Twine: A Unified Cluster Management System for Shared Infrastructure

Data center geographic regions

- Papillion, NE
- Eagle Mountain, UT
- Prineville, OR
- Los Lunas, NM
- Fort Worth TX
- Altoona, IA
- DeKalb, IL
- New Albany, OH
- Gallatin, TN
- Henrico, VA
- Forest City, NC
- Newton County, GA
- Huntsville, AL
- Luleå, Sweden
- Clonee, Ireland
- Odense, Denmark
- Singapore

New construction
Serving traffic
Cluster management systems help manage all of our services and machines.
What design decisions did Twine make differently?

<table>
<thead>
<tr>
<th>Decision 1</th>
<th>Decision 2</th>
<th>Decision 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic machine partitioning</td>
<td>Customization in shared infrastructure</td>
<td>Small machines</td>
</tr>
<tr>
<td>over static clusters</td>
<td>over private pools</td>
<td>over big machines</td>
</tr>
</tbody>
</table>

What design decisions did Twine make differently?
What design decisions did Twine make differently?

<table>
<thead>
<tr>
<th>Decision 1</th>
<th>Decision 2</th>
<th>Decision 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic machine partitioning over static clusters</td>
<td>Customization in shared infrastructure over private pools</td>
<td>Small machines over big machines</td>
</tr>
</tbody>
</table>
What if we used Kubernetes?

Building large clusters

Support

At v1.19, Kubernetes supports clusters with up to 5000 nodes. More specifically, we support configurations that meet all of the following criteria:
What if we used Kubernetes?
What if we used Kubernetes?

Stranded capacity: M6, M7, M8 are available, but jobs in Cluster 1 cannot use them.

Cluster 1

Cluster 2

M1, M2, M3, M4

M5, M6, M7, M8
What if we used Kubernetes?

**Stranded capacity:** M6, M7, M8 are available, but jobs in Cluster 1 cannot use them.

**Intuition:** Make machine assignment dynamic.
How does Twine avoid stranded capacity?
How does Twine avoid stranded capacity?
How does Twine perform fleet-wide optimization?
How does Twine perform fleet-wide optimization?

Use M5 in newly-constructed Data Center 3 to improve spread for fault tolerance.
How does Twine perform fleet-wide optimization for an entire geographic region?
How does Twine perform fleet-wide optimization for an entire geographic region?
How well does the Twine scheduler scale?
How do we mitigate risks with 1M machines per deployment?

Control plane failure
- replicated
- tasks keep running
- sharded

Network partitions within a region
- fail-safes
- network redundancy
- gradual and frequent releases
- run dependencies
- run itself
- recurring large-scale failure test
What design decisions did Twine make differently?

Decision 1: Dynamic machine partitioning over static clusters

Decision 2: Customization in shared infrastructure over private pools

Decision 3: Small machines over big machines
Private pools or shared infrastructure?
Private pools or shared infrastructure?

ML Private Pool

M1
M2
M3
M4

Web Private Pool

M5
M6
M7
M8

M6 and M7 are underutilized and cannot be shared across private pools
Private pools or shared infrastructure?

ML Private Pool

Web Private Pool

M6 and M7 are underutilized and cannot be shared across private pools

Shared Infrastructure

M6 and M7 can be reused to run other workloads
What are the challenges with supporting ubiquitous shared infrastructure?

**Challenge 1**

Performance wins through host customization
What is host customization?

- NIC settings
- kernel version
- CPU Turbo
- file system settings (e.g., btrfs, XFS)
- sysctls (e.g., huge pages, kernel scheduler settings)
What is host customization?

**Intuition:** Dynamically reconfigure machines as needed by workloads.
Challenge: Performance wins through host customization
Challenge: Performance wins through host customization

Automatically reconfigure M3 host settings

Kernel 5.6, Huge Pages, Turbo

Entitlement 1

M1

Entitlement 2

M3

Kernel 5.2, NIC sysctls

M3

M5

Data Center 1

Data Center 2

Data Center 3
What is the overhead for host profile switches?

On average, a machine changes host profiles once every ~2 days

<table>
<thead>
<tr>
<th>Time (seconds) (log scale)</th>
<th>CPU Turbo</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>Huge Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.0</td>
<td>3.6</td>
<td>4.1</td>
<td>5.1</td>
<td>5.6</td>
<td>7.3</td>
<td>8.5</td>
<td>12.2</td>
<td>27.9</td>
<td>244.0</td>
</tr>
</tbody>
</table>
What drives host profile changes?

Change machines to run ALM-tracking services
What drives host profile changes?

Change machines to run ALM-tracking services

Change machines to run non-ALM-tracking services
What drives host profile changes?

Shared infrastructure makes machines fungible and improves efficiency!
What are the challenges with supporting ubiquitous shared infrastructure?

**Challenge 1**
Performance wins through host customization

**Challenge 2**
Tasks are not homogenous

**Approach**
Host profiles
Challenge: Tasks are not homogenous

Need at least 1 replica of each shard up at all times
Challenge: Tasks are not homogenous

Task 4 becomes unavailable.
Challenge: Tasks are not homogenous

Task 1 restarted for software release. Shard C unavailable!
Intuition: Collaborate with applications to handle lifecycle events.

Task 1 restarted for software release.
Shard C unavailable!
How does Twine collaborate with applications?

TaskController State

Scheduler

TaskController
How does Twine collaborate with applications?

2 and 3 have no overlapping shards with 4, but overlap in B. Can update only one of these.
How does Twine collaborate with applications?

Scheduler

Time

S0

Update Job

request=[1,2,3,4]  completed=[ ]

S1

ack=[2]

S2

request=[1,3,4]  completed=[2]

S3

ack=[3]

Can only update 3 to maintain C’s availability

TaskController

TaskController State

A  B  A  C
1  2  3  4
4 unavailable
How does Twine collaborate with applications?
How does Twine collaborate with applications?

```
Time

Scheduler

S0  
Update Job
request=[1,2,3,4] completed=[ ]

S1
request=[1,3,4] completed=[2]
ack=[2]

S2
request=[1,4] completed=[3]
ack=[3]

S3

TaskController

TaskController State

A  B  A  C
1  2  3  4
```

Can update 1 or 4, but they overlap in C. Can update only one of these.
What are the challenges with supporting ubiquitous shared infrastructure?

**Challenge 1**
Performance wins through host customization

**Challenge 2**
Tasks are not homogenous

**Approach**
Host profiles

**Approach**
TaskControl API
What is our shared infrastructure adoption?

twshared: shared infrastructure for compute

- Web fully on twshared
- New compute capacity in twshared only
- Host profiles created
- TaskControl created

Graph showing the adoption rate of shared infrastructure from 2013 to 2020.
How easy is it to migrate onto shared infrastructure?

Case study with PGx: O(100s) services, O(100Ks) machines

800 Entitlements, dense stacking

3 TaskControllers

3 Host profile settings
How easy is it to migrate onto shared infrastructure?

Case study with PGx: O(100s) services, O(100Ks) machines

0% to 70% capacity in twoshared in 9 months.
No new customizations needed.
Power is our most constrained resource

- Papillion, NE
- Eagle Mountain, UT
- Prineville, OR
- Los Lunas, NM
- Fort Worth TX
- Altoona, IA
- DeKalb, IL
- New Albany, OH
- Gallatin, TN
- Henrico, VA
- Forest City, NC
- Newton County, GA
- Huntsville, AL
- Luleå, Sweden
- Clonee, Ireland
- Odense, Denmark
- Singapore

New construction
Serving traffic
What design decisions did Twine make differently?

**Decision 1**
Dynamic machine partitioning over static clusters

**Decision 2**
Customization in shared infrastructure over private pools

**Decision 3**
Small machines over big machines
Big machines or small machines?

**Big machines**
- 2 CPUs, 20 cores each
- 256GB RAM
- Dedicated NIC
- 30 machines per rack

**Small machines**
- 1 CPU, 18 cores
- 64GB RAM
- 4 machines share 1 multi-host NIC
- 92 machines in a rack

**A small-machine rack vs. a big-machine rack**
- 38% more cores
- 34% less memory
Why use small machines?

Existing large services are optimized to fully utilize whole machines.

Resource control historically not mature.

Not a public cloud: no external users.
How much do we save by using small machines?

Relative total cost of ownership

Different Services

Relative TCO

33% 48% 67% 73% 73% 76% 77% 80% 83% 88% 93% 93% 95% 97% 100% 100% 100% 100% 100% 100% 110%
How much do we save by using small machines?

Relative total cost of ownership

18% power savings fleet-wide. 17% savings in total cost of ownership.
What lessons did we learn using small machines?

- Stacking tasks is less effective
- More work required when working with hardware vendors
- Large effort to rearchitect services to fit into less memory
Conclusion
Evolving Twine over the past 10 years

Dynamic machine partitioning
Avoids stranded capacity in isolated clusters and enables fleet-wide optimizations

Customization in shared infrastructure
Support ubiquitous shared infrastructure to improve efficiency without sacrificing workload performance or capability

Small machines
Achieve higher power efficiency globally
Thank you!

Chunqiang Tang, Kenny Yu*, Kaushik Veeraraghavan, Jonathan Kaldor, Scott Michelson, Thawan Kooburat, Aravind Anbudurai, Matthew Clark, Kabir Gogia, Long Cheng, Ben Christensen, Alex Gartrell, Maxim Khutornenko, Sachin Kulkarni, Marcin Pawlowski, Tuomas Pelkonen, Andre Rodrigues, Rounak Tibrewal, Vaishnavi Venkatesan, Peter Zhang, and everyone who contributed to Twine’s development over the past decade.

Contact: kennyyu@fb.com
FACEBOOK Infrastructure