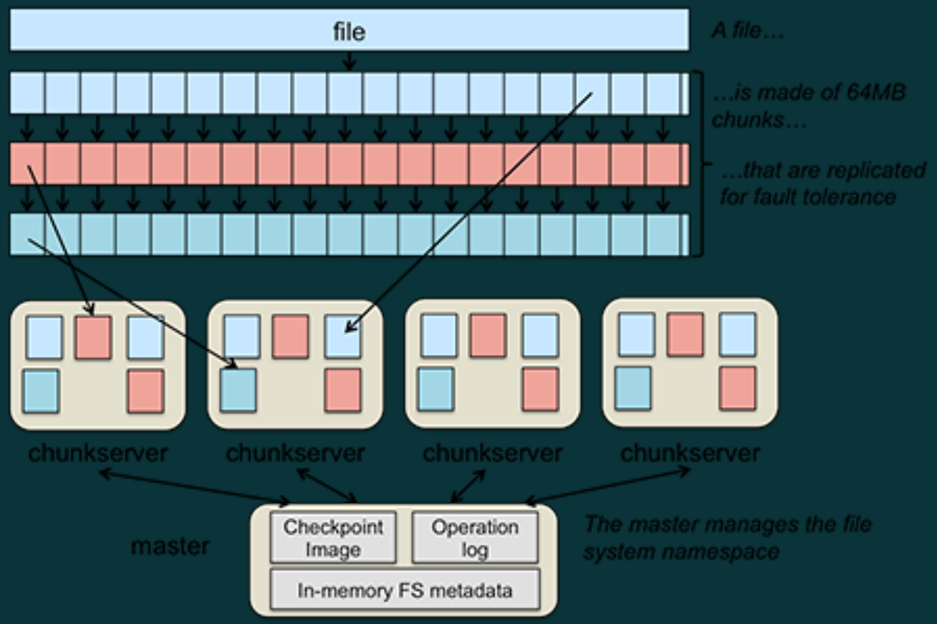
Distributed Messaging

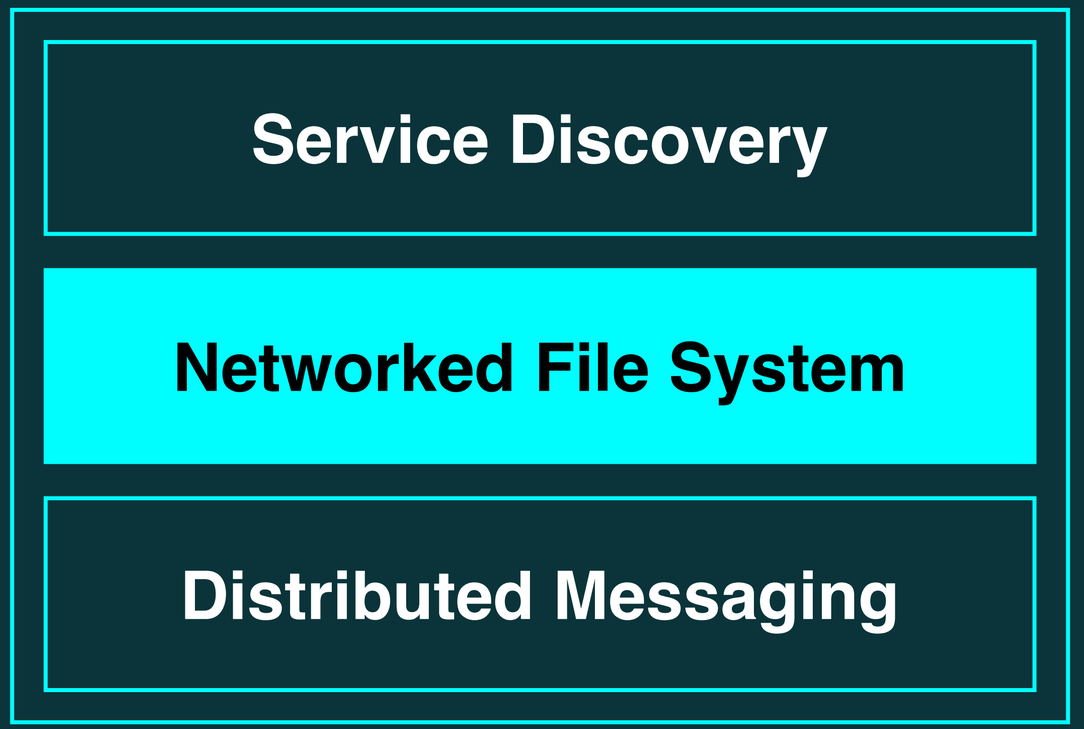
Decentralized!

Gridcontrol: Under the Hood

Networked File System

- Peer to Peer file Exchange
- Rabin File Chunking and Merkle Tree

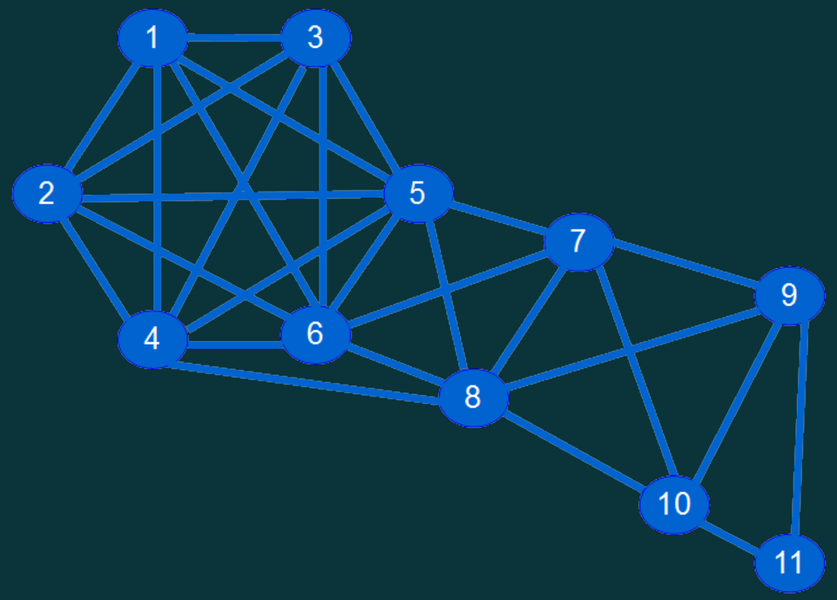


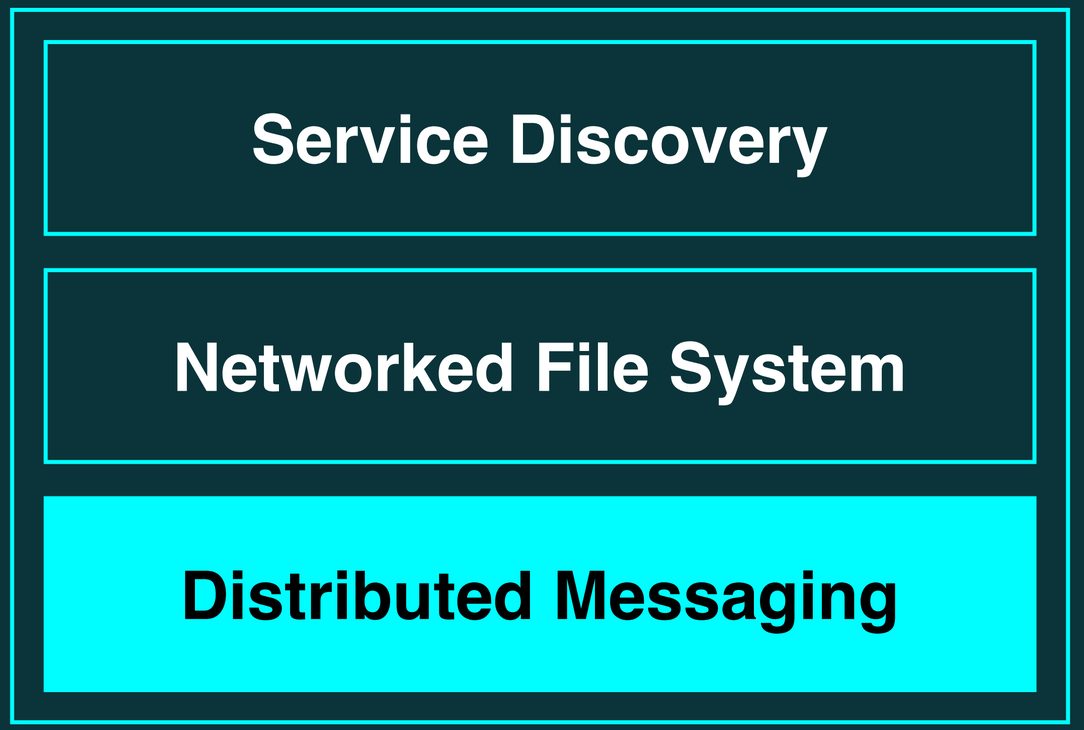


Gridcontrol: Under the Hood

Distributed Messaging

- Mesh Network
- Opinionated PUB/SUB system
- HTTTP over TCP internal protocol





1# Setup a Grid

Setup a Grid: Install CLI

Install Grid CLI:

```
$ npm install grid-cli -g
```

gridcontrol

Setup a Grid: Gridfile

Generate a Gridfile

$ grid new

“It's like an access card for servers that can be shared with your team"

```
>>> cat /tmp/Gridfile
#
# Gridfile Server Access ID Card
#

grid_name     = 'grid-test-1'
#grid_password =
```

Grid name + password

```
#keymetrics_public =
#keymetrics_secret =

servers = [
  'root@212.47.227.158',
  'root@163.172.151.155',
  'root@163.172.145.241'
]
```

Server Access List

```
ssh_key = '''
-----BEGIN RSA PRIVATE KEY-----
MIICWwIBAAKBgQCrpmznGC93PD0fb8ZGSTeu6G8vZf/2OdJUk+znfg8BxXnj+XFJ
0PY6PS1JwLg1v91c1cR7Jhkq7WBJRfmdRxd5C8znpS8jEMNuege9qdHQlavYFSdk
ZTHaj4Aae282bWXHgidPgi+/2itT8FYEOLv5+gAYj1au5BZAxF7K0Wv6JQIDAQAB
AoGAPqpVm9Cyk39EeVOfOU3FrG9jGBTO65+UcBhJJwJYjDcCFv6cI2zMV9q5y/E1
zzmqoRfR0XvJOGjM/DM1u/uWT5tNl1pR0urMXH6XPljlQeWpO4Sj4yvsaO8Z9AtT
bNI6c/W8xzDTthRxqlY                                    QQDhfF5/j7Pdb8Nn
69mWMjvReFwIXlaBL5I                                    fkjbR/evJWAch9oJ
ftr6aw4JAkEAwuD+2Bo                                    /+On0WXzTbMlBmP
B7TvwNfW8vjPKgmtW9ysXw5JIHlVwsUSPQJAD3cnSBqEHKkAvwz3JI0XezuFHHHT
/xJTM2xutAH0yusazdP0a/Se57eL/jik4SasOpYKD/JBUFOlCcewn0JeeQJAIX9N
LtYk37/+6ZwrWS88+Wgi38VBpkqee35jl4PU/xmWQ0b1GUvfYwK951bycqdD7MTn
EJA47Mnm1fclP6piPQJACtVfNwe2HQslGyA44ZK4mozisdjWugDm/cHeoRGytN9i
+96+dMFQnb3uQSYp+BQCqkwLS+DocVpLDJjCI6304g==
-----END RSA PRIVATE KEY-----
'''
```

SSH PRIVATE KEY

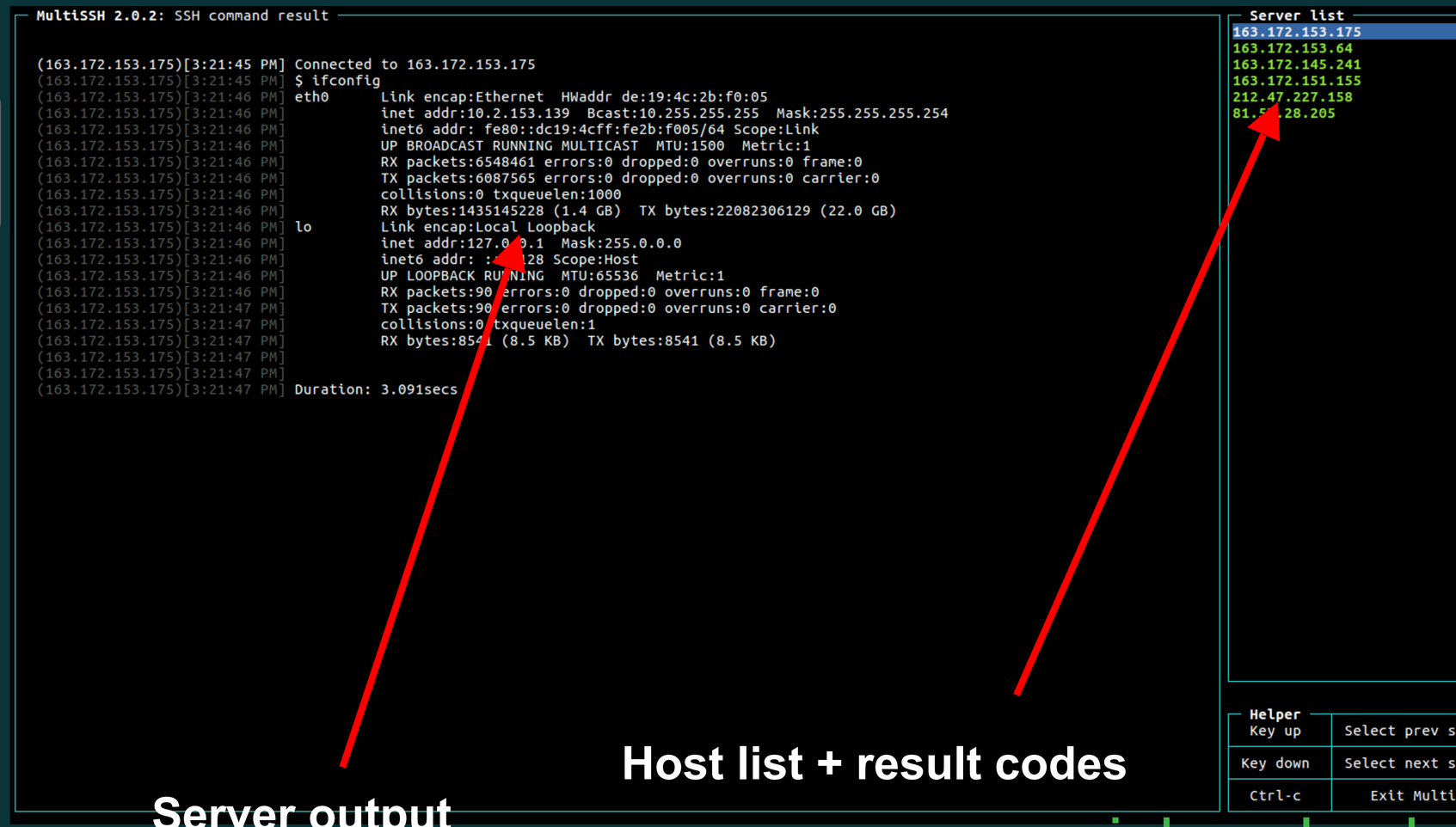
```
ssh_public_key = '''
ssh-rsa AAAAB3NzaC1y                                    3PD0fb8ZGSTeu6G8vZf/2
2itT8FYEOLv5+gAYj1au
'''
```

SSH PUBLIC KEY

Setup a Grid: Provisioning

$ grid provision

Copy Public Key
Install NodeJS
Install PM2
Install Gridcontrol
Setup configuration



gridcontrol

$ grid dash

Grid Dashboard

Grid Nodes

Node name	Public IP	Private IP	Sync
fluffy-donkey	163.172.145.241	10.2.87.217	true
grandiose-able	163.172.153.242	10.2.50.23	true
magenta-pigs	163.172.143.223	10.3.68.7	true
cruel-degree	163.172.151.155	10.3.127.67	true
plastic-note	212.47.227.158	10.2.142.67	true
fallacious-treatment	23.91.96.115	172.31.9.79	true
obeisant-quiet	23.91.96.115	10.8.16.223	master

Load Balancer Stats

Task Name	Invok.	Succ.	Err.	Remote
echo	0	0	0	0
get-ip	91	87	0	24

Local Gridcontrol logs

```
gridcontrol: Fri, 02 Sep 2016 16:07:55 GMT gc:load-balancer status=routing task=get-ip
gridcontrol: Fri, 02 Sep 2016 16:07:55 GMT gc:main Peer synced! { link: '240b043b74bd9
gridcontrol: Fri, 02 Sep 2016 16:07:55 GMT gc:load-balancer status=routing task=get-ip
gridcontrol: Fri, 02 Sep 2016 16:07:55 GMT gc:load-balancer status=routing task=get-ip
gridcontrol: Fri, 02 Sep 2016 16:07:56 GMT gc:tasks status=action_success task=get-ip
gridcontrol: Fri, 02 Sep 2016 16:07:56 GMT gc:load-balancer status=routing task=get-ip
gridcontrol: Fri, 02 Sep 2016 16:07:56 GMT gc:load-balancer status=routing task=get-ip
gridcontrol: Fri, 02 Sep 2016 16:07:56 GMT gc:load-balancer status=routing task=get-ip
gridcontrol: Fri, 02 Sep 2016 16:07:57 GMT gc:load-balancer status=routing task=get-ip
gridcontrol: Fri, 02 Sep 2016 16:07:57 GMT gc:load-balancer status=routing task=get-ip
gridcontrol: Fri, 02 Sep 2016 16:07:57 GMT gc:load-balancer status=routing task=get-ip
gridcontrol: Fri, 02 Sep 2016 16:07:57 GMT gc:tasks status=action_success task=get-ip
gridcontrol: Fri, 02 Sep 2016 16:07:57 GMT gc:load-balancer status=routing task=get-ip
gridcontrol: Fri, 02 Sep 2016 16:07:58 GMT gc:load-balancer status=routing task=get-ip
gridcontrol: Fri, 02 Sep 2016 16:07:58 GMT gc:load-balancer status=routing task=get-ip
gridcontrol: Fri, 02 Sep 2016 16:07:58 GMT gc:load-balancer status=routing task=get-ip
gridcontrol: Fri, 02 Sep 2016 16:07:59 GMT gc:load-balancer status=routing task=get-ip
gridcontrol: Fri, 02 Sep 2016 16:07:59 GMT gc:load-balancer status=routing task=get-ip
gridcontrol: Fri, 02 Sep 2016 16:07:59 GMT gc:load-balancer status=routing task=get-ip
```

Tasks being processed

Task Name	Started	Public IP	Private
get-ip	00:09:58	23.91.96.115	
get-ip	00:09:59	163.172.145.241	
get-ip	00:09:59	163.172.153.242	

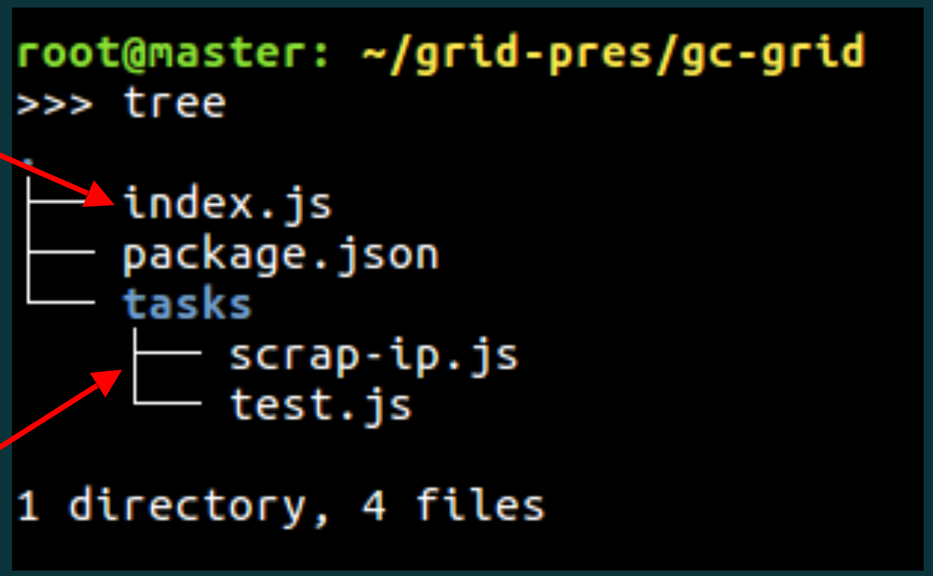
2# Let’s play with the Grid

Base Project Structure

$ grid sample

Main application that invokes services

Those are our services



gridcontrol

Triggering a Service Task

Do a request to get the current node IP

```
var request = require('request');

exports.getIp = function(data, cb) {
  request('https://api.ipify.org/?format=json', function (error, res, body) {
    if (error || res.statusCode != 200)
      return cb(error);
    return cb(null, body);
  });
};
```

Invoke the function in round robin

```
var grid = require('grid-api');

grid.init({
  instances_per_task : 2
});

grid.invoke('req.getIp', function(err, result, node_meta) {
  console.log('Got result from node %s', node_meta.public_ip);
  console.log('Result = ', result);
});
```

gridcontrol


GOLDSTONE 24
70 – Meter Antenna
DEEP SPACE NETWORK
Demo Time!
PM2
gridcontrol

谢谢

PM2

github.com/Unitech/pm2

gridcontrol

github.com/gridcontrol/gridcontrol

Alexandre Strzelewicz