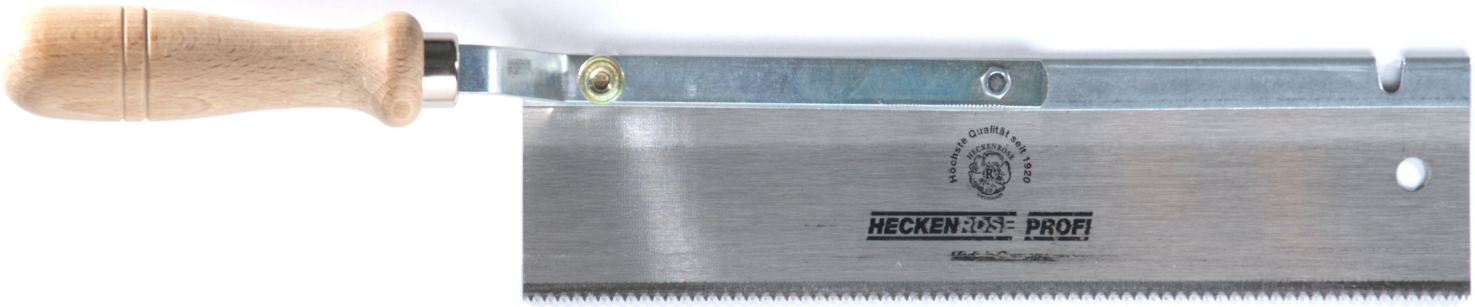
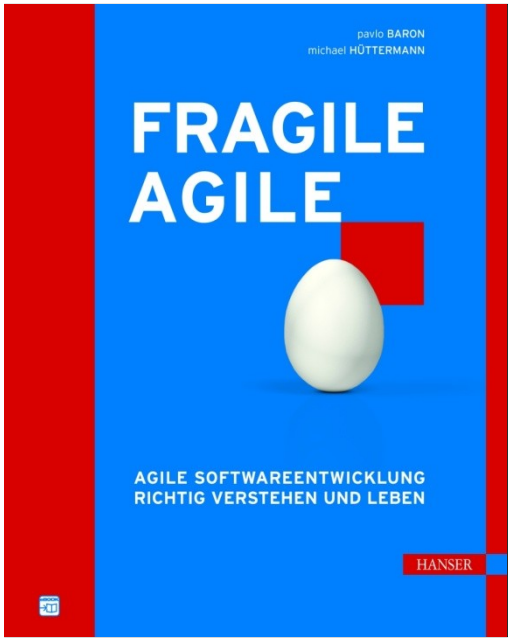
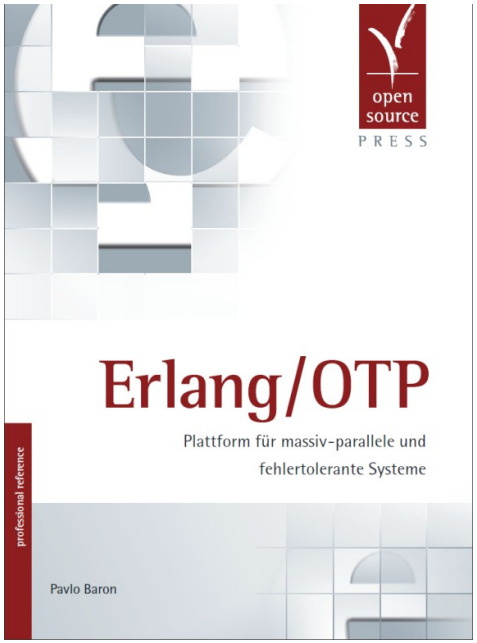
What can be done with

but should better be done with







Geek's
Guide
To The Working Life

Pavlo Baron

pavlo.baron@codecentric.de
@pavlobaron

Hey, dude.

What sort of application
should be optimally implemented
using your language?

.NET dude:

win*

Python dude:

sci*

Ruby dude:

cool*

Clojure, Groovy, Scala dude:

java*

Putonghua dude:

谢谢

Java dude:

Erlang dude:

```
fit -> do();

_ -> badarg.
```

This dude in front of you

is very picky.

So, let' assume that:

=

=

```
List<Integer> l =
    Arrays.asList(1, 2, 3, 4, 5);
List<Integer> r =
    new ArrayList<Integer>();
    for (int i : l) {
      r.add(i * 2);
    }
```

```
List<Integer> l =
   Arrays.asList(1, 2, 3, 4, 5);
Iterable<Integer> t =
   Iterables.transform(l,
      new Function<Integer, Integer>() {
         public Integer apply(Integer i) {
            return I * 2;
         }
      });
```

And this is just a simple map.
It gets even worse
with **filter** and **fold**


```
[X * 2 || X <- lists:seq(1, 5)].
```

so what?

```
List<Integer> l = Arrays.asList(1, 2, 3, 4, 5); Iterable<Integer> t = Iterables.transform(l, new Function<Integer, Integer>() {public Integer apply(Integer i) {return l * 2;}});
```

your colleagues will love it


Community
Education
What's New
Samples
Tutorials
Workbench
Web Services
Web
XML Tools
Server Tools
Java EE
JSF
eclipse
Java EE
ide

That works!

```
synchronized (this) {
      if (!crawledSites.contains(site)) {
            linkedSites.add(site);
      }
 }

…

 public class CountingSemaphore {
      private int signals = 0;
      public synchronized void take() {
            this.signals++;
            this.notify();
      }

      public synchronized void release()
                                    throws InterruptedException {
            while (this.signals == 0) wait();
            this.signals--;
      }
}
```


```
1> A = 5.
5
2> A = 10.
** exception error: no match of
right hand side value 10
3>
```

so what?

The simplest way to avoid problems with concurrency is to share only immutable data between threads. Immutable data is data which can not be changed.

(from a Java concurrency tutorial)

Immutable class:

- all its fields are final
- class declared as final
- “this” reference is not allowed to escape during construction

Any fields which refer to mutable data objects:

- are private
- have no setter method
- are never directly returned of otherwise exposed to a caller
- if they are changed internally in the class this change is not visible and has no effect outside of the class

That works!

Java - erl/src/T.java - Eclipse

File Edit Run Source Refactor Navigate Search Project Window

Java

T.java

```
public class T {
    public static void main(String[] args) {
        T t = new T();
        t.rec(10);
    }

    protected void rec(int i) {
        rec(i);
    }
}
```

Pr @ J D T O S C L J

<terminated> T [Java Application] /usr/lib/jvm/java-7-oracle/bin/java (N

```
Exception in thread "main" java.lang.StackOverflowError
        at T.rec(T.java:9)
        at T.rec(T.java:9)
        at T.rec(T.java:9)
        at T.rec(T.java:9)
        at T.rec(T.java:9)
        at T.rec(T.java:9)
        at T.rec(T.java:9)
        at T.rec(T.java:9)
        at T.rec(T.java:9)
        at T.rec(T.java:9)
        at T.rec(T.java:9)
        at T.rec(T.java:9)
```

Writable Smart Insert 9 : 12


```
1> F = fun(F) -> F(F) end.
#Fun<erl_eval.6.80247286>
2> F(F).

.....hours later.....

BREAK: (a)bort (c)ontinue (p)roc info
(i)nfo (l)oaded (v)ersion (k)ill (D)b-tables
(d)istribution

a
```

so what?

```
...
for (paramFiber = recursion1.doit__1(paramEProc, paramEObject);
        paramFiber == EProc.TAIL_MARKER;
        paramFiber = (EFun)localFiber.getCallee())
    {
      S_O localS_O;
      switch (localFiber.up())
      {
      case 2:
        paramEProc.tail.go(paramEProc, localFiber.down());
        localS_O = new S_O();
        localS_O.f0 = paramEProc;
        localFiber.setState(localS_O, this, 1);
        return null;
      case 3:
        null;
        return null;
      case 1:
        localS_O = (S_O)localFiber.curState;
        paramEProc = (EProc)localS_O.f0;
      case 0:
      }
    }
...
```

That works!

Java - erl/src/B.java - Eclipse

File Edit Run Source Refactor Navigate Search Project Window

Java

T.java B.java

```
public class B {

    protected void doit(Integer i) {
        System.out.println("integer");
    }

    protected void doit(String s) {
        System.out.println("string");
    }

    public static void main(String[] args) {
        B b = new B();
        b.doit(15);
        b.doit("whatever");
    }
}
```

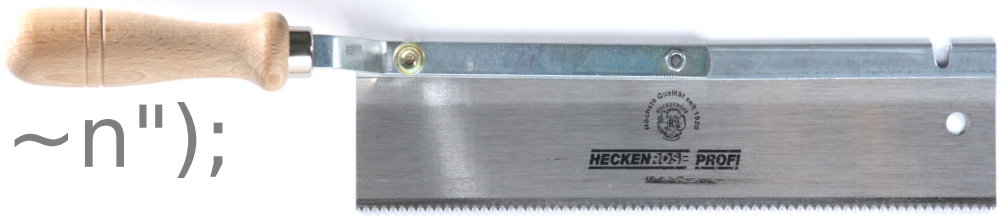
Pr @ J D T O S C L J

<terminated> B [Java Application] /usr/lib/jvm/java-7-oracle/bin/java (N

```
integer
string
```

Writable Smart Insert 17 : 1

```
1> F = fun([_|_]) ->
1> io:format("string~n");
1> (B) ->
1> io:format("integer~n")
1> end.
#Fun<erl_eval.6.80247286>
2> F(15).
integer
ok
3> F("whatever").
string
ok
4>
```



so what?

Java - erl/src/B.java - Eclipse

File Edit Run Source Refactor Navigate Search Project Window

T.java B.java

```
public class B {

    protected void doit(Object o) {
        if (o instanceof Integer) {
            System.out.println("integer");
        } else if (o instanceof String) {
            System.out.println("string");
        }
    }

    public static void main(String[] args) {
        B b = new B();
        b.doit(15);
        b.doit("whatever");
    }
}
```

<terminated> B [Java Application] /usr/lib/jvm/java-7-oracle/bin/java (N

```
integer
string
```

Writable Smart Insert 17 : 1

That works!

```
try {
    // Create a new class loader with the directory
    ClassLoader cl = new URLClassLoader(urls);

    // Load in the class
    Class cls = cl.loadClass("MyReloadableClassImpl");

    // Create a new instance of the new class
    myObj = (MyReloadableClass)cls.newInstance();
} catch (IllegalAccessException e) {
} catch (InstantiationException e) {
} catch (ClassNotFoundException e) {
}
```

Excessive class-reloading using ClassLoader hierarchies is expensive and leads to JVM instance fatigue

That's why it's strongly recommended to do hot deployment in app servers


```
% hot_swap:sum(Num)
% adds 1 to Num
1> c(hot_swap).
{ok,hot_swap}
2> hot_swap:sum(1).
2

...
% hot_swap.erl has changed.
% now it adds 2 to Num

...
3> c(hot_swap).
{ok,hot_swap}
4> hot_swap:sum(1).
3
5>
```


```
code_change(OldVsn, State, Extra) ->
    ...code to convert state (and more) during
    code change...
    {ok, NewState}.
```

so what?

Java™

OSGi™ Alliance

JVM HotSwap is limited to method bodies.

OSGi is invasive and state-unaware

so what?

JRebel

LiveRebel

your managers will love *Rebel in production

your ops will love *Rebel in production

That works!

Java - erl/src/C.java - Eclipse

File Edit Run Source Navigate Search Project Refactor Window Help

C.java

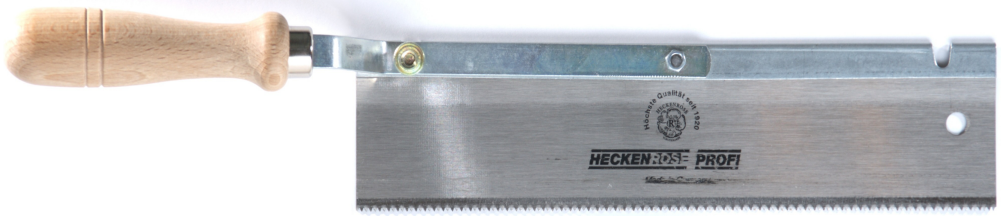
```
public class C {
    protected Integer divide(Integer A, Integer B) {
        return A / B;
    }

    public static void main(String[] args) {
        C c = new C();
        c.divide(5, 0);
    }
}
```

<terminated> C [Java Application] /usr/lib/jvm/java-7-oracle/bin/java (Mar 11, 2012

```
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at C.divide(C.java:3)
        at C.main(C.java:8)
```

Writable Smart Insert 11 : 1

```
1> A = 20.
20
2> self().
<0.32.0>
3> 5 / 0.
** exception error: bad argument in an
arithmetic expression
     in operator  '/'/2
        called as 5 / 0
4> self().
<0.36.0>
5> A.
20
6>
```



so what?

Java - erl/src/C.java - Eclipse

File Edit Run Source Navigate Search Project Refactor Window Help

C.java

```
public class C {
    protected Integer divide(Integer A, Integer B) {
        Integer C = 0;
        if (A instanceof Integer &&
            B instanceof Integer &&
            (Integer)A * 10 > (Integer)B &&
            (Integer)B != 0) {
            try {
                C = (Integer)A / (Integer)B;
            } catch (ArithmeticException a) {
                //hm, what now?
            } catch (WrongStellarConstellationException w1) {
                //move the stars
            } catch (WrongSunPositionException w2) {
                //move after the sun
            } catch (Angle45DegreeToHorizon w3) {
                //run for your life
            } catch (Exception e) {
                //just for the case
            } catch (Throwable t) {
                //if smth. weird is going on
            }
        }

        return C; //great, C = 0 in case of error
    }
}
```

Writable Smart Insert 28 : 1

That works!

Memory Analysis - /home/pb/java_pid24156.0001.hprof - Eclipse

File Edit Run Navigate Search Project Refactor Window Help

Memory ... Java

java_pid24156.0001.hprof

Overview OQL

```
select * from M$M2
```

Class Name	Shallow Heap	Retained Heap
<Regex>	<Numeric>	<Numeric>
M$M2 @ 0x7acdaa640	32	608
<class> class M$M2 @ 0x7acdaa370	0	0
this$0 M @ 0x7acda91a0	16	16
m java.util.HashMap @ 0x7acdaa8d8	64	576
Σ **Total: 3 entries**		

M.java

```
import java.util.HashMap;
import java.util.Map;

public class M {

    public class M2 {
        Map<Integer, String> m = new HashMap<Integer, String>();
        public void add(Integer i, String s) {
            m.put(i, s);
        }
    }

    public static void main(String[] args) {
        M m = new M();
        M2 m2 = m.new M2();
        m2.add(1, "One");
        m2.add(2, "Two");
        m2.add(3, "Three");
        while (true) {
        }
    }
}
```


```
1> bit_size(<<3:5>>).
5
2>
```

so what?

Memory Analysis - /home/pb/java_pid25338.0001.hprof - Eclipse

File Edit Run Navigate Search Project Refactor Window Help

Memory ... Java

java_pid25338.0001.hprof

Overview OQL

```
select * from M
```

Class Name	Shallow Heap	Retained Heap
<Regex>	<Numeric>	<Numeric>
M @ 0x7acda9968	32	192
<class> class M @ 0x7acda9170	0	0
i_s int[3] @ 0x7acdab3d0	32	32
s_s byte[3][] @ 0x7acdab3f0	48	128
Total: 3 entries		

M.java

```
import java.util.Arrays;

public class M {

    int[] i_s = {};
    byte[][] s_s = {};

    public void add(Integer i, String s) {
        i_s = Arrays.copyOf(i_s, i_s.length + 1);
        i_s[i_s.length - 1] = i;
        s_s = Arrays.copyOf(s_s, s_s.length + 1);
        s_s[s_s.length - 1] = s.getBytes();
    }

    public static void main(String[] args) {
        M m = new M();
        m.add(1, "One");
        m.add(2, "Two");
        m.add(3, "Three");
        while (true) {
        }
    }
}
```

That works!

```
QueueConnectionFactory connFactory =
      new QueueConnectionFactory();
QueueConnection conn = connFactory.createQueueConnection();
QueueSession session = conn.createQueueSession(false,
      Session.AUTO_ACKNOWLEDGE);
Queue q = new Queue("world");
QueueSender sender = session.createSender(q);
TextMessage msg = session.createTextMessage();
msg.setText("Hello there!");
sender.send(msg);

QueueReceiver receiver = session.createReceiver(q);
conn.start();
Message m = receiver.receive();
if (m instanceof TextMessage) {
      TextMessage txt = (TextMessage) m;
}

session.close();
conn.close();
```

```
register(serv, spawn(?MODULE, loop, [])),
serv ! {self(), “Hello there!”},
receive
    {_Pid, Msg} ->
    ...
end.
```

...

```
loop() ->
    receive
        {From, Txt} ->
            ...
            loop();
```


```
$ erl +P 134217727

Erlang R14B04 (erts-5.8.5) [source] [64-bit]
[smp:8:8] [rq:8] [async-threads:0] [hipe]
[kernel-poll:false]

Eshell V5.8.5  (abort with ^G)
1> erlang:system_info(process_limit).
134217727
2>
```

so what?

```
import kilim.Mailbox;
import kilim.Pausable;
import kilim.Task;

public class SimpleTask extends Task {
    static Mailbox<String> mb = new Mailbox<String>();

    public static void main(String[] args) throws Exception {
        new SimpleTask().start();
        Thread.sleep(10);
        mb.putnb("Hello ");
        mb.putnb("World\n");
        mb.putnb("done");
    }

    public void execute() throws Pausable {
        while (true) {
            String s = mb.get();
            if (s.equals("done")) break;
            System.out.print(s);
        }
       System.exit(0);
    }
}
```

your ops will love this in production

That works!

```
QueueConnectionFactory connFactory =
      new QueueConnectionFactory();
QueueConnection conn = connFactory.createQueueConnection();
QueueSession session = conn.createQueueSession(false,
      Session.AUTO_ACKNOWLEDGE);
Queue q = new Queue("world");
QueueSender sender = session.createSender(q);
TextMessage msg = session.createTextMessage();
msg.setText("Hello there!");
sender.send(msg);

QueueReceiver receiver = session.createReceiver(q);
conn.start();
Message m = receiver.receive();
if (m instanceof TextMessage) {
      TextMessage txt = (TextMessage) m;
}

session.close();
conn.close();
```

```
{serv, 'serv@pc'} ! {self(), “Hello there!”},
receive
    {_Pid, Msg} ->
    ...
end.
```


```
...

loop() ->
    receive
        {From, Txt} ->
            ...
            loop();
```

so what?

```
import org.gridgain.grid.*;
import org.gridgain.grid.gridify.*;
import org.gridgain.grid.gridify.aop.spring.*;

public final class GridifyHelloWorldSessionExample {
    private GridifyHelloWorldSessionExample() { //ensure singleton }

    @Gridify(taskClass = GridifyHelloWorldSessionTask.class, timeout = 3000)
    public static int sayIt(String phrase) {
        System.out.println(phrase);
        return phrase.length();
    }
   public static void main(String[] args) throws GridException {
        if (args.length == 0) {
                GridFactory.start();
        }
        else {
                GridFactory.start(args[0]);
        }
        try {
                int phraseLen = sayIt("Hello World");
                System.out.println(„number of characters is '" + phraseLen + "'.");
        } finally {
                GridFactory.stop(true);
        }
    }
}
```

your ops will love this in production

That works!

One GC per OS process.
Complex generation management.
Different GC stategies.
Stop the world situations possible

One GC per Erlang process.
Simple generation management.
(Mostly) one GC stategy.
Stop the world situations unlikely.

so what?


elixir
M2U51264DS8HC3G-5T
512MB DDR-400MHz-CL3
PC3200U-30331
Warranty Void If Removed
0542.MN05A1107A.5YM0410N08.XX.CN
elixir

That works!

Hey, dude.

So do you imply Erlang is better than Java for everything?

Erlang dude:

```
body() -> [
    #h1 { text="My Simple Application" },
    #label { text="What is your name?" },
    #textbox { },
    #button { text="Submit" }
].
```

Java dude:

LOL

at

you

Ruby on rails dude:

ROFL at y'all

Erlang dude:

```
calc_month(S) ->
    L = ["Jan", "Feb", "Mar", "Apr", "May",
         "Jun", "Jul", "Aug", "Sep", "Oct",
         "Nov", "Dec"],
    find_ix(L, S, 1).
```

Java dude:

LOL

at

you

Erlang dude:

Doing number crunching, you would completely utilize the available cores with few (as many) threads.

Erlang is for time sharing and doesn't like long blocking processes.

Java dude:

LOL

at

you

C dude:

ROFL
at
y'all

Thank you

Some code examples were
taken from public blogs / sites

Most images originate from
istockphoto.com

except few ones taken
from Wikipedia and product pages
or generated through public
online generators