BugSense

BigData for Mobile

@jonromero - BugSense.com


The BugSense Team

BugSense Trivia

- Second biggest SDK in the world (after Google)
- Analyzing data from more than 500M devices
- Custom BigData database
- Eleven engineers
- Cash positive

Splunk to Acquire BugSense - Analyst Blog

By Zacks.com, September 20, 2013, 06:36:21 PM EDT

AAA

Real time operational intelligence software maker **Splunk Inc.** (SPLK) recently entered into an agreement to acquire analytics solution provider BugSense Inc. The acquisition will enable Splunk to analyze machine data directly from devices and will help combine it with other information available from sensors and IT gear to improve operational intelligence.

BugSense was founded in 2011 and offers analytics to determine mobile app performance and collects information to troubleshoot problems. The solutions offered by BugSense support Android, iOS and Windows Phone.

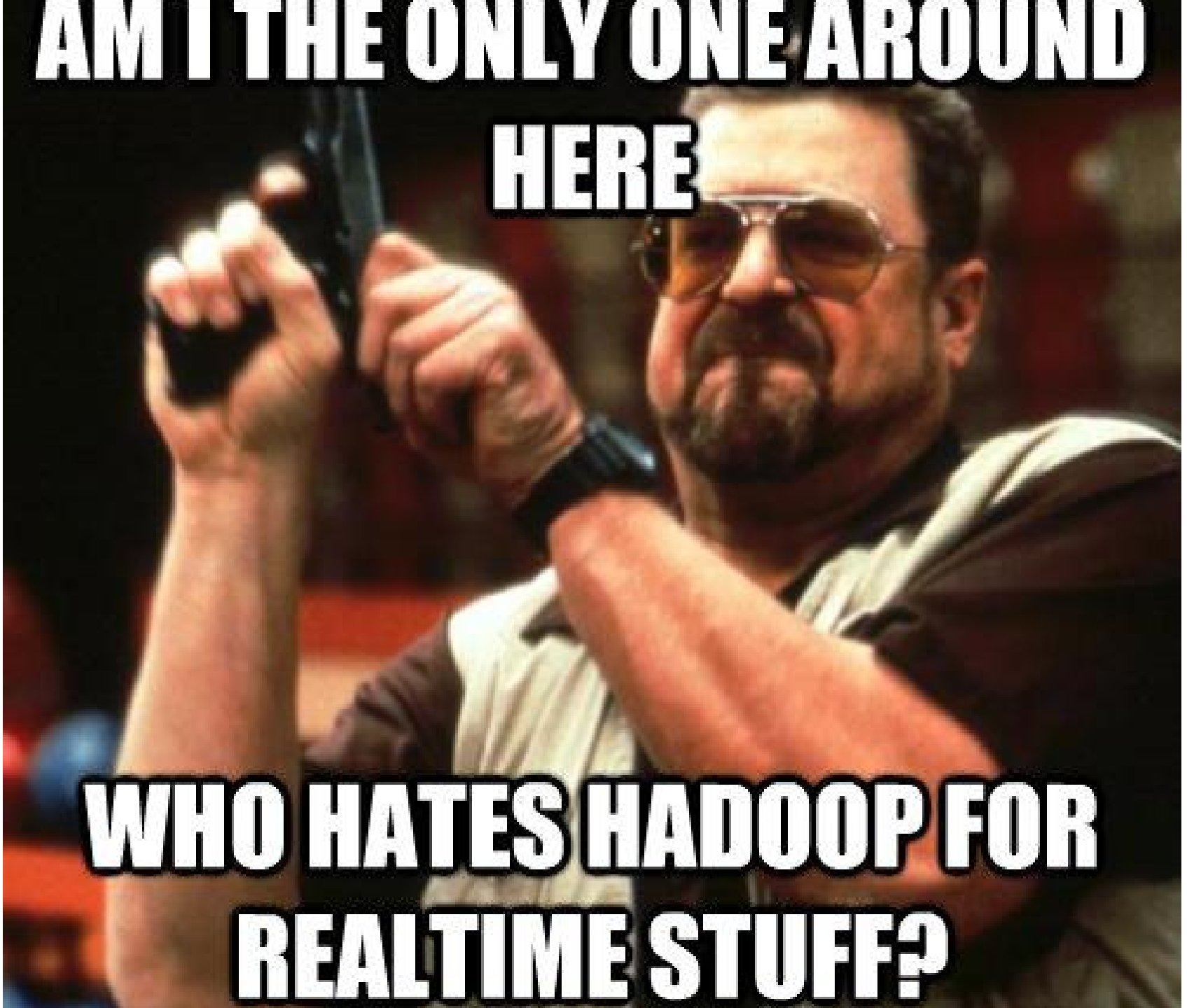
Splunk is one of the pioneers in the field of mining machine data and has served customers in the IT market for quite some time now. This acquisition will help the company to focus on mobile data analysis.

Due diligence

From Wikipedia, the free encyclopedia

For other uses, see Diligence (disambiguation).

"**Due diligence**" is a term used for a number of conc
involving either an investigation of a business or perso
prior to signing a contract, or an act with a certain
standard of care. It can be a legal obligation, but the
will more commonly apply to voluntary investigations
common example of due diligence in various industr
the process through which a potential acquirer evalu
its assets for acquisition.[1]


AM I THE ONLY ONE AROUND HERE
WHO HATES HADOOP FOR REALTIME STUFF?

LDB

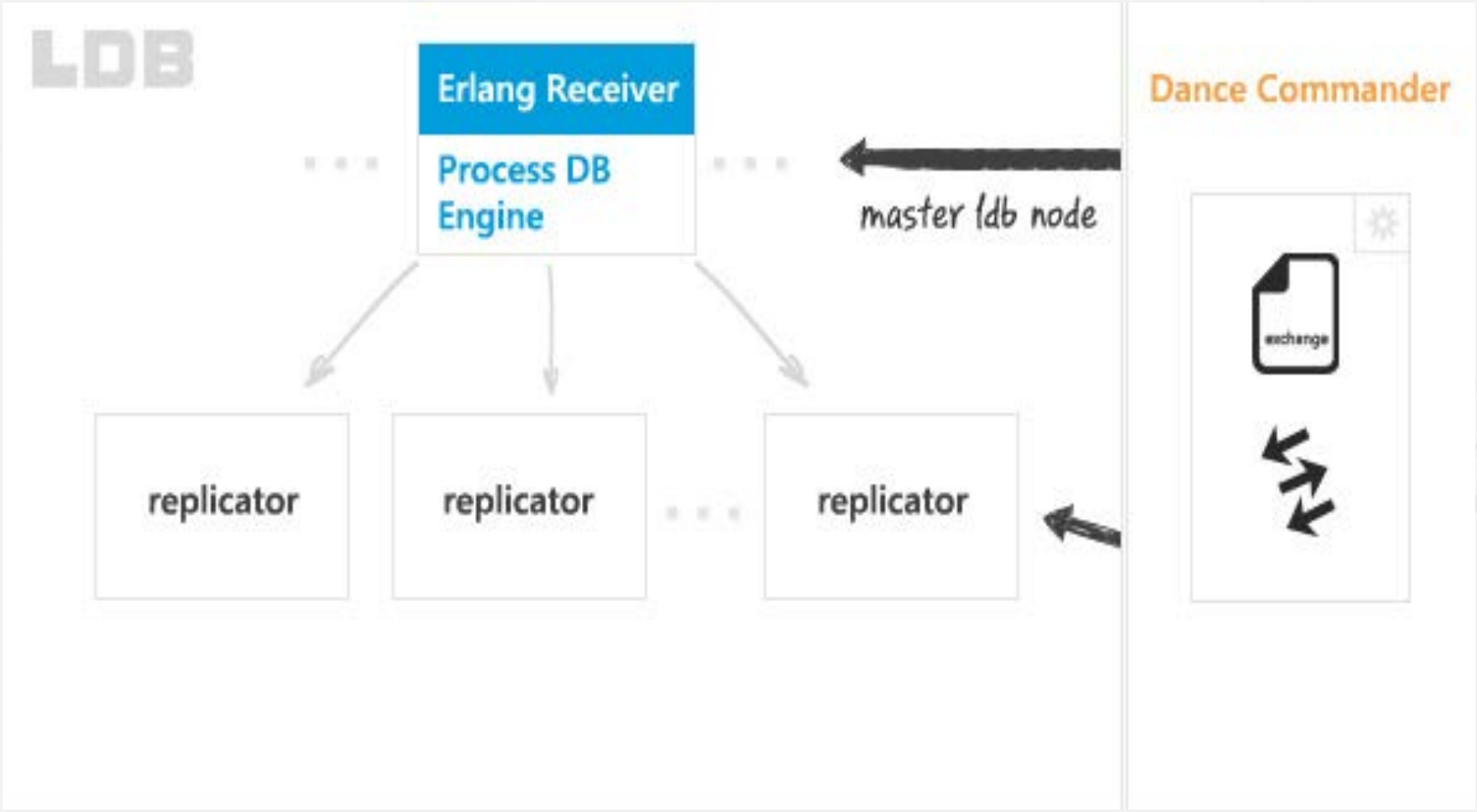
Cloud Memory

LDB Facts

- Complex event processing in-memory DB
- Super easy to setup/use - one package
- No table, row, column locks
- "Describe your data" mentality
- C runs circles around Java (speed)
- Predictable behaviour
- Lazy loading of files
- Saves data to disk
- "Let it crash"

Let it crash???

Architecture



Example

```
;; everything that happens here is for one day
(timespace "day")

;; build a json-like, timeseries response like that
;; "{'ts':[250,210,240,244,230,212,292]}"
;; Yes, with one line!!!
(timeseries (days-ago 7)
            (unique "users"))
```

Why Erlang

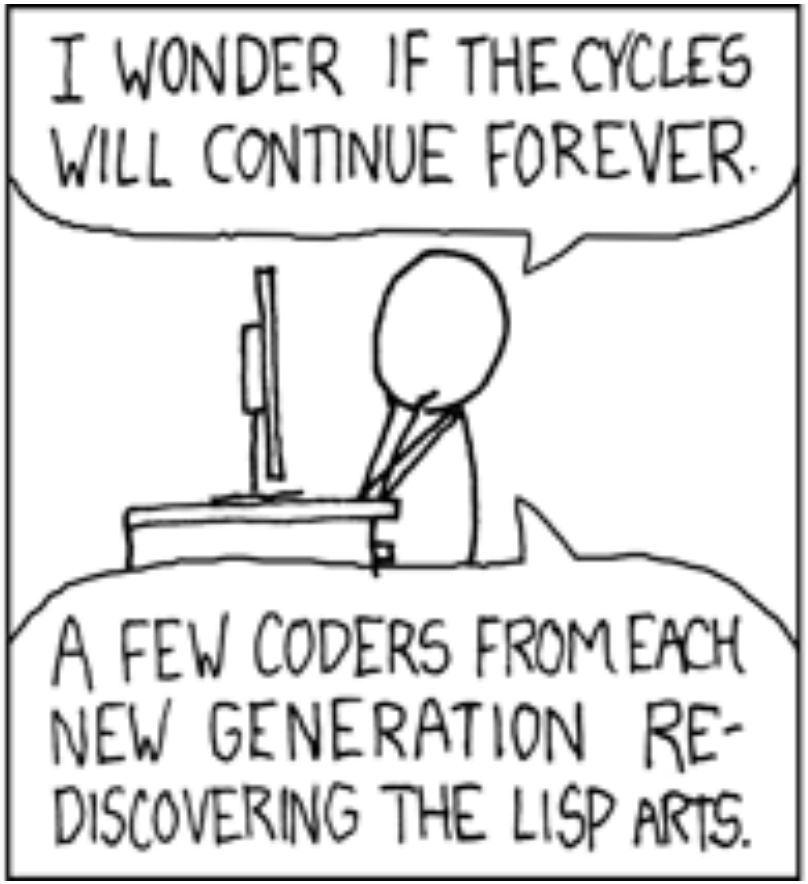
- Handles lots of connections efficiently (Cowboy)
- Sending/receiving messages from/to nodes is trivial
- Building a replication/take over engine is easy
- Mnesia to "share" config files
- C Linkedin Drivers
- Being able to connect to remote nodes
- All of the above - integrated in one language

Why C

- 19 machines -> more than 30k customers
- Correlations and processing
- In realtime
- Fine-grained locking
- Predictable behaviour (no GC)

Why LISP


ALF A
D IT
ERFECT,
OUT IT.

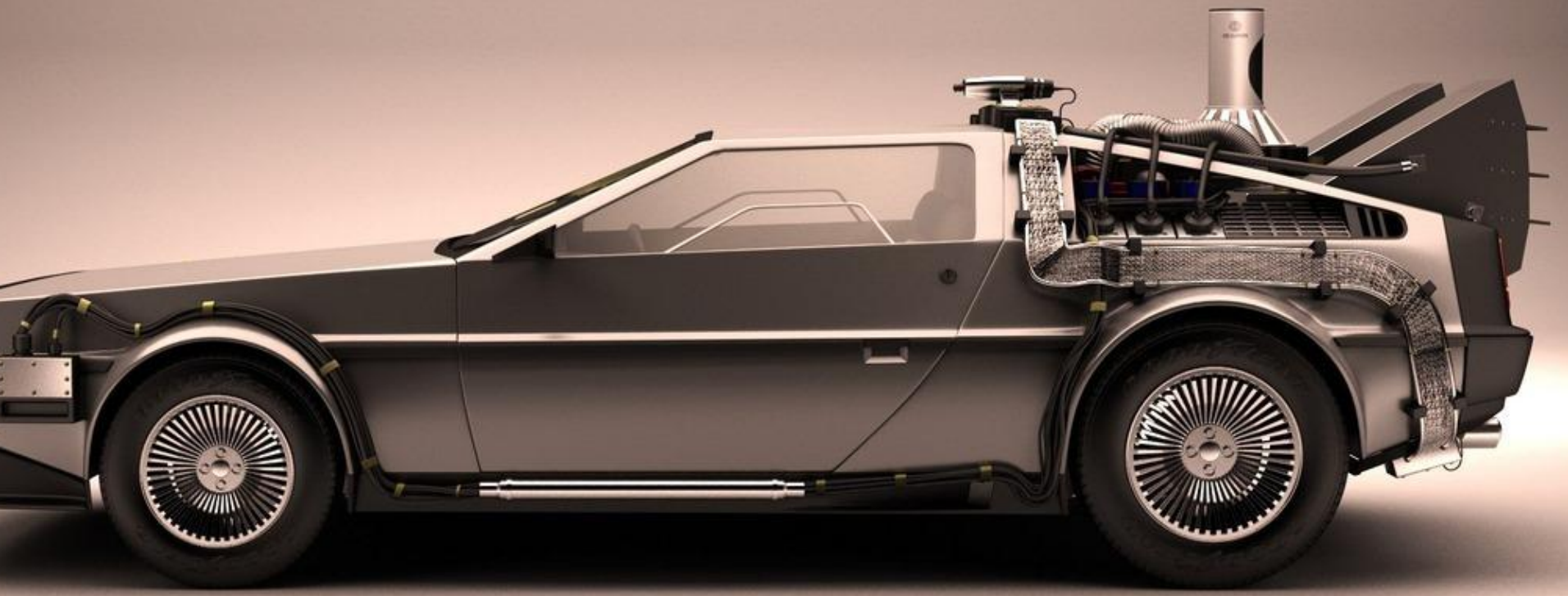

I WONDER IF THE CYCLES WILL CONTINUE FOREVER.
A FEW CODERS FROM EACH NEW GENERATION RE-DISCOVERING THE LISP ARTS.


THESE ARE YOU
FATHER'S PARENTHES
ELEGANT WEAPONS
FOR A MORE... CIVILIZE

Why LISP

- Super easy parser/lexer/interpret (two days to prototype)
- SQLish DSLs
- Expressive power
- Functional => Ideal for parallel computing
- Tons of Data => Data transformation FTW!

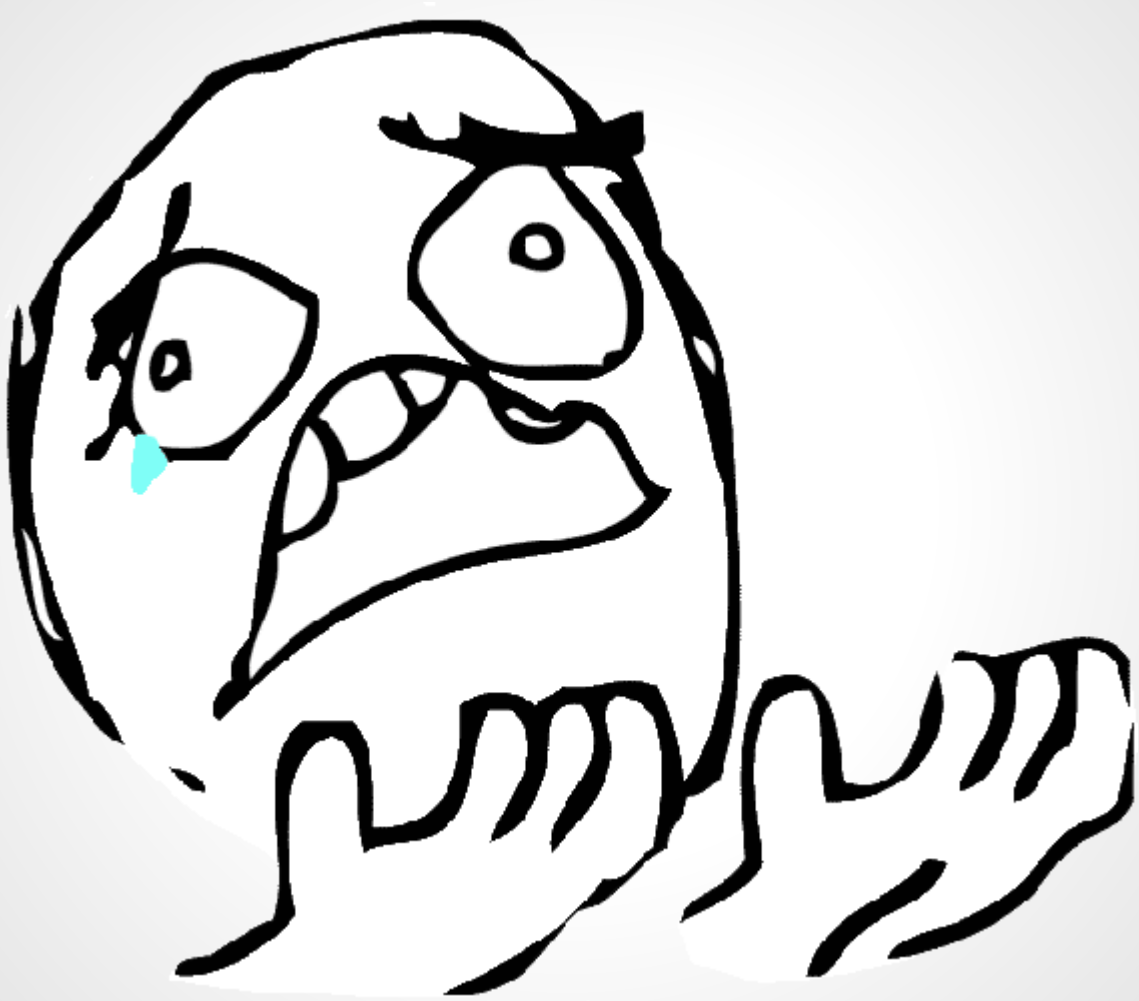
ROADS?



WHERE WE'RE GOING,
WE DON'T NEED ROADS.

LDB 3

- Erlang all the way (one code base)
- Scheduler is happy
- Faster development
- Not crashing the whole nodes
- Much faster!



Linkedin Drivers

- Crashing the c layer, crashes the VM (bad)
- Scheduler is not efficient anymore
- Very difficult to scale horizontally
- You need to be careful with (de)serialization
- Test, test, test

Linkedin Drivers

- When it is works, it is FAST
- C libraries (statistical, data structures etc)
- Mutable data structures
- Legacy code is trivial to integrate

LDB - OpenSource

- Still a possibility
- Opening up the old version
- Opening up a “community” version - LDB3

Thanks

Send me any questions

@jonromero