Remote Function Applications

A framework towards scala grid parallel collections

Nermin Serifovic

@higherkinded

Co-founder of Boston Area Scala Enthusiasts

Co-organizer of NEScala 2011

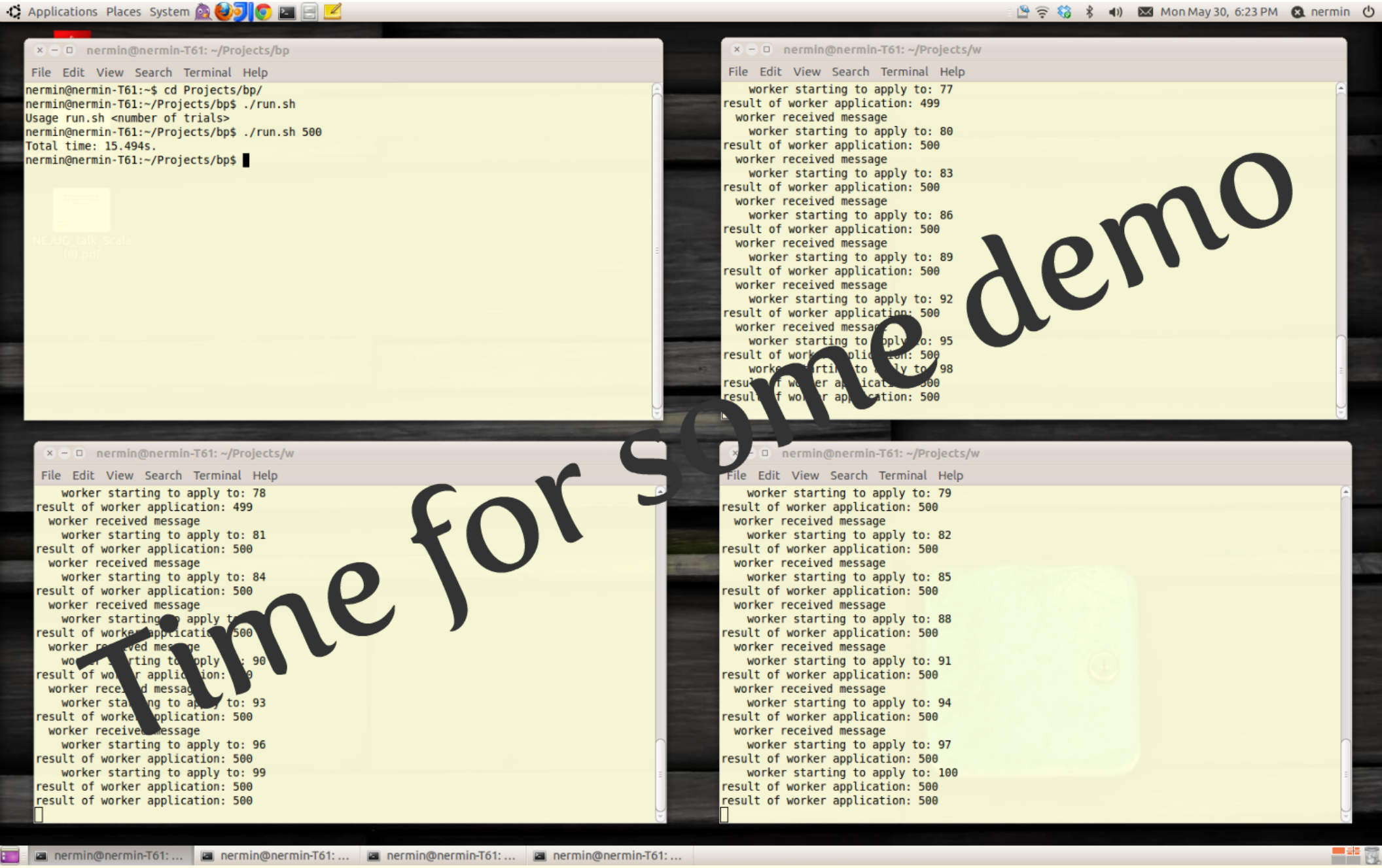
Actors

MapReduce

CS264


ZeroMQ

Parallel Collections


Applications Places System
Mon May 30, 6:23 PM nermin
nermin@nermin-T61: ~/Projects/bp
File Edit View Search Terminal Help
nermin@nermin-T61:~$ cd Projects/bp/
nermin@nermin-T61:~/Projects/bp$ ./run.sh
Usage run.sh <number of trials>
nermin@nermin-T61:~/Projects/bp$ ./run.sh 500
Total time: 15.494s.
nermin@nermin-T61:~/Projects/bp$
nermin@nermin-T61: ~/Projects/w
File Edit View Search Terminal Help
worker starting to apply to: 77
result of worker application: 499
worker received message
worker starting to apply to: 80
result of worker application: 500
worker received message
worker starting to apply to: 83
result of worker application: 500
worker received message
worker starting to apply to: 86
result of worker application: 500
worker received message
worker starting to apply to: 89
result of worker application: 500
worker received message
worker starting to apply to: 92
result of worker application: 500
worker received message
worker starting to apply to: 95
result of worker application: 500
worker starting to apply to: 98
result of worker application: 500
result of worker application: 500
nermin@nermin-T61: ~/Projects/w
File Edit View Search Terminal Help
worker starting to apply to: 78
result of worker application: 499
worker received message
worker starting to apply to: 81
result of worker application: 500
worker received message
worker starting to apply to: 84
result of worker application: 500
worker received message
worker starting to apply to: 90
worker starting to apply to: 93
worker received message
worker starting to apply to: 96
result of worker application: 500
worker starting to apply to: 99
result of worker application: 500
result of worker application: 500
nermin@nermin-T61: ~/Projects/w
File Edit View Search Terminal Help
worker starting to apply to: 79
result of worker application: 500
worker received message
worker starting to apply to: 82
result of worker application: 500
worker received message
worker starting to apply to: 85
result of worker application: 500
worker received message
worker starting to apply to: 88
result of worker application: 500
worker received message
worker starting to apply to: 91
result of worker application: 500
worker received message
worker starting to apply to: 94
result of worker application: 500
worker received message
worker starting to apply to: 97
result of worker application: 500
worker starting to apply to: 100
result of worker application: 500
result of worker application: 500
Time for some demo
nermin@nermin-T61: ...
nermin@nermin-T61: ...
nermin@nermin-T61: ...
nermin@nermin-T61: ...

```
private object Sender extends Actor {
  private val senderSocket = context.socket(ZMQ.PUSH)
  override def scheduler = DaemonScheduler

  def act = {
    senderSocket.bind("tcp://*:" + System.getProperty("request.bind"))
    while (true) {
      receive {
        case Send(payload) => senderSocket.send(payload, 0)
      }
    }
  }
}
```

```
private object Receiver extends Thread {
  private val receiverSocket = context.socket(ZMQ.PULL)

  override def run = {
    receiverSocket.bind("tcp://*:" + System.getProperty("response.bind"))
    while (true) {
      val ois = new ObjectInputStream(new ByteArrayInputStream(receiverSocket.recv(0)))
      try {
        val uuid = ois.readObject.asInstanceOf[UUID]
        val queue = gridOps.get(uuid)
        /*
                  if queue is null, it has already been removed, which means this result has already been received
                  so just discard it
                */
        if (queue != null) {
          queue.put((ois.readInt, ois.readObject))
        }
      } finally {
        ois.close
      }
    }
  }
}
```

```
class BoundedExecutor(bound: Int) {
  val semaphore = new Semaphore(bound)
  val executor = Executors.newFixedThreadPool(bound)

  def submitTask(command: Runnable) = {
    semaphore.acquire
    try {
      executor.execute(new Runnable {
        def run = {
          try {
            command.run
          } finally {
            semaphore.release
          }
        }
      })
    } catch {
      case ex: RejectedExecutionException => semaphore.release
    }
  }
}
```

```scala
private object Sender extends Actor {
  private val senderSocket = context.socket(ZMQ.PUSH)
  override def scheduler = DaemonScheduler

  def act = {
    senderSocket.bind("tcp://*:" + System.getProperty("request.bind"))
    while (true) {
      receive {
        case Send(payload) => senderSocket.send(payload, 0)
      }
    }
  }
}
```

```
private object Receiver extends Thread {
  private val receiverSocket = context.socket(ZMQ.PULL)

  override def run = {
    receiverSocket.bind("tcp://*:" + System.getProperty("response.bind"))
    while (true) {
      val ois = new ObjectInputStream(new ByteArrayInputStream(receiverSocket.recv(0)))
      try {
        val uuid = ois.readObject.asInstanceOf[UUID]
        val queue = gridOps.get(uuid)
        /*
                  if queue is null, it has already been removed, which means this result has already been received
                  so just discard it
                */
        if (queue != null) {
          queue.put((ois.readInt, ois.readObject))
        }
      } finally {
        ois.close
      }
    }
  }
}
```

```
class BoundedExecutor(bound: Int) {
  val semaphore = new Semaphore(bound)
  val executor = Executors.newFixedThreadPool(bound)

  def submitTask(command: Runnable) = {
    semaphore.acquire
    try {
      executor.execute(new Runnable {
        def run = {
          try {
            command.run
          } finally {
            semaphore.release
          }
        }
      })
    } catch {
      case ex: RejectedExecutionException => semaphore.release
    }
  }
}
```

Amazon high-cpu (c1.xlarge) instances
15,000 trials

4 node cluster:

# threads	time (sec)
1	39.06
2	27.30
4	19.30
6	24.49
8	34.48
16	34.87
32	34.31
64	37.41

5 node cluster:

# threads	time (sec)
1	29.65
2	20.29
4	14.88
6	18.90
8	26.05
16	23.68
32	25.57
64	27.56

Limitations:

- single parameter functions
- failure detection - timeout
- doesn't work for closures

thank you