Google™ 10 IO

Testing Techniques For Google App Engine

Max Ross
May 20, 2010

http://bit.ly/GAETesting

View live notes and ask questions about this session on Google Wave

Who Am I?

- Max!
- Software Engineer at Google
- Member of the App Engine team
  - Datastore
  - JPA/JDO Implementation
  - Java Runtime
- http://gae-java-persistence.blogspot.com/
- I feel naked and sad without good test coverage

Agenda

- Developer Testing
- Testing App Engine Apps Locally
- Testing App Engine Apps In The Cloud
- Wrap Up
- Questions

Developer Testing

Terminology

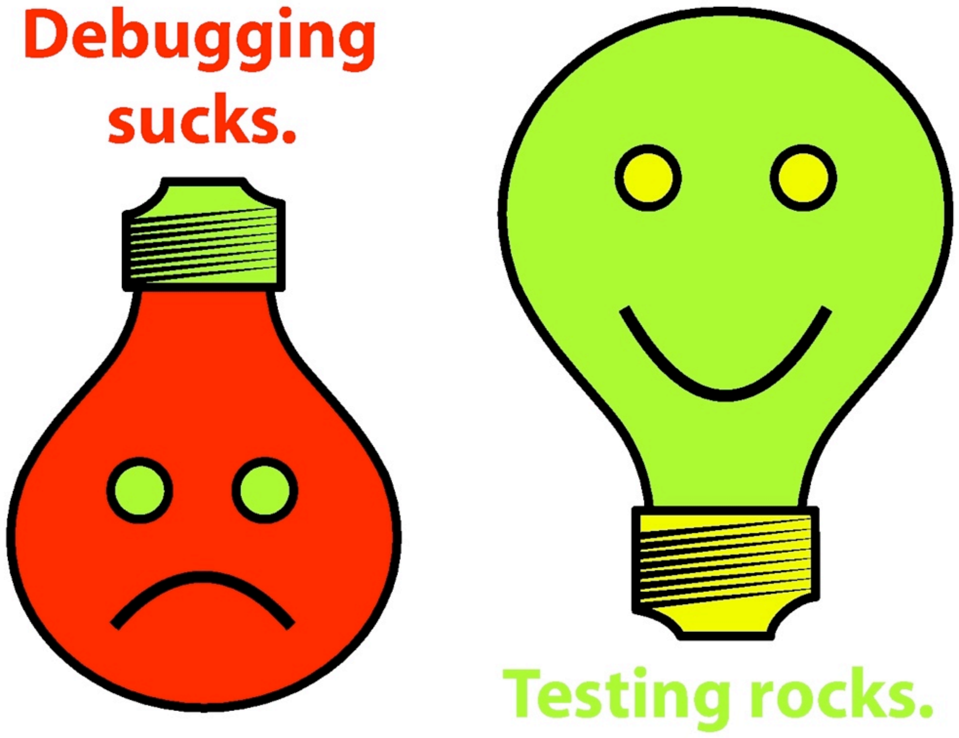
- Unit test
  - tests a single method/class
  - behavior of dependencies controlled by the test
- Integration test
  - tests interaction between multiple methods, classes, or components
  - behavior of dependencies *might* be controlled by the test
- End to end test
  - tests behavior by driving requests through external access points
  - for App Engine that means HTTP

Soapbox: Why Developer Testing Is Important

- Demonstrates correctness of new code
  - important
- Demonstrates continued correctness of old code
  - more important
- Provides a safety net when making large or risky changes
  - code always evolves organically
  - refactoring can help restore order
- Treat your tests as first class citizens in your code base
  - please
- Deploying to App Engine doesn't change any of this!

Testing @ Google

- part of our engineering culture
  - guiceberry
  - googletest
  - googlemock
  - thread-weaver
- googletesting.blogspot.com



App Engine Execution Environments

- Cloud
  - serving HTTP traffic
    - does not have to be prod
- Local
  - serving HTTP traffic
  - equivalent functionality
  - nonequivalent scale
- Tests!
  - serving HTTP traffic and test harness requests
  - local and cloud
  - do we need App Engine specific testing strategies?

Testing App Engine Apps Locally

Local Testing - What’s The Same?

- Frameworks
  - JUnit, pyUnit, Selenium
- Best practices
  - design
  - avoiding flakiness
  - isolation
- End to end testing
- This is great news!

Local Testing - What’s Different?

- RPC layer assumes dev appserver sets up the environment

```
// from com.google.appengine.api.datastore.DatastoreApiHelper.java

static String getCurrentAppId() {
  ApiProxy.Environment environment = ApiProxy.getCurrentEnvironment();
  if (environment == null) {
    throw new NullPointerException(
        "No API environment is registered for this thread.");
  }
  return environment.getAppId();
}
```

- What if there is no dev appserver?
  - set up the environment by hand or...
  - use the App Engine Testing APIs!
    - currently Java only

Local Testing - Demo!

- Guestbook Persistence
- Integration test with the datastore and task queue
  - high(er)-availability writes

Testing App Engine Apps In The Cloud

I Want To Run My Tests In The Cloud!

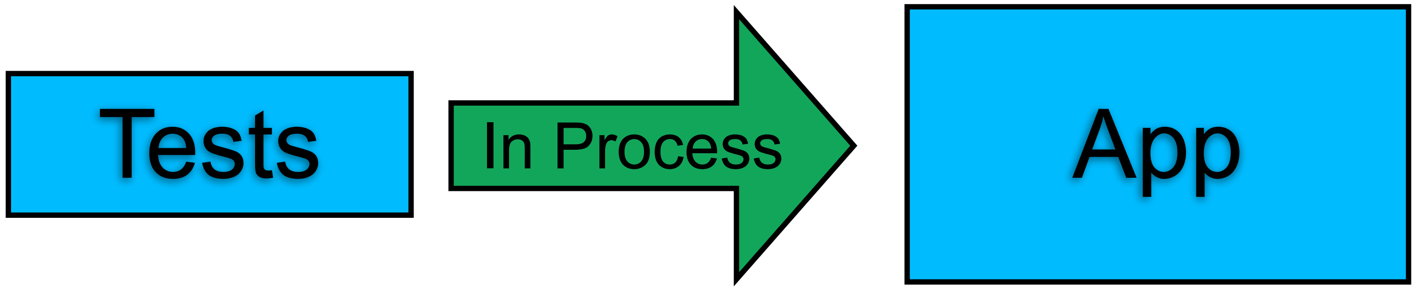
- Why?
  - Fidelity
    - true behavior and performance
  - Efficiency
    - run large test suites faster
    - let someone else manage your test grid

Cloud Testing - What's The Same?

- Same similarities as local testing
  - frameworks
  - best practices
  - end to end testing
- This is also great news!

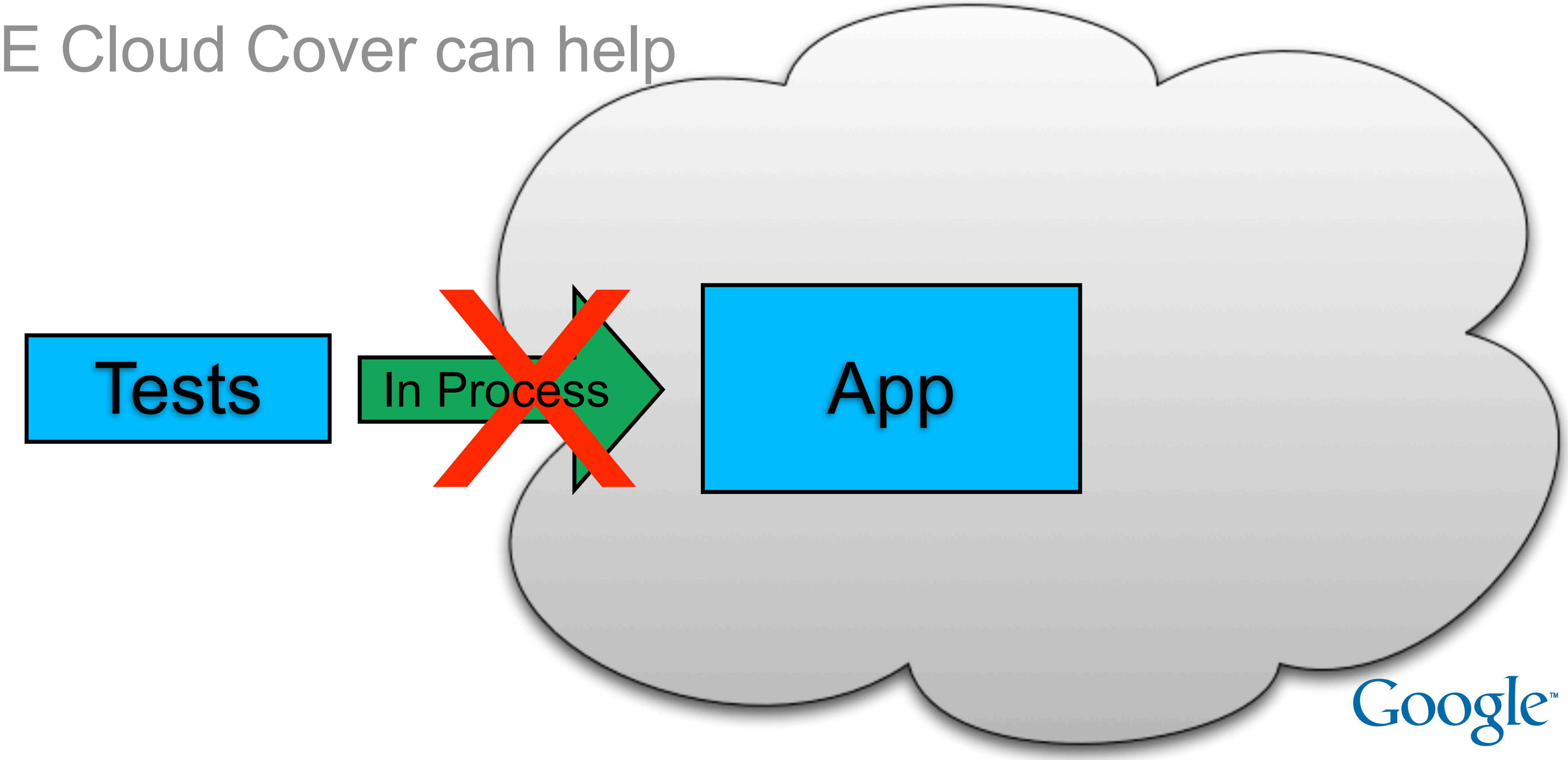
Cloud Testing - What's Different?

- Each test must complete in 30 seconds
- App code and test code must obey sandbox restrictions
  - threads, direct network access, etc
- Must invoke tests via HTTP
  - no need to worry about setting up the environment
- GAE Cloud Cover can help



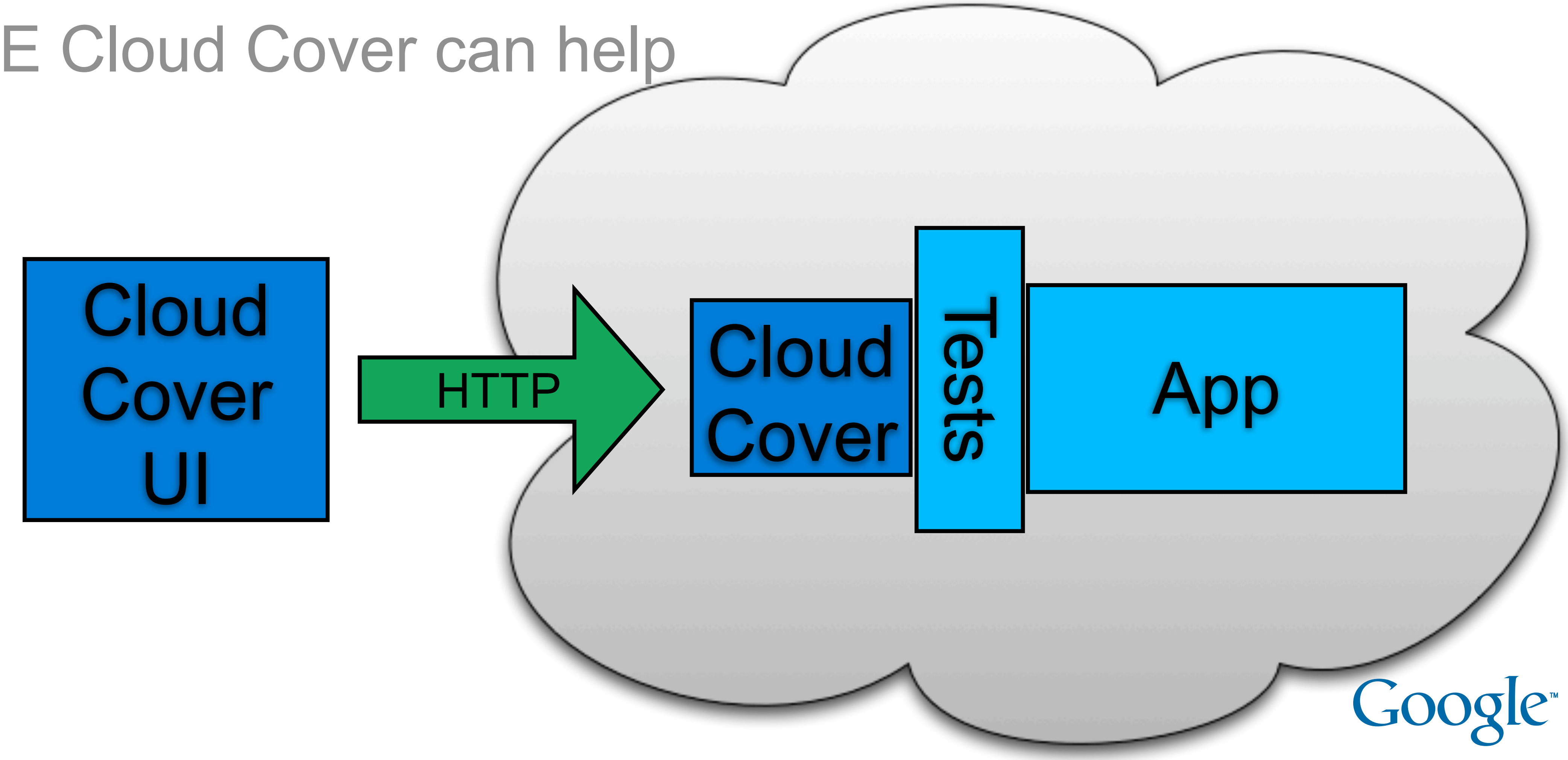
Cloud Testing - What's Different?

- Each test must complete in 30 seconds
- App code and test code must obey sandbox restrictions
  - threads, direct network access, etc
- Must invoke tests via HTTP
  - no need to worry about setting up the environment
- GAE Cloud Cover can help



Cloud Testing - What’s Different?

- Each test must complete in 30 seconds
- App code and test code must obey sandbox restrictions
  - threads, direct network access, etc
- Must invoke tests via HTTP
  - no need to worry about setting up the environment
- GAE Cloud Cover can help

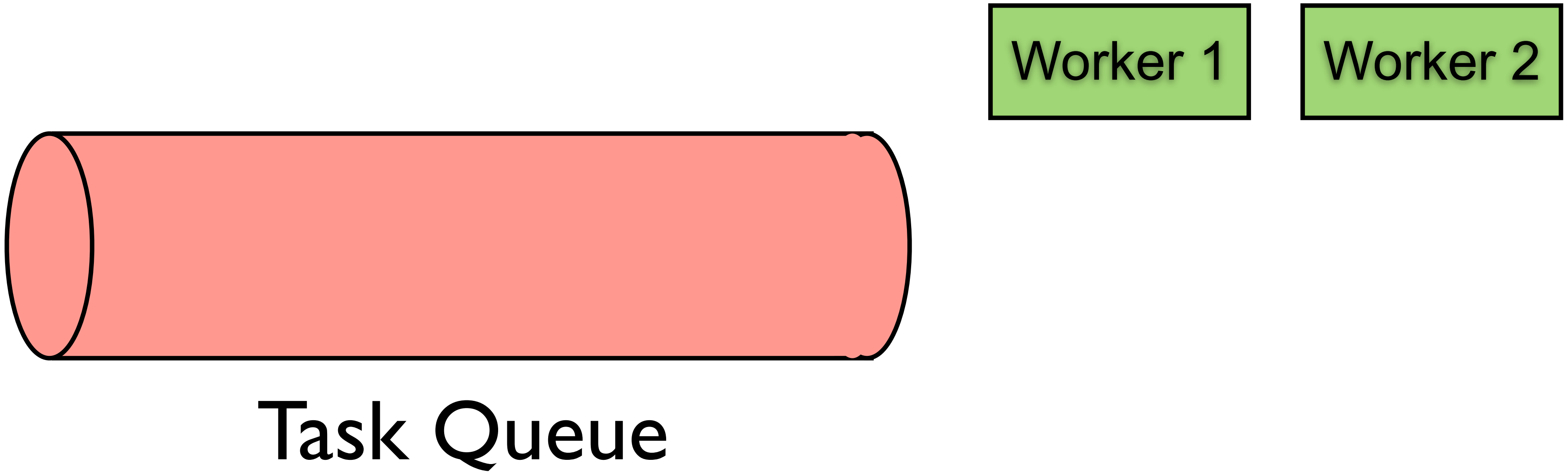


Cloud Testing - Simple Demo!

- GAE Cloud Cover is designed to run *existing* test suites
- Guestbook Persistence tests running in the cloud:
  - install JAR
  - configure Servlet
  - replace LocalServiceTestHelper
  - extend JUnit3Config
  - deploy
  - run
  - curl up under a cozy warm blanket of cloud test coverage

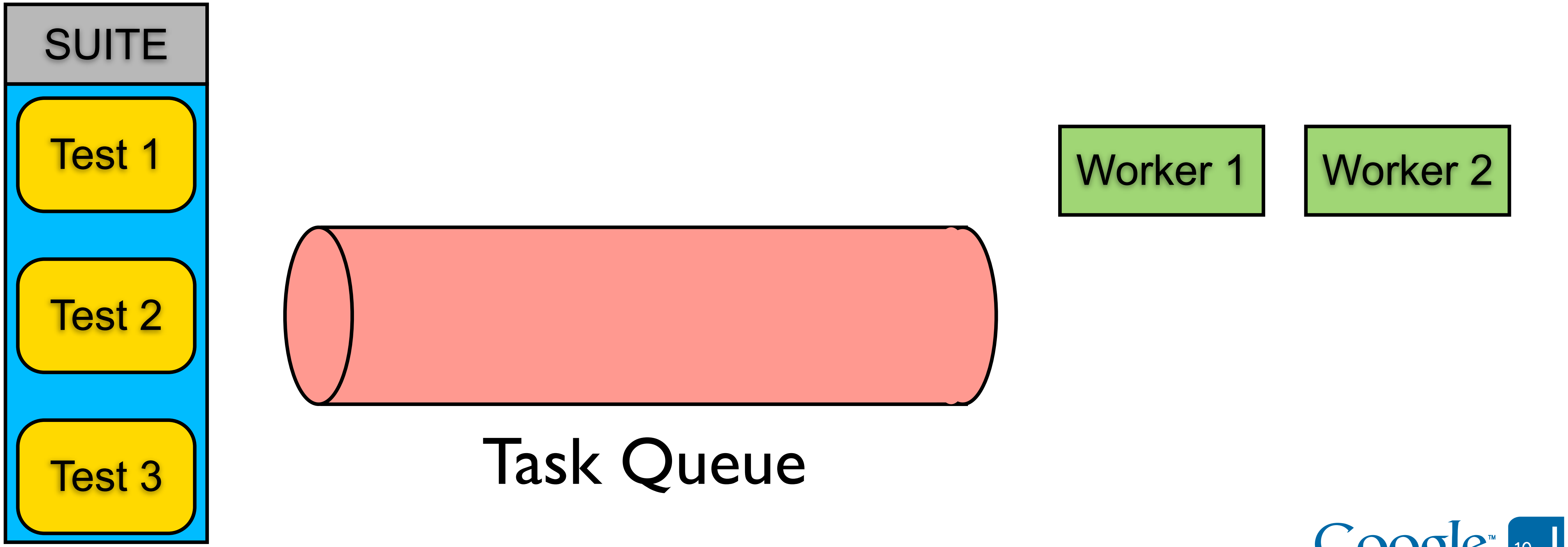
How It Works

- A Suite is a collection of Tests
- GAE Cloud Cover creates one task queue task per Test
- Tests execute in parallel
  - number of workers determined by queue config
  - GAE is good at fan-out!



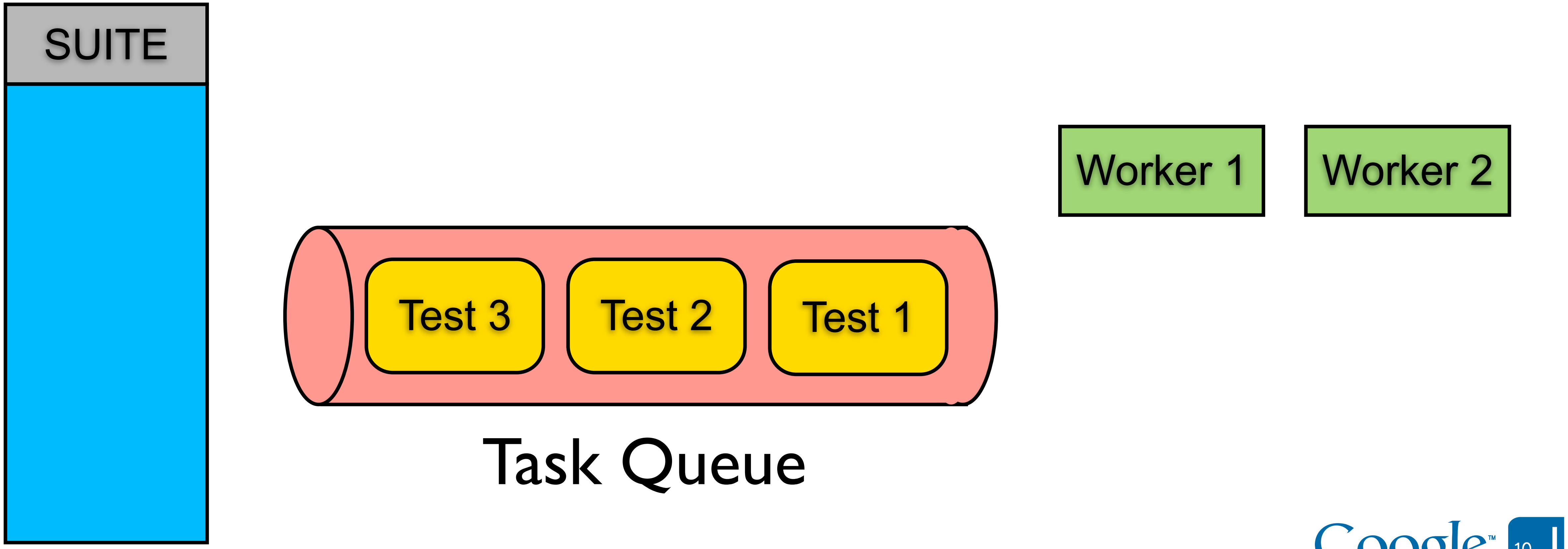
How It Works

- A Suite is a collection of Tests
- GAE Cloud Cover creates one task queue task per Test
- Tests execute in parallel
  - number of workers determined by queue config
  - GAE is good at fan-out!



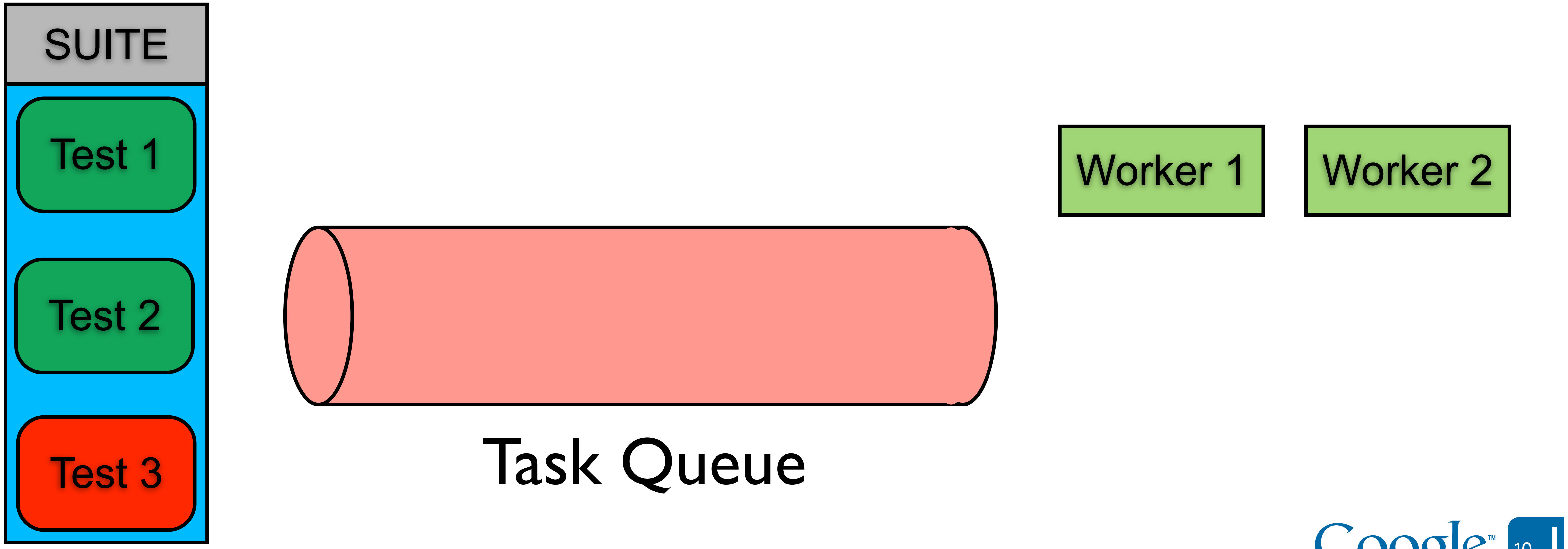
How It Works

- A Suite is a collection of Tests
- GAE Cloud Cover creates one task queue task per Test
- Tests execute in parallel
  - number of workers determined by queue config
  - GAE is good at fan-out!



How It Works

- A Suite is a collection of Tests
- GAE Cloud Cover creates one task queue task per Test
- Tests execute in parallel
  - number of workers determined by queue config
  - GAE is good at fan-out!



Isolation Problems

- Tests must be thread-safe

```
Thread 1: testReadThruCache Thread 2: testQuery
 1) populate cache            1) perform query
 2) install ErrorDelegate     2) verify results
 3) fetch val from cache
```

- Test data must be “thread-safe”

```
Thread 1: testCreateUser    Thread 2: testDeleteUser
 1) create User “m”           1) create User “m”
 2) assert “m” exists         2) delete “m”
                              3) assert “m” does not exist
```

- Non thread-safety yields flakiness
- Flaky tests are worse than consistently failing tests

Isolation Problems

- Tests must be thread-safe

```
Thread 1: testReadThruCache Thread 2: testQuery
 1) populate cache           1) perform query
 2) install ErrorDelegate    2) verify results
 3) fetch val from cache
```

- Test data must be “thread-safe”

```
Thread 1: testCreateUser    Thread 2: testDeleteUser
 1) create User “m”          1) create User “m”
 2) assert “m” exists        2) delete “m”
                             3) assert “m” does not exist
```

- Non thread-safety yields flakiness
- Flaky tests are worse than consistently failing tests

Isolation Problems

- Tests must be thread-safe

```
Thread 1: testReadThruCache Thread 2: testQuery
 1) populate cache           1) perform query
 2) install ErrorDelegate    2) verify results
 3) fetch val from cache
```

- Test data must be “thread-safe”

```
Thread 1: testCreateUser    Thread 2: testDeleteUser
 1) create User “m”          1) create User “m”
 2) assert “m” exists        2) delete “m”
                             3) assert “m” does not exist
```

- Non thread-safety yields flakiness
- Flaky tests are worse than consistently failing tests

Isolation Solutions

- Use ThreadsafeDelegate instead of ApiProxy.Delegate
- Use Namespaces to enforce test data isolation
  - coming soon to an App Engine SDK near you
  - implemented as a way to “stripe” persistent data with arbitrary identifiers
    - this is exactly what we need!

Kind	Namespace	Key
Greeting	testCreateUser	“max”
Greeting	testDeleteUser	“max”

Cloud Testing - Fancy Demo!

- App Engine JDO/JPA implementation test suite
  - userland library
  - 1300+ unit and integration tests
  - makes extensive use of the datastore
  - uses custom ApiProxy.Delegate implementations

Running Arbitrary Test Suites

- Why limit ourselves to test suites for App Engine apps?
- GAE Cloud Cover is framework agnostic
  - requires a small plugin to hook up a framework
    - com.google.appengine.testing.cloudcover.spi
    - implement 3 simple interfaces
- If you have a large test suite, App Engine can be your grid!

Cloud Testing - Arbitrary Demo!

- Google Collections Framework
  - http://code.google.com/p/google-collections/
- 44,776 tests
- Has nothing to do with App Engine

Wrap Up

Looking Ahead

- Testing APIs
  - Python
  - running tests inside the local sandbox
  - launching dev appserver from tests

- GAE Cloud Cover:
  - IDE-integration
  - support for dynamically generated tests
  - project is completely open source
  - http://code.google.com/p/cloudcover/
    - Help!

Conclusions

- Developer testing is Important!
- Your testing skillz transfer
- App Engine Testing API makes local testing easy
- GAE Cloud Cover makes cloud testing easy
- You don’t have to build an App Engine app to use App Engine

Questions

http://bit.ly/GAETesting

Google™ 10 IO