Quilt

Ethan J. Jackson, Aurojit Panda, Kevin Lin,
Johann Schleier-Smith, Nicholas Sun, Luise Valentin,
Yuen Mei Wan, Scott Shenker

SPARK SUMMIT 2016
DATA SCIENCE AND ENGINEERING AT SCALE

quilt.io

Everything has an API

Compute


amazon
web services™
EC2


Google Cloud Platform


openstack™


Microsoft Azure


DigitalOcean


docker
SWARM


MESOS


Nomad


kubernetes
Google

Network


socketplane


weaveworks


vmware
NSX


OVN
(Open Virtual Network)


OPEN
DAYLIGHT


PROJECT
CALICO

DevOps

1. Choose a Compute API
2. Choose a Network API
3. Write a Deployment Script

Deployment Script

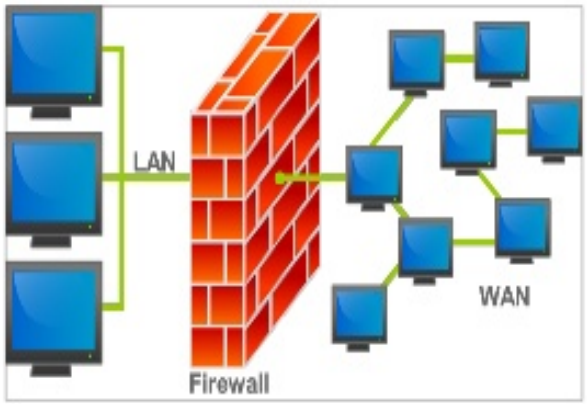
Simple right?

spark-ec2.py

- Official Spark Script
- 1528 Lines of Code
- Incomprehensible

Network Security

- Status Quo
  - Secure the Perimeter
- A Better Way
  - Tight East-West Firewall
  - Increased script complexity



Portability

amazon web services™ | EC2

Microsoft Azure

?

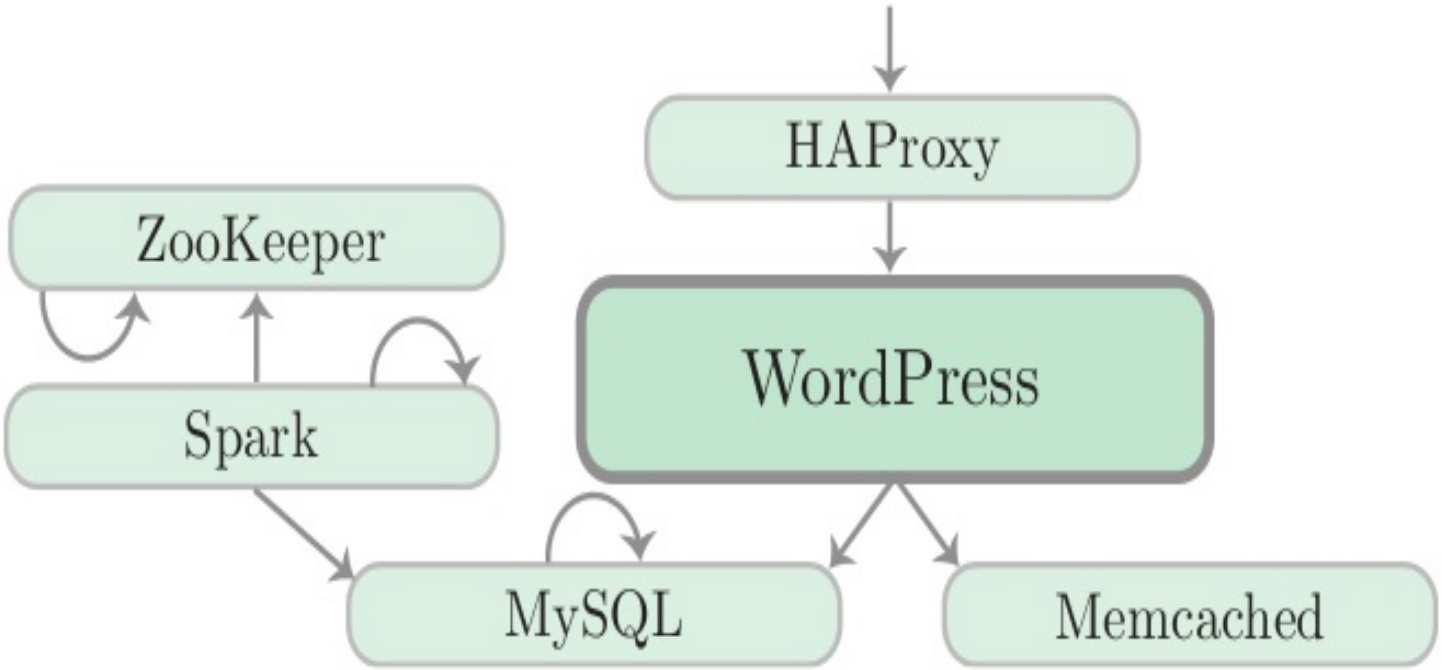
Quilt

Automated Deployment

Quilt DSL: Stitch

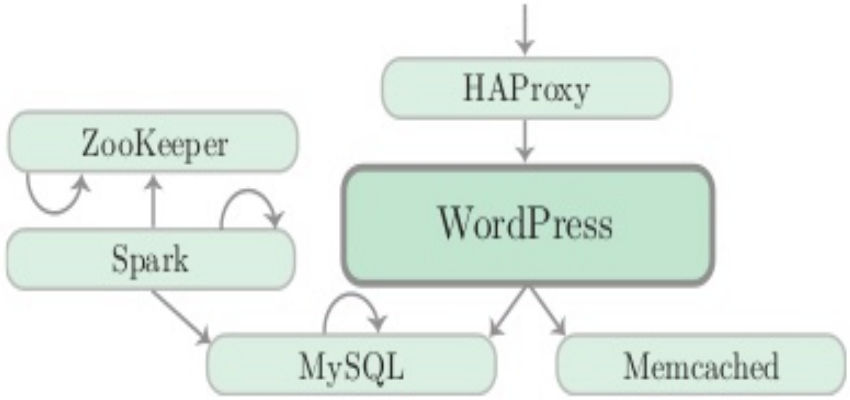
- Declarative Application Specification
- Lisp Dialect
- Declaration Includes:
  - Application Network and Compute
  - Infrastructure

Example: Wordpress



WordPress

```
(import "haproxy")
(import "memcached")
(import "mysql")
(import "spark")
(import "wordpress")
(import "zookeeper")

(let ((db (mysql.New "db" 2))
      (memcd (memcached.New "memcd" 3))
      (wp (wordpress.New "wp" 8 db memcd))
      (hap (haproxy.New "hap" 2 wp))
      (zk (zookeeper.New "zk" 3))
      (spark (spark.New "spark" 2 4 zk)))
  (connect 7077 (hmapValues spark)
                (hmapValues db))
  (connect 80 "public" hap))
```



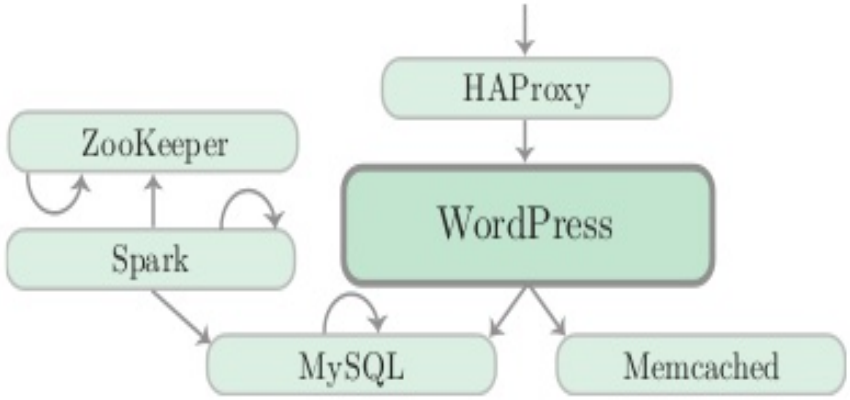
wordpress.New

```
1 (define (New name n db memcd)
2   (let ((dk (makeList n (docker image)))
3         (labelNames (strings.Range name n))
4         (wp (map label labelNames dk)))
5     (configure wp db memcd)
6     (connect 3306 wp (hmapGet db "master"))
7     (connect 3306 wp (hmapGet db "slave"))
8     (connect 11211 wp memcd)
9     wp))
```

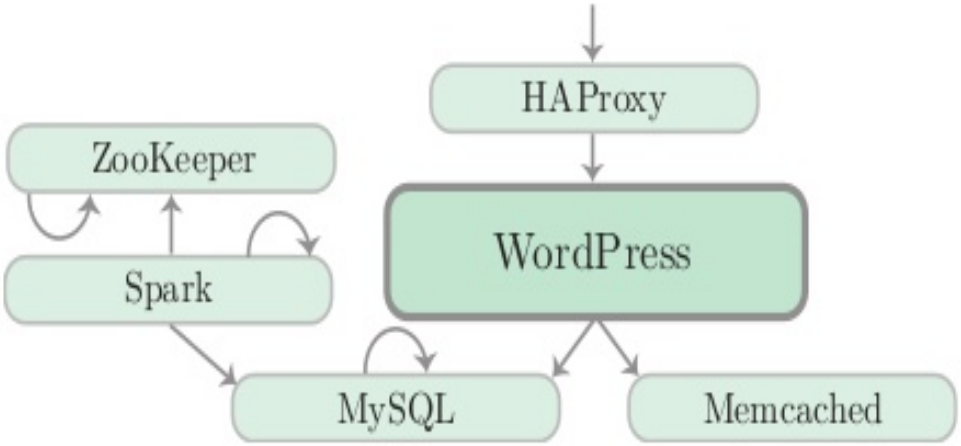
```
(wordpress.New "wp" 8 db memcd)
```

WordPress

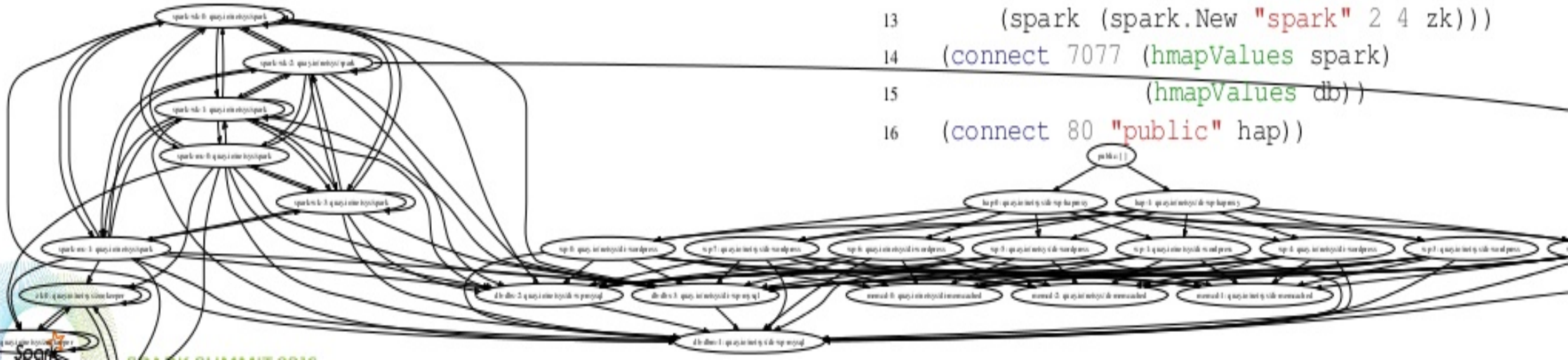
```
(import "haproxy")
(import "memcached")
(import "mysql")
(import "spark")
(import "wordpress")
(import "zookeeper")

(let ((db (mysql.New "db" 2))
      (memcd (memcached.New "memcd" 3))
      (wp (wordpress.New "wp" 8 db memcd))
      (hap (haproxy.New "hap" 2 wp))
      (zk (zookeeper.New "zk" 3))
      (spark (spark.New "spark" 2 4 zk)))
  (connect 7077 (hmapValues spark)
                (hmapValues db))
  (connect 80 "public" hap))
```



WordPress




```
(import "haproxy")
(import "memcached")
(import "mysql")
(import "spark")
(import "wordpress")
(import "zookeeper")

(let ((db (mysql.New "db" 2))
      (memcd (memcached.New "memcd" 3))
      (wp (wordpress.New "wp" 8 db memcd))
      (hap (haproxy.New "hap" 2 wp))
      (zk (zookeeper.New "zk" 3))
      (spark (spark.New "spark" 2 4 zk)))
  (connect 7077 (hmapValues spark)
                (hmapValues db))
  (connect 80 "public" hap))
```



Infrastructure

```
1 (define cfg
2   (list (provider "Amazon") (region "us-west-1")
3         (ram 32 64) (cpu 4 8) (sshkey "elided")))
4
5 (makeList 3 (machine (role "Master") cfg))
6 (makeList 32 (machine (role "Worker") cfg))
```

Infrastructure

```
1 (define cfg
2   (list (provider "Amazon") (region "us-west-1")
3         (ram 32 64) (cpu 4 8) (sshkey "elided")))
4
5 (makeList 3 (machine (role "Master") cfg))
6 (makeList 32 (machine (role "Worker") cfg))
```

Infrastructure

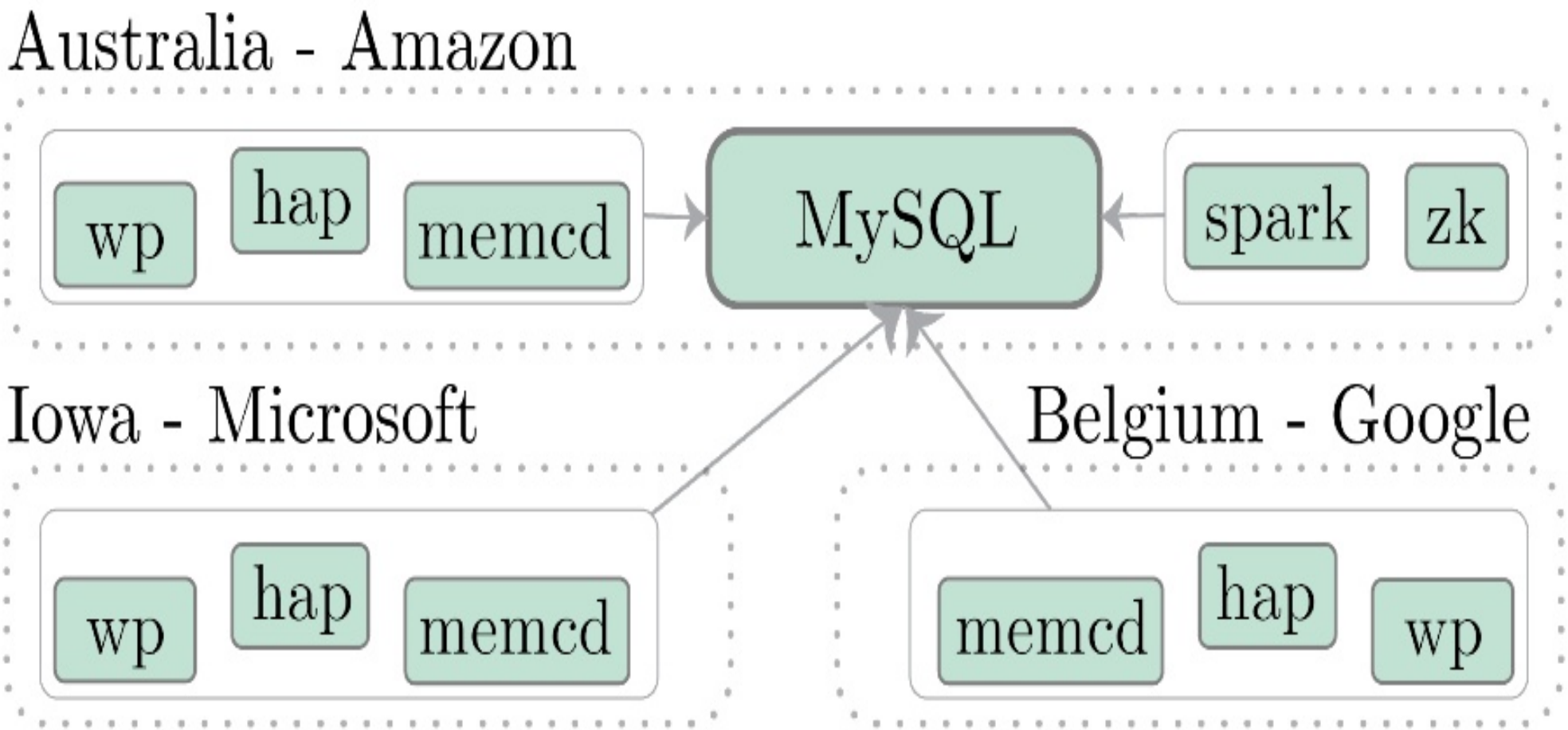
```
1 (define cfg
2   (list (provider "Amazon") (region "us-west-1")
3         (ram 32 64) (cpu 4 8) (sshkey "elided")))
4
5 (makeList 3 (machine (role "Master") cfg))
6 (makeList 32 (machine (role "Worker") cfg))
```

Azure

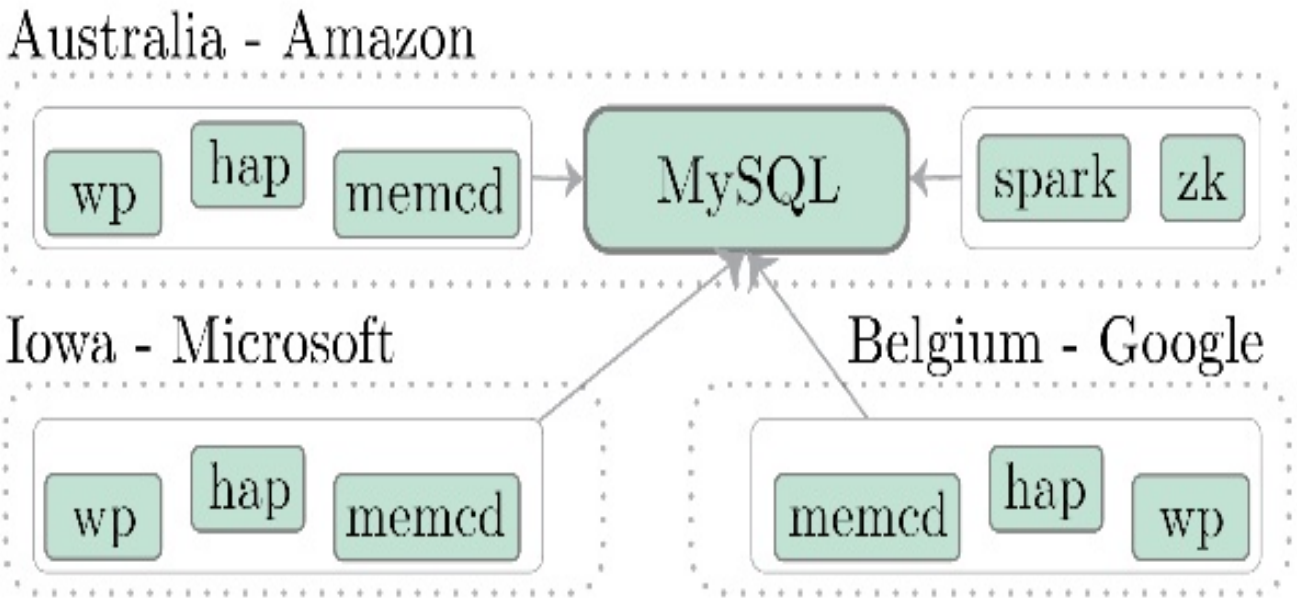
Central US

amazon web services™ EC2 → Microsoft Azure ?

Geographical Distribution



Geographical Distribution




```
(define cfg (list (ram 32 64) (cpu 4 8)
                  (sshkey "<elided>")))

(define db (mysql.New "db" 2))
(define zk (zookeeper.New "zk" 3))
(define spark (spark.New "spark" 2 4 zk))
(connect 7077 (hmapValues spark) (hmapValues db))

(define (makeLoc prvd rgn)
  (list (provider prvd) (region rgn)))

(define (makePod name)
  (let ((memcd (memcached.New (+ name "-mem") 1))
        (wp (wordpress.New (+ name "-wp")
                           2 db memcd))
        (hap (haproxy.New (+ name "-hap") 1 wp)))
    (connect 80 "public" hap)
    (list memcd wp hap)))

(define (deploy pod loc)
  (makeList 16 (machine (role "Worker") cfg loc))
  (place (machineRule "on" loc) pod))

(deploy (makePod "gce")
        (makeLoc "Google" "europe-west1-b"))

(deploy (makePod "azure")
        (makeLoc "Azure" "Central US"))

(let ((loc (makeLoc "Amazon" "ap-southeast-2"))
      (nodes (append (makePod "aws") zk
                     (hmapValues db)
                     (hmapValues spark))))
```

Stitch

Stitch

- Lisp (Scheme)
  - Variables
  - Arithmetic
  - Functions
  - Modules
- Domain Specific Primitives

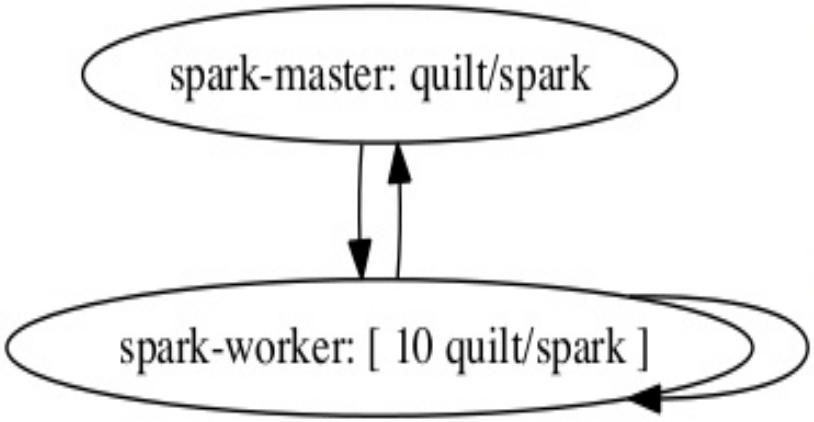
Stitch — Primitives

- Application Primitives
  - “docker”, “label”, “connect”, “place”, “setEnv”
- Infrastructure Primitives
  - “machine”
  - “role”, “provider”, “region”, “ram”, “cpu”, “size”

Stitch — Primitives

```
(label "spark-master" (docker "quilt/spark" "start-master.sh"))

(label "spark-worker"
       (makeList 10 (docker "quilt/spark" "start-worker.sh"
                            "spark://spark-master.di:7077")))

// Spark workers listen on random ports. Must open up everything.
(connect (list 1000 65535)
         (list "spark-master" "spark-worker")
         "spark-worker")

(connect 7077 "spark-worker" "spark-master")
```

Stitch — Primitives



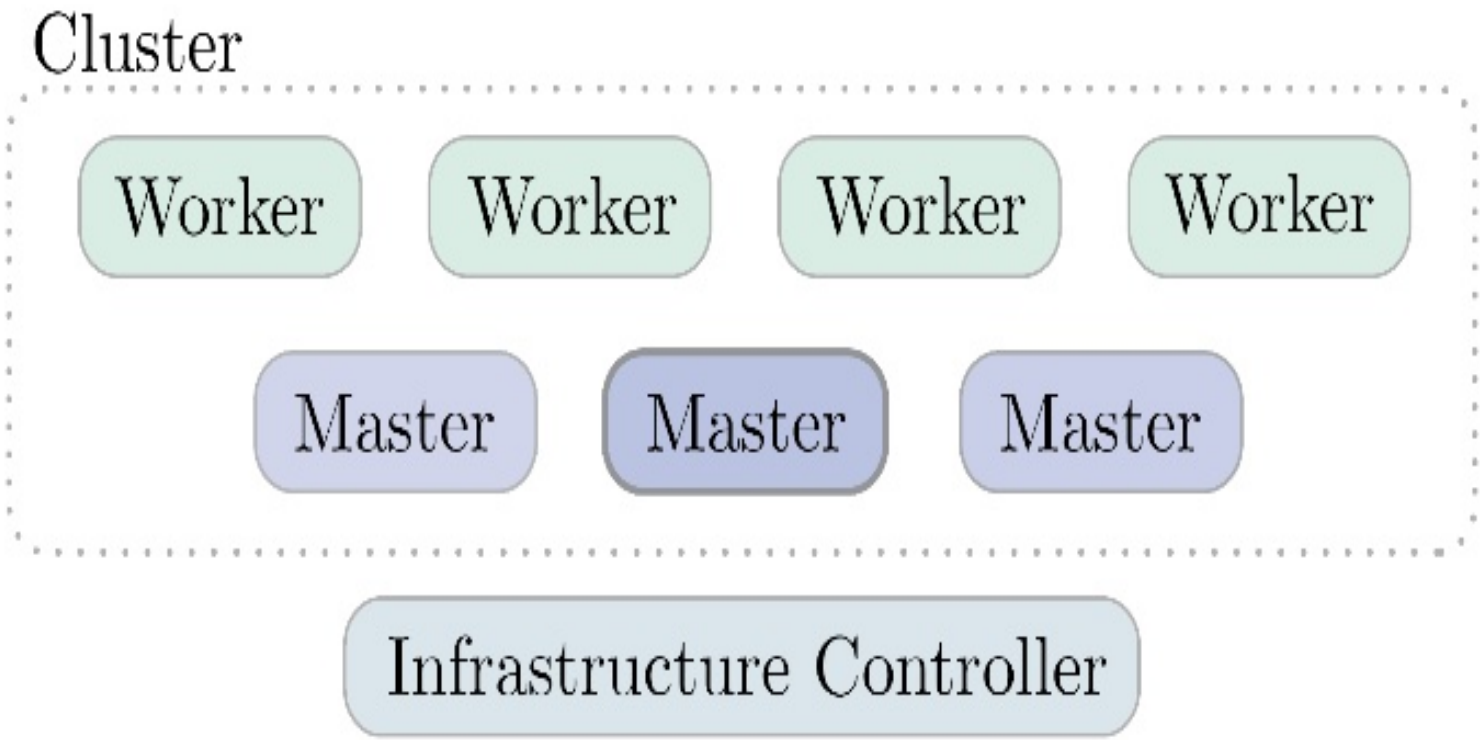

```
(label "spark-master" (docker "quilt/spark" "start-master.sh"))

(label "spark-worker"
       (makeList 10 (docker "quilt/spark" "start-worker.sh"
                            "spark://spark-master.di:7077")))

// Spark workers listen on random ports. Must open up everything.
(connect (list 1000 65535)
         (list "spark-master" "spark-worker")
         "spark-worker")

(connect 7077 "spark-worker" "spark-master")
```

Quilt Architecture

Goals

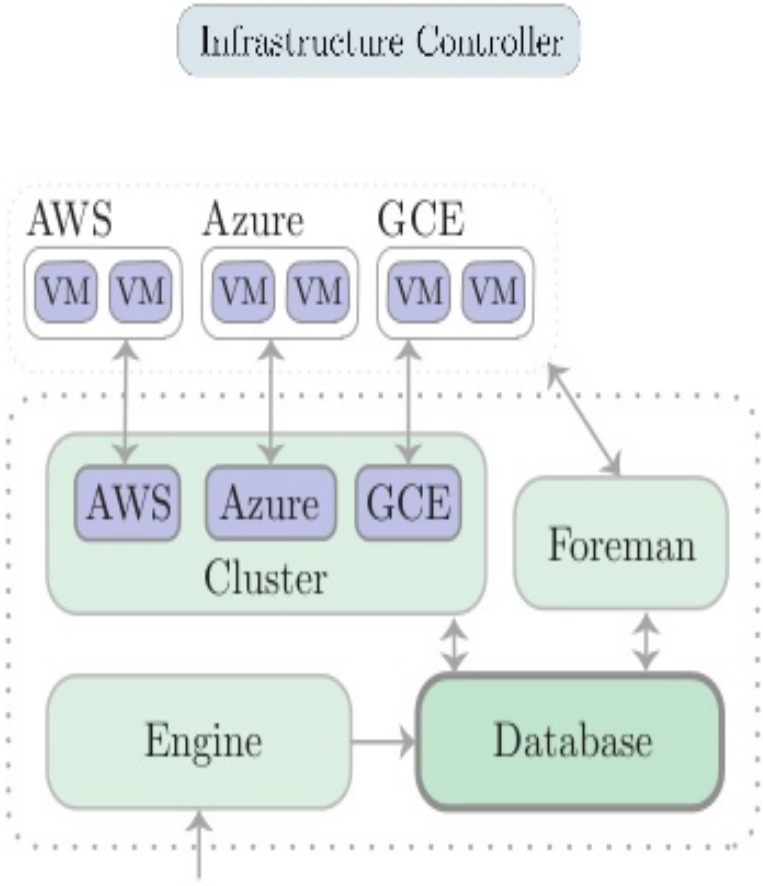
- Simple
- Robust
- Portable

Quilt Architecture



Infrastructure Controller

- Import Infrastructure Spec
- Update Cluster
- Cloud Provider Plugins
  - Amazon EC2
  - Google Compute Engine
  - Microsoft Azure



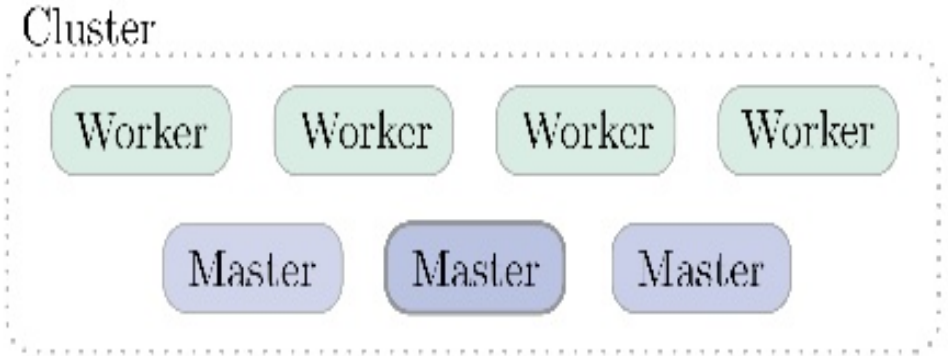
Cloud Provider

- Boot, Stop, List
- Network Reachability
- **Application Agnostic**

```
type Provider interface {
    Connect(namespace string) error

    List() ([]Machine, error)

    Boot([]Machine) error

    Stop([]Machine) error

    SetACLs(acls []string) error

    Disconnect()

    ChooseSize(ram dsl.Range, cpu dsl.Range,
               maxPrice float64) string
}
```

Quilt Cluster

- Virtual Machines Running …
- Application Containers
- Open Virtual Network
  - SDN Overlay
- **Infrastructure Agnostic**



Unsolved Problems

- Application Configuration
- Container Security
- State
- External Services

Related Work

Related Work

- Container Orchestrators
  - Kubernetes, Docker Swarm, Mesos, Nomad
  - No explicit application specification
  - No tight network firewall
- Quilt is a policy layer above these systems

Related Work

- Docker Compose / Kubernetes Helm
  - Declare Groups of Containers to Boot
- Static Data Serialization Format
  - Poor modularity
- Missing network graph

Future Work

Stitch: New Domains

- Security policy
  - Key Management
  - User Management
- Data
- Application Configuration

Stitch Analysis

- Verification
  - Stitch specifies app *entirely*
  - Simpler to verify than deployed systems
- Reachability
- Availability

Summary

- Portable Application Deployment
- Strict Network Security
- Modular, Shareable, Reusable Specifications
- In Future — Formal Analysis

Thank you

quilt.io

ejj@eecs.berkeley.edu