WebObjects + Scala

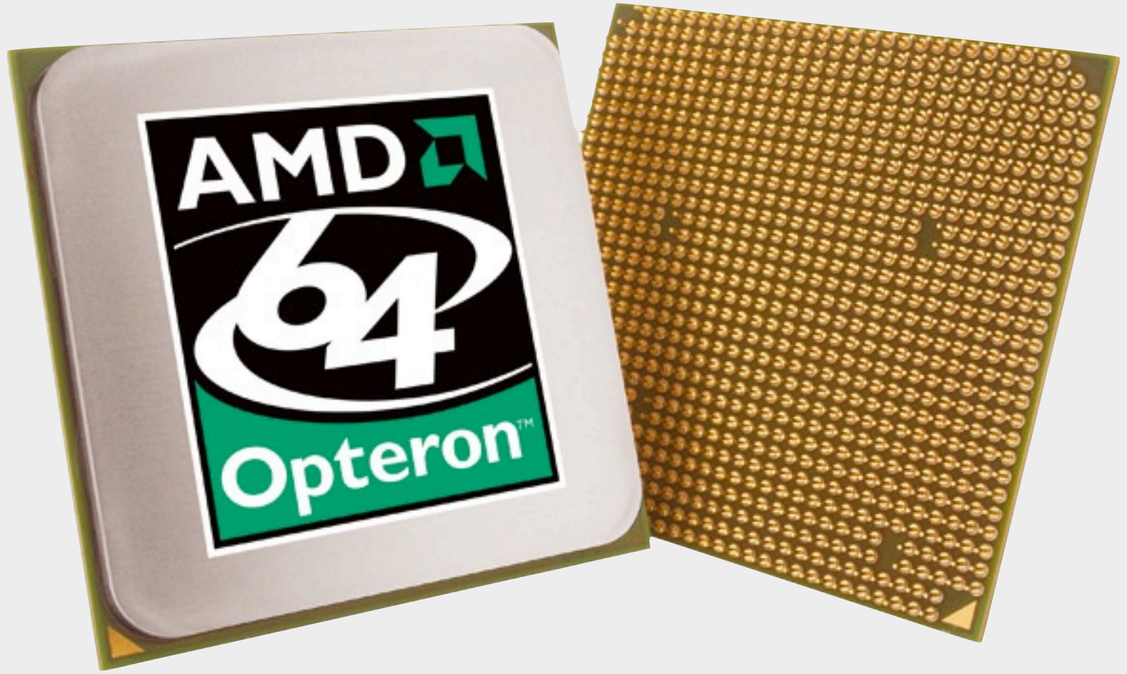
Building Concurrent WebObjects applications with Scala

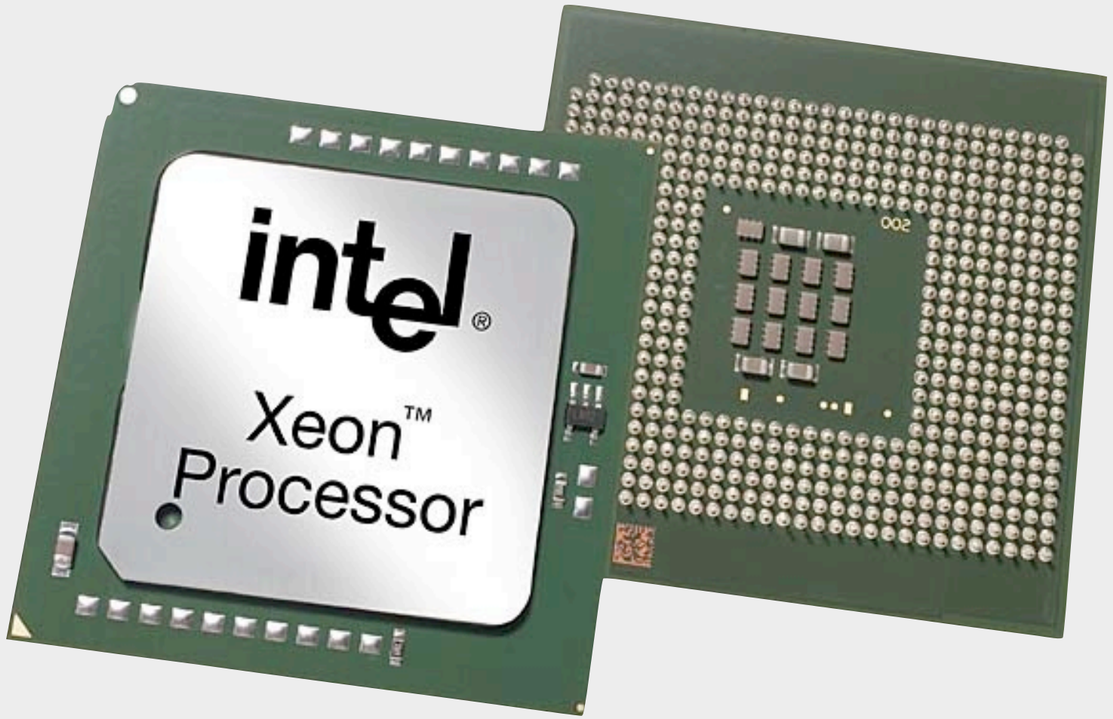
Ravi Mendis

Why Concurrent Programming?

2005

The year of Dual Core




intel
Xeon™
Processor

2010

Today

Entry-Level

		Cores	Threads
AMD Opteron		4	4
IBM Power7		4	16
Intel Xeon		4	8

High-End

		Cores	Threads
AMD Opteron		12	12
IBM Power7		8	32
Intel Xeon		6	12

2011

Tomorrow

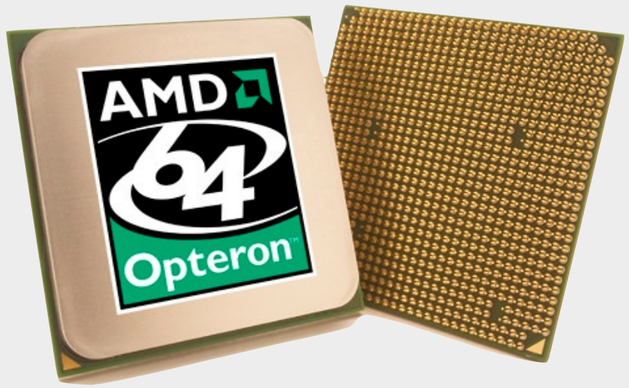
Roadmap

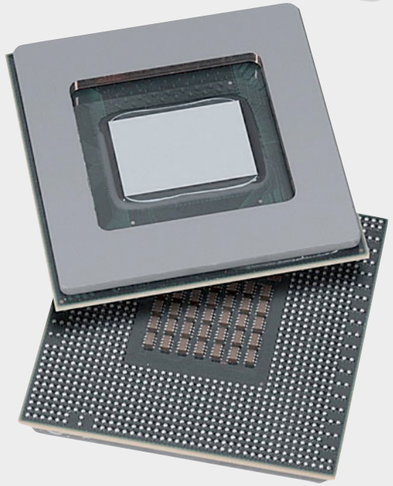
		Cores	Threads
AMD Opteron		16	16
IBM Power7		?	??
Intel Xeon		8	16

"By 2015 we will likely have over 100 cores on a many-core processing chip in our notebook computers."

- Computerworld

Welcome to the world of Multi-cores!





Q: How do we take advantage of multi-core processors?

A: Concurrent Programming

#1 Threads & Locks

"Concurrency is hard. It involves a lot of problems that are very difficult to think about and reason about and understand"

- Tim Bray co-inventor of XML

#1 Threads & Locks

- HARD to program
- HARD to scale
- Contentious

java.lang.IllegalArgumentException: Cannot determine primary key for entity ASCCarveout from row: {charge = 3027.00; claimID = 321839138; }
    at com.webobjects.eoaccess.EODatabaseChannel._fetchObject(EODatabaseChannel.java:348)
    at com.webobjects.eoaccess.EODatabaseContext._objectsWithFetchSpecificationEditingContext(EODatabaseContext.java:3071)
    at com.webobjects.eoaccess.EODatabaseContext.objectsWithFetchSpecification(EODatabaseContext.java:3195)
    at com.webobjects.eocontrol.EOObjectStoreCoordinator.objectsWithFetchSpecification(EOObjectStoreCoordinator.java:488)
    at com.webobjects.eocontrol.EOEditingContext.objectsWithFetchSpecification(EOEditingContext.java:4069)
    at er.extensions.eof.ERXEC.objectsWithFetchSpecification(ERXEC.java:1111)
    at com.webobjects.eoaccess.EODatabaseContext.objectForGlobalID(EODatabaseContext.java:4084)
    at com.webobjects.eocontrol.EOObjectStoreCoordinator.objectsForSourceGlobalID(EOObjectStoreCoordinator.java:634)
    at com.webobjects.eocontrol.EOEditingContext.objectsForSourceGlobalID(EOEditingContext.java:3923)
    at er.extensions.eof.ERXEC.objectsForSourceGlobalID(ERXEC.java:1169)
    at com.webobjects.eoaccess.EODatabaseContext._fireArrayFault(EODatabaseContext.java:4245)
    at com.webobjects.eoaccess.EOAccessArrayFaultHandler.completeInitializationOfObject(EOAccessArrayFaultHandler.java:77)
    at com.webobjects.eocontrol._EOCheapCopyMutableArray.willRead(_EOCheapCopyMutableArray.java:37)
    at com.webobjects.eocontrol._EOCheapCopyMutableArray.count(_EOCheapCopyMutableArray.java:86)
    at com.mpv.evaluation.ClaimEvaluator.validateClaim(ClaimEvaluator.java:398)
    ...

...Deadlock!

BBC2, Top Gear - Series 15, Episode 1 - June 27 '10

#2 Actor Model

(A Share NOTHING Model)

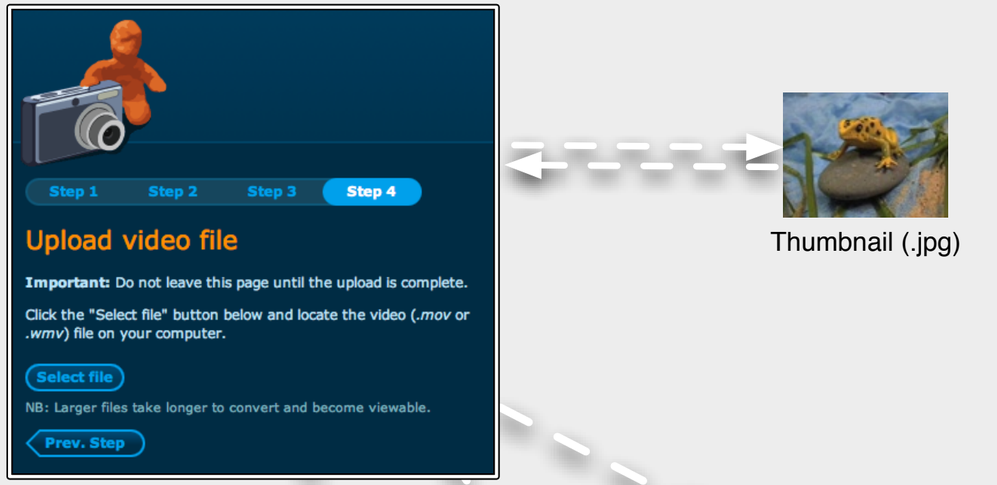
Slowmation

Demo

html5 <video>

```
<video poster="/slowmation/screenshots/0/2/4/425.jpg">
    <source type="video/ogg" src="/slowmation/videos/6/d/8/423.ogg" />
    <source type="video/mp4" src="/slowmation/videos/d/6/4/424.mp4" />
</video>
```

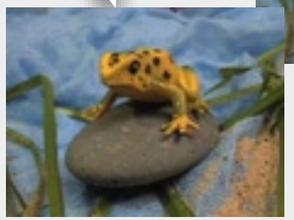
HTML5 Video Conversion



Thumbnail (.jpg)

Screenshot (.jpg)

Video (.mp4)

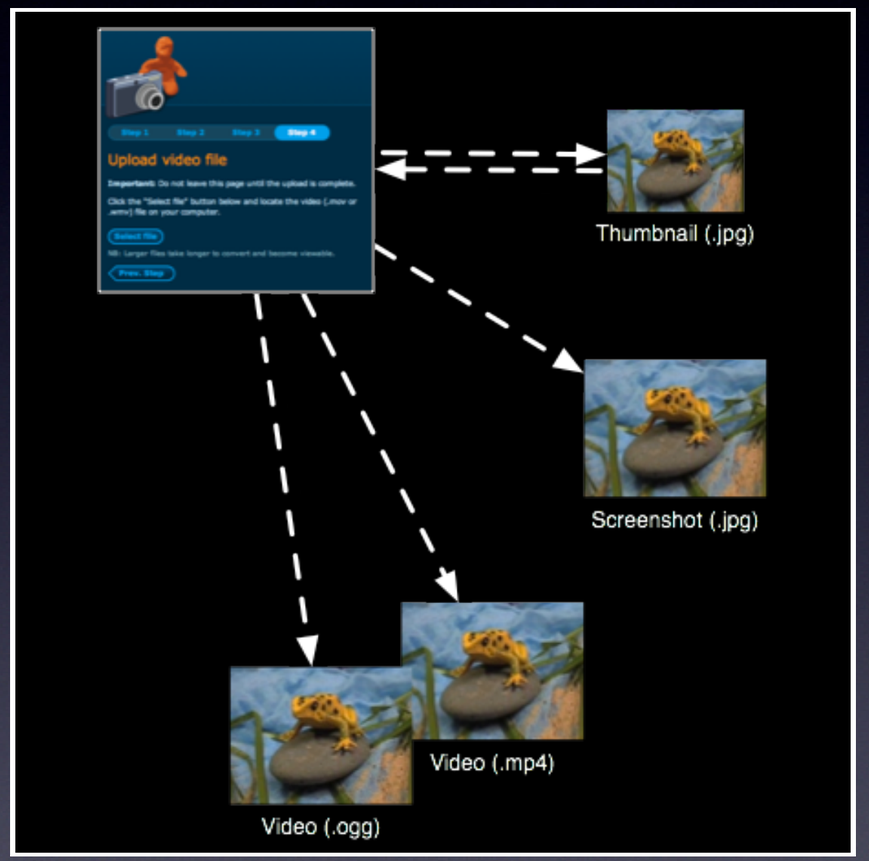


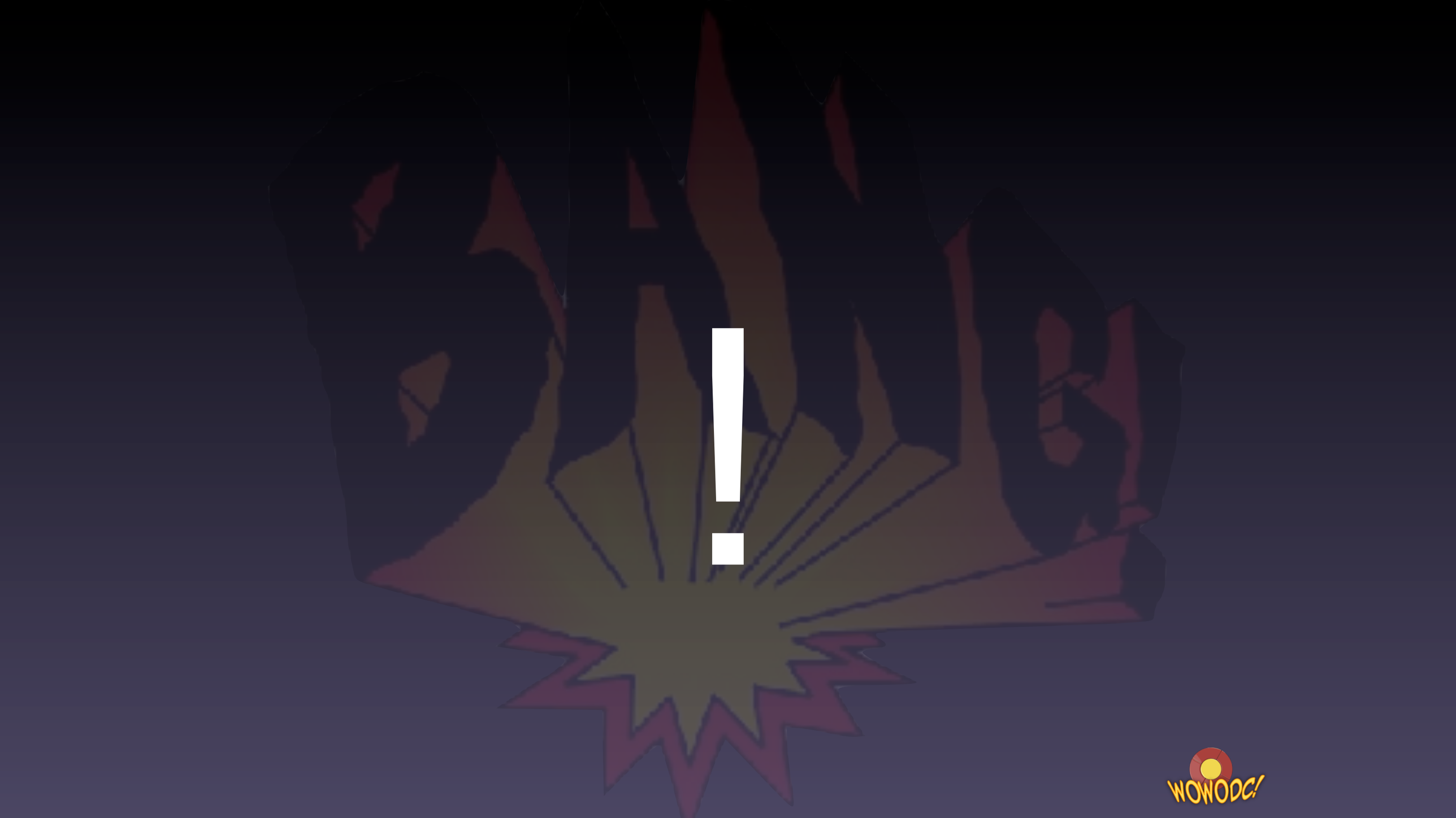
Video (.ogg)

Slowmation Actor

(Video Processor)

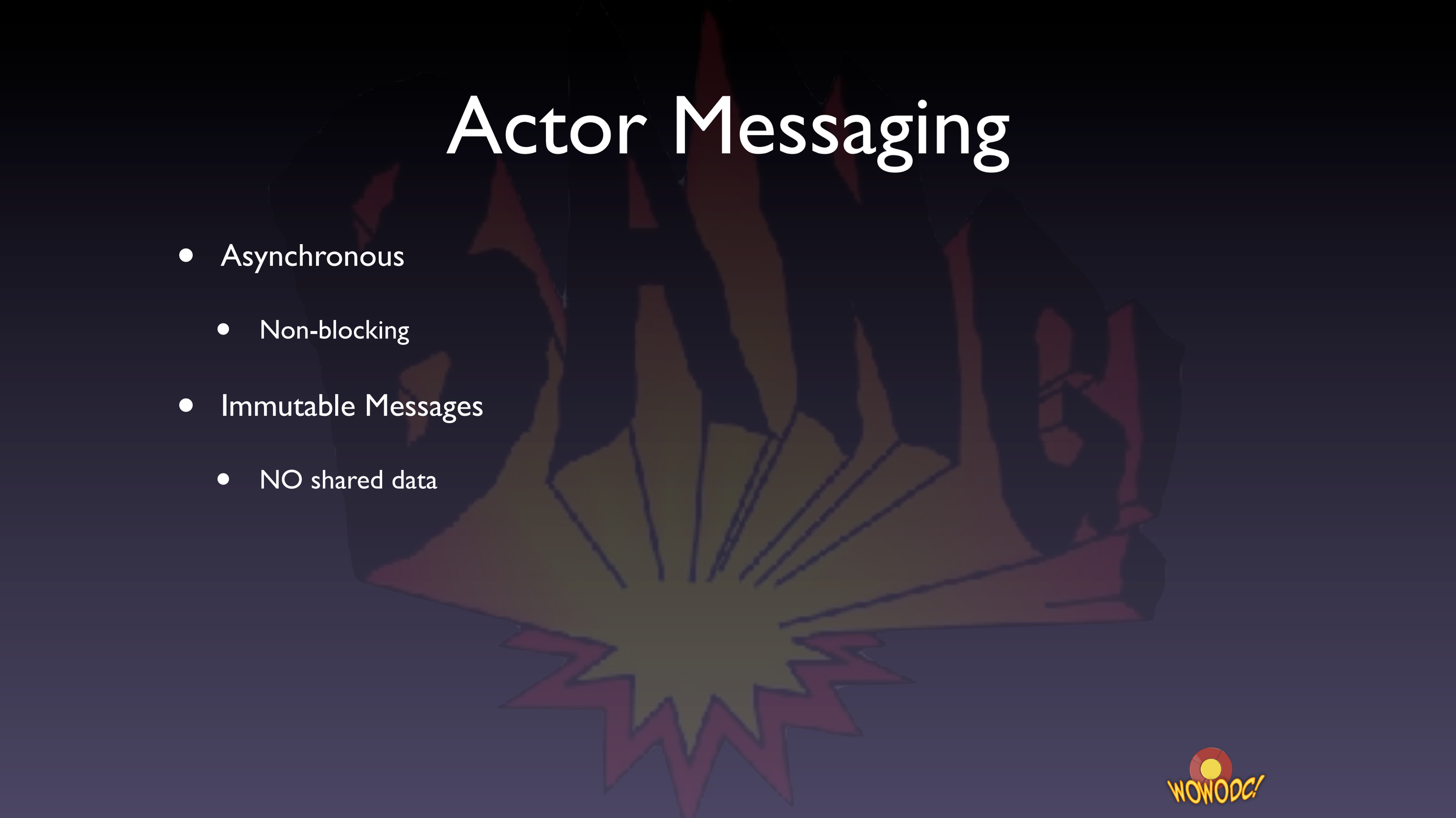
- actor ! THUMBNAIL
- actor ! GRAB
- actor ! CONVERT2H264
- actor ! CONVERT2OGG





Actor Messaging

- Asynchronous
  - Non-blocking
- Immutable Messages
  - NO shared data




Scala


WOWODC!

Scala

- Immutable/Mutable datatypes
- Anonymous Functions (~Closures)
- No Static variables and methods
- Extensible

Scala

```
val greeting: String = “Hello World”;
```

Scala

```
val greeting = “Hello World”;
```

Scala

```
val greeting = “Hello World”
```

Scala

```
val greeting = “Hello World”    // immutable
var response = new String()     // mutable

response = “Hey!”
```

Java - Static Vars

```
public class _Talent extends EOGenericRecord {
     public static final String ENTITY_NAME   = "Talent";
}
```

Scala - Companion Object

```
object Talent extends EOGenericRecord {
     val ENTITY_NAME = "Talent"
}
```

Thread-Safe

Scala - Pattern Matching

```
case a => {
   ...
}
```

Scala - Pattern Matching

```
try {
     var epi: EditPageInterface = D2W.factory.editPageForNewObjectWithEntityNamed(entityName, session)
} catch {
     case e: IllegalArgumentException => {
            var epf: ErrorPageInterface = D2W.factory.errorPage(session)
            epf.setMessage(e.toString)
     }
}
```

Scala - Case Classes

```
case class PING
case class PONG
```

Scala - Case Classes

```
actor ! PING
actor ! PONG
```

Scala - Case Classes

```
case PING => {
    ...
}

case PONG => {
    ...
}
```

Scala - Anonymous Functions

```
x => x^2
```

Scala - Anonymous Functions

```
x, y => x + y
```

Scala - Anonymous Functions

```
x, y => {x + y}
```

Scala - Anonymous Functions

```
case (PING => {...})

case (PONG => {...})
```

λ - *Expressions*

Scala Actors - Example

```
case class THUMBNAIL(filepath: String)      // Actor msg case class

val processor = actor {
     loop {
          react() {
               case THUMBNAIL(filepath: String) => {
                   ...
                }
          }
     }
}

// asynchronous
processor ! THUMBNAIL(“/Library/WebServer/Documents/slowmation/screenshots/0/2/4/422.jpg”)
```

#2 Actor Model

- EASY to program
- Scalable
- NO Deadlocks!

Q: Why can Scala be used with WebObjects?

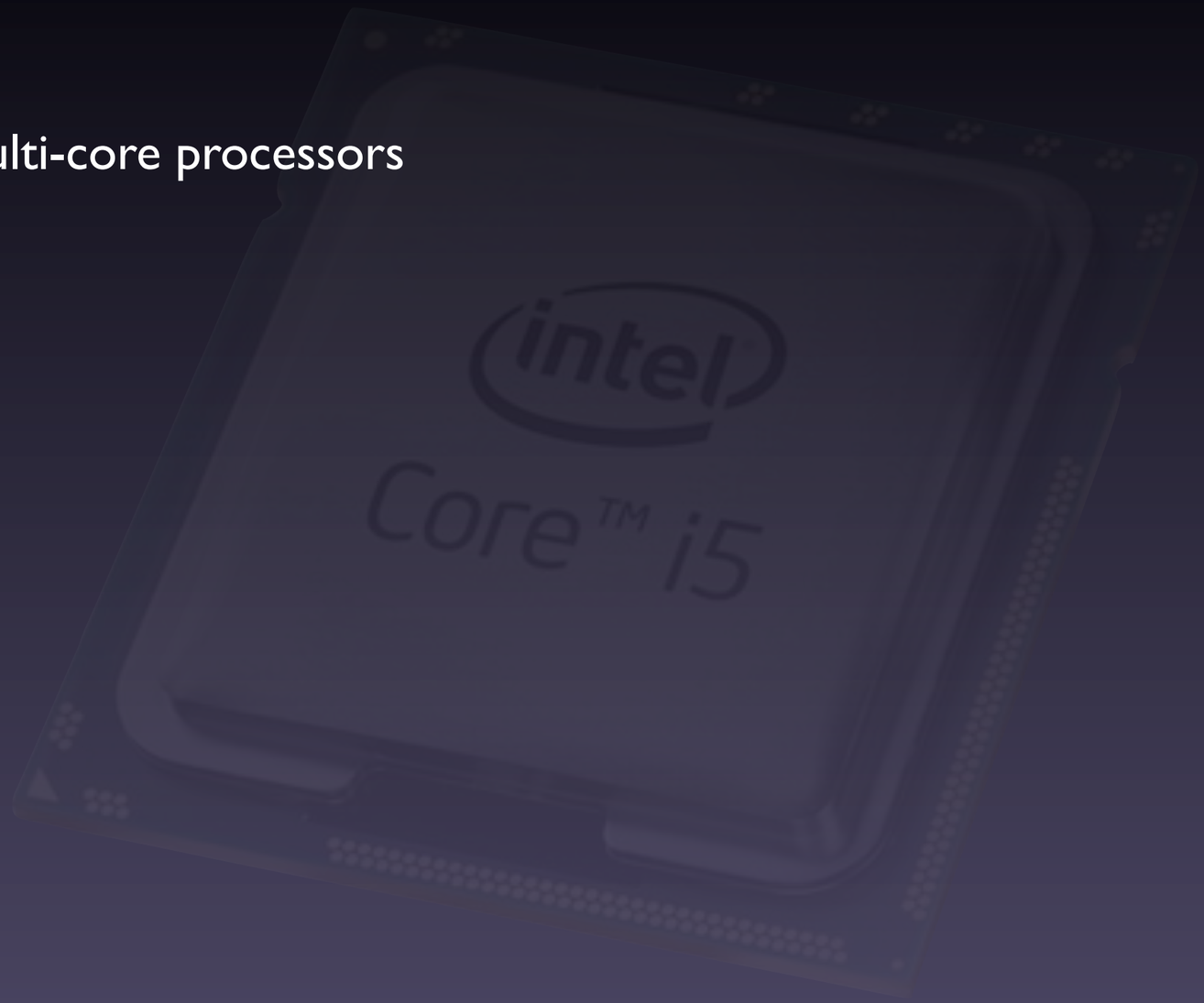
A: Scala compiles to Java Byte-code

WebObjects + Concurrency

Why?

Performance

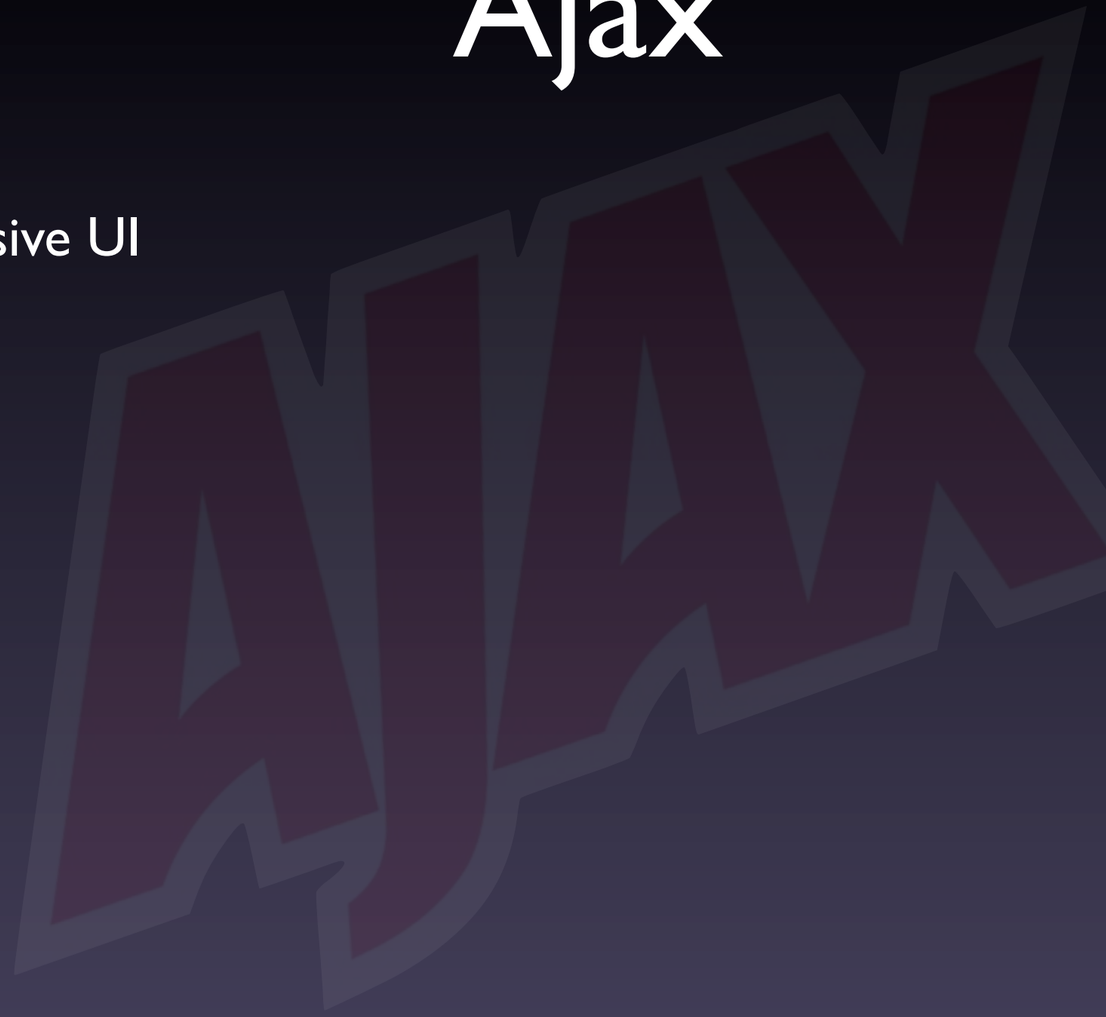
- Exploit multi-core processors



WOWODC!

Ajax

- More responsive UI



How?

Properties

```
WOAllowsConcurrentRequestHandling=true
```

Java - Synchronize Static Variables

```
private static String greeting = “Hello”;        // private

public void setGreeting(String aGreeting) {
    synchronized(greeting) {                     // synchronized block
        greeting = aGreeting;
    }
}
```

Free!

WO - What is Shared?

- Application
- Session (concurrent WO)

Use ERXEC

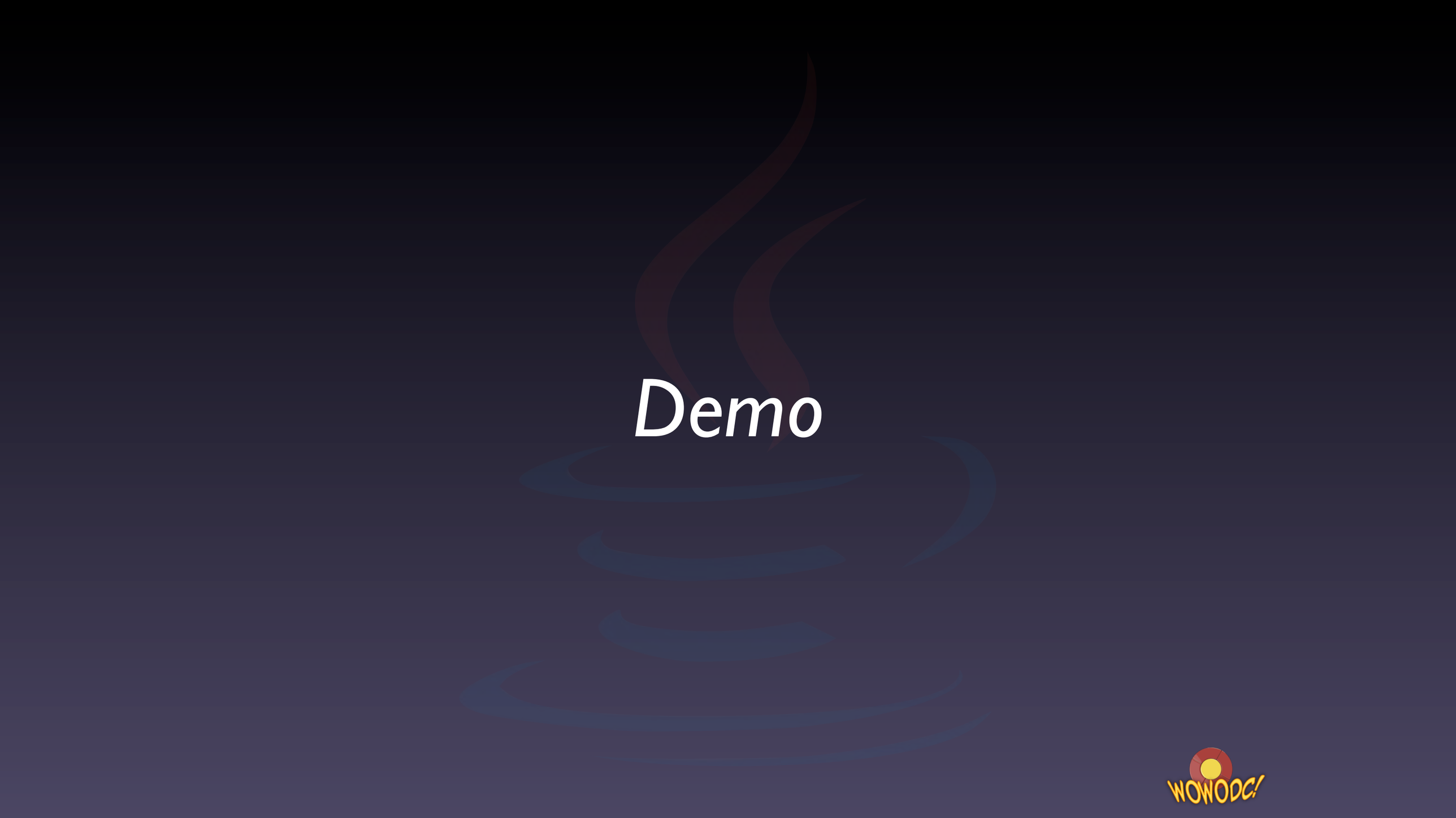
- Project Wonder
- Automatic lock/unlock handling

Properties

```
er.extensions.ERXEC.safeLocking=true
```

Debugging

Demo



Properties

```
-Dcom.sun.management.jmxremote=true
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false

// Specific port
-Dcom.sun.management.jmxremote.port=20102
```

Bottlenecks

EOF - Bottleneck

- Shared Object-cache
  - Antithesis to Share Nothing model
  - Uses single database connection
  - Single-Threaded

WOWODC!

#3 STM

(Software Transactional Memory)

EOF + Scala

- DOESN'T work in Scala Actors!
- 

Solutions

- Use Raw SQL
- Alternative database access
  - Squeryl
  - ...

WOWODC!

Squeryl

Demo

Squeryl

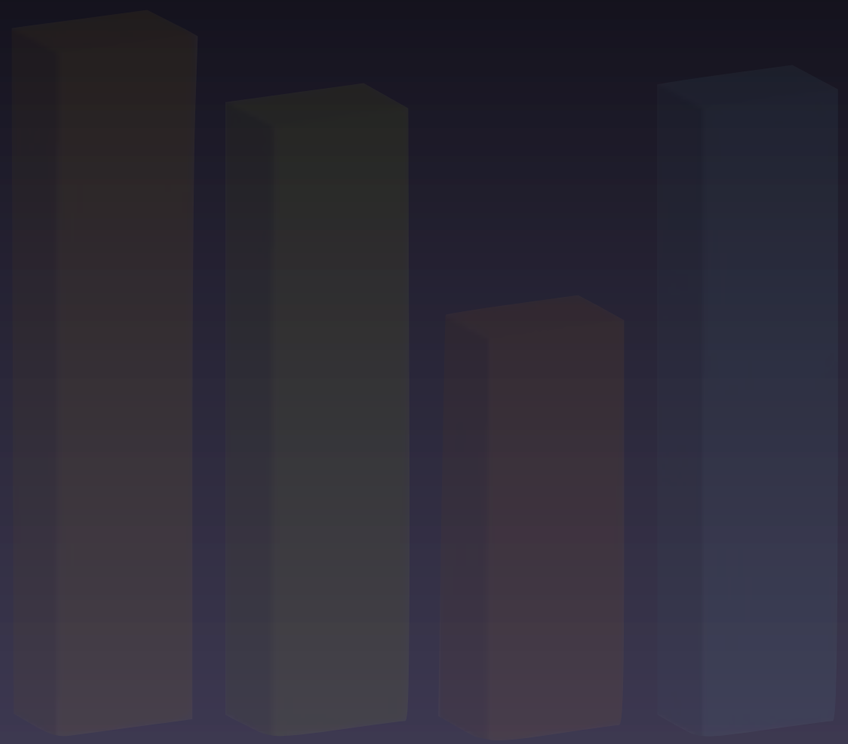
- POSOs
- Actor Compatible
- Concurrent

“Scala as a concurrent programming language is powerful, *safe* and *easy-to-use*”

Benchmarks

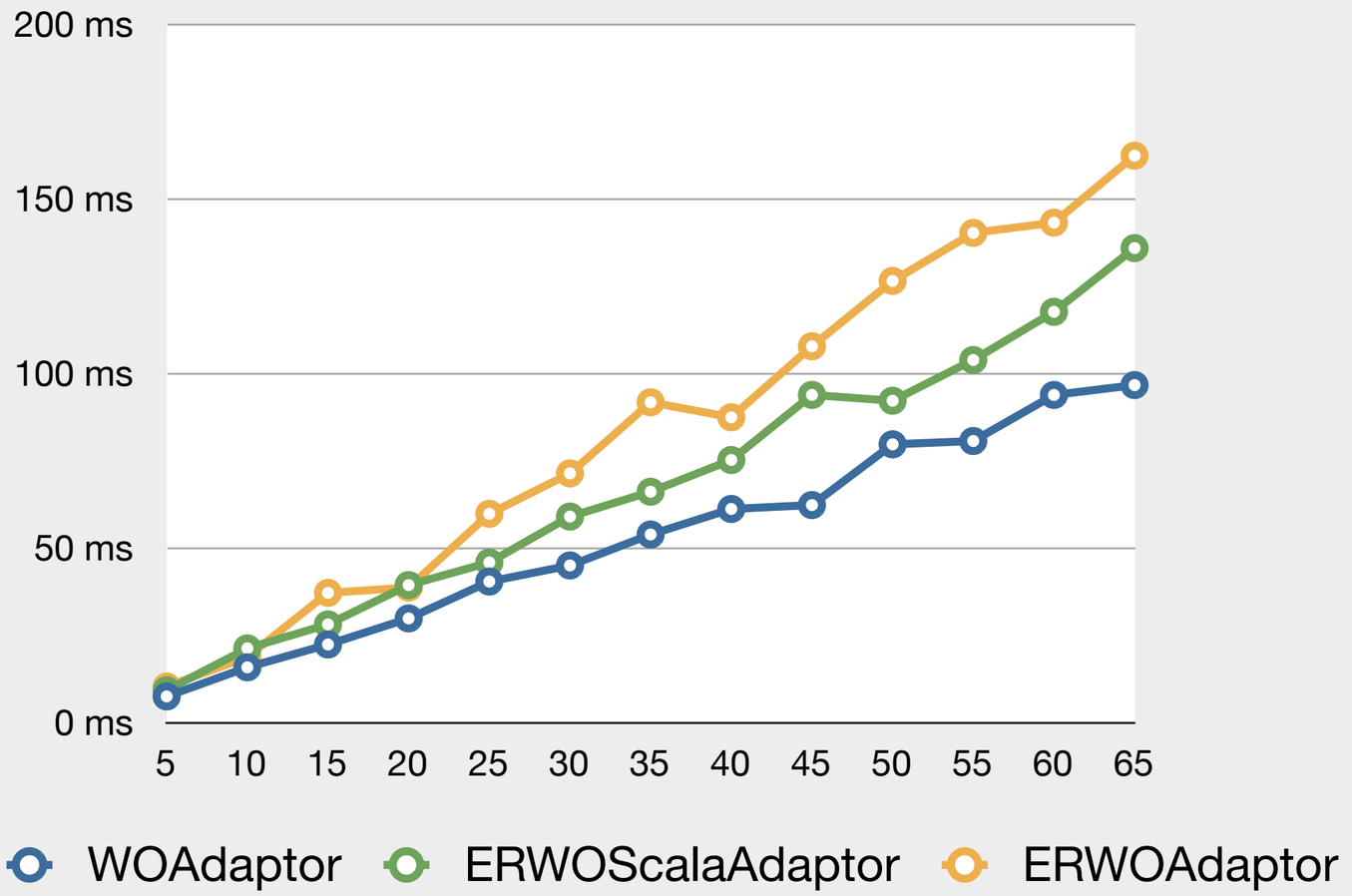
The Test

- ERWOAdaptor (Mina)
  - WO Worker as Actor
- Apache bench
  - c = 5...65
  - n = 500

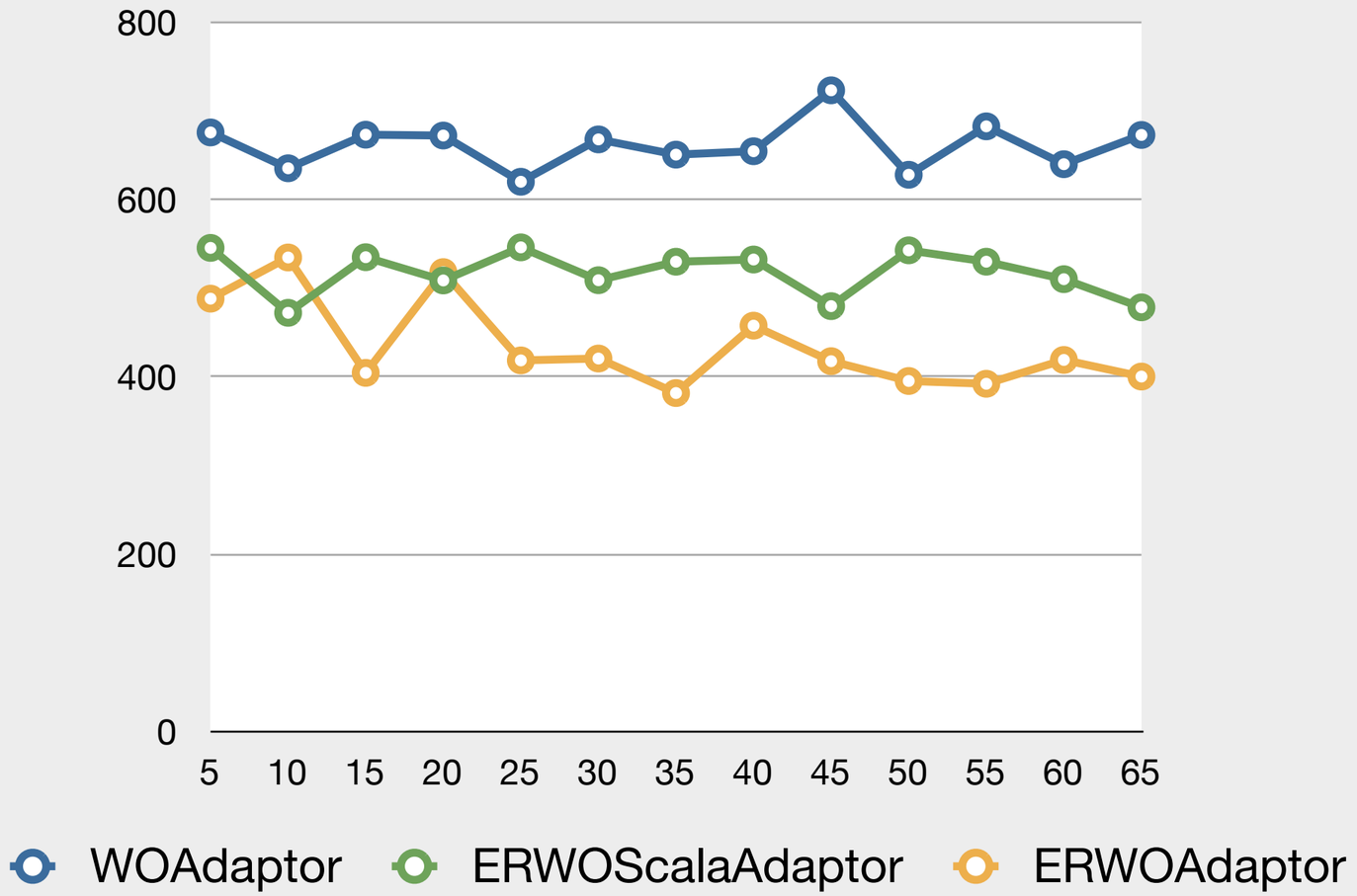


Results

Time per req. (mean)



Requests per second



"Scala as a concurrent programming language is *powerful*, *safe* and *easy-to-use*"

The R&D Imperative

In life long love
The key
Is R&D

The ladder of invention
Has eternal slide extension
Rung by rung
You'll climb to heaven above.

Constant innovation
Guarantees a satiation
Of that ever changing want
And deepening need

Nourishment and care
To think anew…
Makes passions flair
And lets a culture breed
Between the sheets…

- by *Emma Ahmad*

Q&A

References

- What will YOU do with 100 cores?
  http://www.computerworld.com.au/article/354261/
- WebObjects with Scala
  http://wiki.objectstyle.org/confluence/display/WO/WebObjects+with+Scala
- Case Study: Slowmation
  http://slowmation.uow.edu.au
- Source: ERWOScalaAdaptor
  http://services.wocommunity.org/wowodc/ERWOScalaAdaptor