

# Swarm

## Transparent Scalability

James Douglas

@jearldouglas

Bay Area Scala Enthusiasts

8 August, 2011

Atlassian in San Francisco

<http://www.svscala.org/>

# The Problem



**Cheap, fast development**



**Scalability**

# The Problem

## MapReduce (Hadoop)

```
void map(KEYIN key, VALUEIN value, Mapper.Context context)
```

```
void reduce(KEYIN key, Iterable<VALUEIN> values, Reducer.Context context)
```

# The Problem

## MapReduce (Hadoop)

```
void map(KEYIN key, VALUEIN value, Mapper.Context context)
```

```
void reduce(KEYIN key, Iterable<VALUEIN> values, Reducer.Context context)
```

- The domain must fit the MapReduce paradigm
- To easily scale later, MapReduce must be an early design choice
- The `void` return types imply reliance on side effects

# The Holy Grail

- Initially build our application cheaply and quickly
- Later scale without changing the software



# Follow the Data

- Let Swarm worry about where in the cluster data is located
- Use Swarm's `Ref` class to dereference the data
- Transparently relocate execution to the node with the data

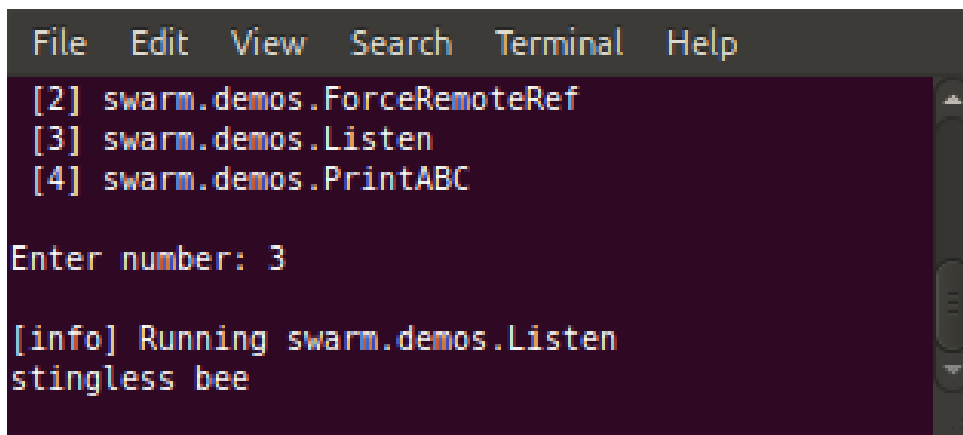


# Swarm

```
val local = new InetSocketAddress(java.net.InetAddress.getLocalHost, 9998)
val remote = new InetSocketAddress(java.net.InetAddress.getLocalHost, remotePort)

val a = Ref(local, "bumble bee")
val b = Ref(local, "honey bee")
val c = Ref(remote, "stingless bee")

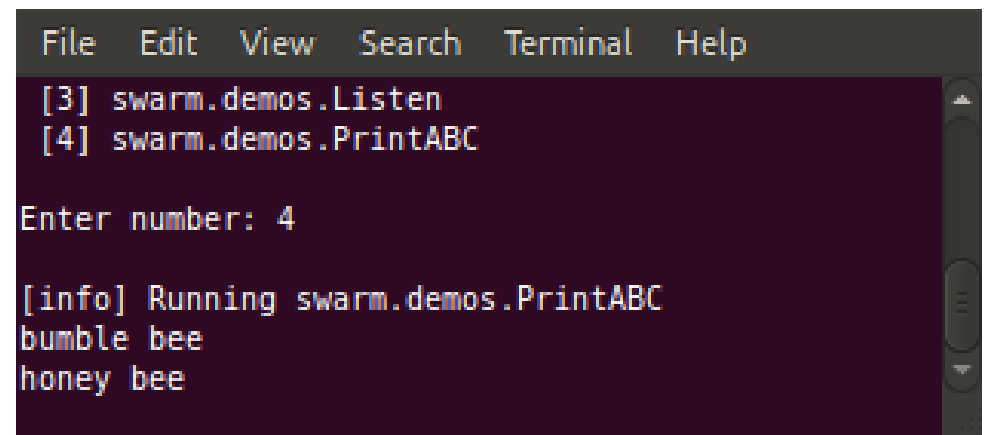
println(a())
println(b())
println(c())
```



```
File Edit View Search Terminal Help
[2] swarm.demos.ForceRemoteRef
[3] swarm.demos.Listen
[4] swarm.demos.PrintABC

Enter number: 3

[info] Running swarm.demos.Listen
stingless bee
```



```
File Edit View Search Terminal Help
[3] swarm.demos.Listen
[4] swarm.demos.PrintABC

Enter number: 4

[info] Running swarm.demos.PrintABC
bumble bee
honey bee
```

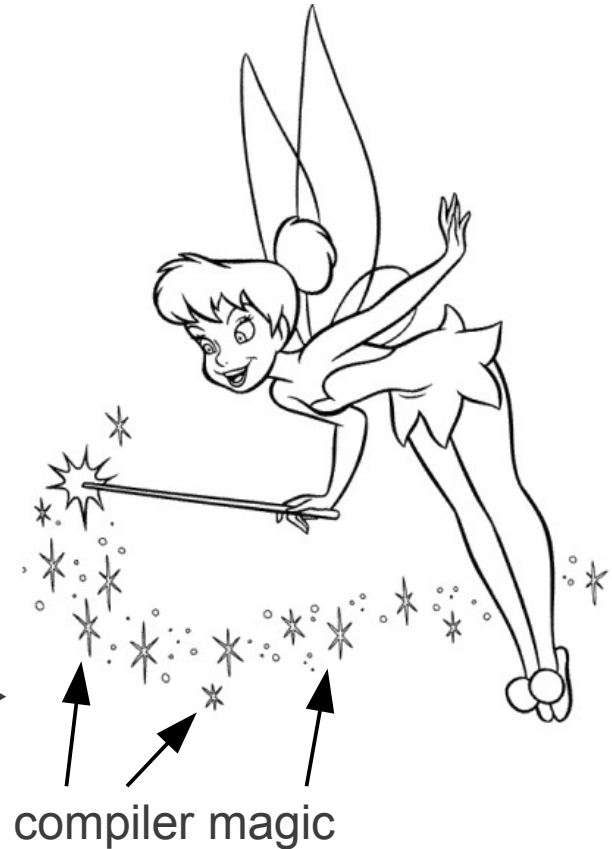
# Delimited Continuations

```
object AsyncConsoleIO extends App {  
  
  type cont = cpsParam[Unit, Unit]  
  
  reset {  
    var i = 5  
    while (i > 0) {  
      shiftIt()  
      i -= 1  
    }  
  }  
  
  def shiftIt() = shift { k: (Unit => Unit) =>  
    val name = prompt()  
    k()  
    greet(name)  
  }  
  
  def prompt() = {println("name> "); readLine }  
  
  def greet(s: String) = { Thread.sleep(1000); println("hello, " + s) }  
}
```



# Delimited Continuations

```
object AsyncConsoleIO extends App {  
  
  type cont = cpsParam[Unit, Unit]  
  
  reset {  
    var i = 5  
    while (i > 0) {  
      shiftIt()  
      i -= 1  
    }  
  }  
  
  def shiftIt() = shift { k: (Unit => Unit) =>  
    val name = prompt()  
    k()  
    greet(name)  
  }  
  
  def prompt() = {println("name> "); readLine }  
  
  def greet(s: String) = { Thread.sleep(1000); println("hello, " + s) }  
}
```



# Delimited Continuations

```
File Edit View Search Terminal Help
Welcome to Scala version 2.9.0.1 (Java HotSpot(TM) Server VM, Java 1.6.0_26).
Type in expressions to have them evaluated.
Type :help for more information.

scala> :load asyncio.scala
Loading asyncio.scala...
import scala.util.continuations._
defined module AsyncConsoleIO

scala> AsyncConsoleIO.main(Array.empty)
name>
name>
name>
name>
name>
hello, jane
hello, sam
hello, bob
hello, mike
hello, john

scala>
```

# Status of Swarm



# Status of Swarm

Lots of cool stuff to do

- Data migration algorithm
- Reference relocation cleanup
- Cluster management
- Security



# Reference

<https://github.com/sanity/Swarm>



<http://www.earldouglas.com/>

# Demo

SwarmTwitter :: node1 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

SwarmTwitter :: node1

http://localhost:8080/

Welcome to SwarmTwitter, a Twitter simulator built using [Swarm](#), a framework allowing the creation of web applications which can scale transparently through a novel portable continuation-based approach.

To use SwarmTwitter, follow the links below to act as any of the sample users, or add your own users by browsing to /<username>.

[view node 1](#)  
[view node 2](#)

## Sample users

[jmcdoe](#)  
[maxpower](#)

## Statuses

[maxpower](#)> Howdy, Swarm! Fri Jun 24 16:18:46 PDT 2011

---

[jmcdoe](#)> Hello, Swarm! Fri Jun 24 16:17:01 PDT 2011

---