

**TO BE
ANNOUNCED**

**The Day after
PRODUCTION**

@lucamaraschi



I LOVE CORE DUMPS!



do epic shit!

"Production is WAR"

— *Bryan Cantrill*



Bryan Cantrill
@bcantrill



Following

OH: "Larry Ellison stepping down to become CTO of Oracle is like Vladimir Putin stepping down to become Prime Minister of Russia."

**3 stories
of real production war**

at scale!

ya...
we faced all 3!

***“Poor is the pupil who
does not surpass his
master.”***

– Leonardo da Vinci

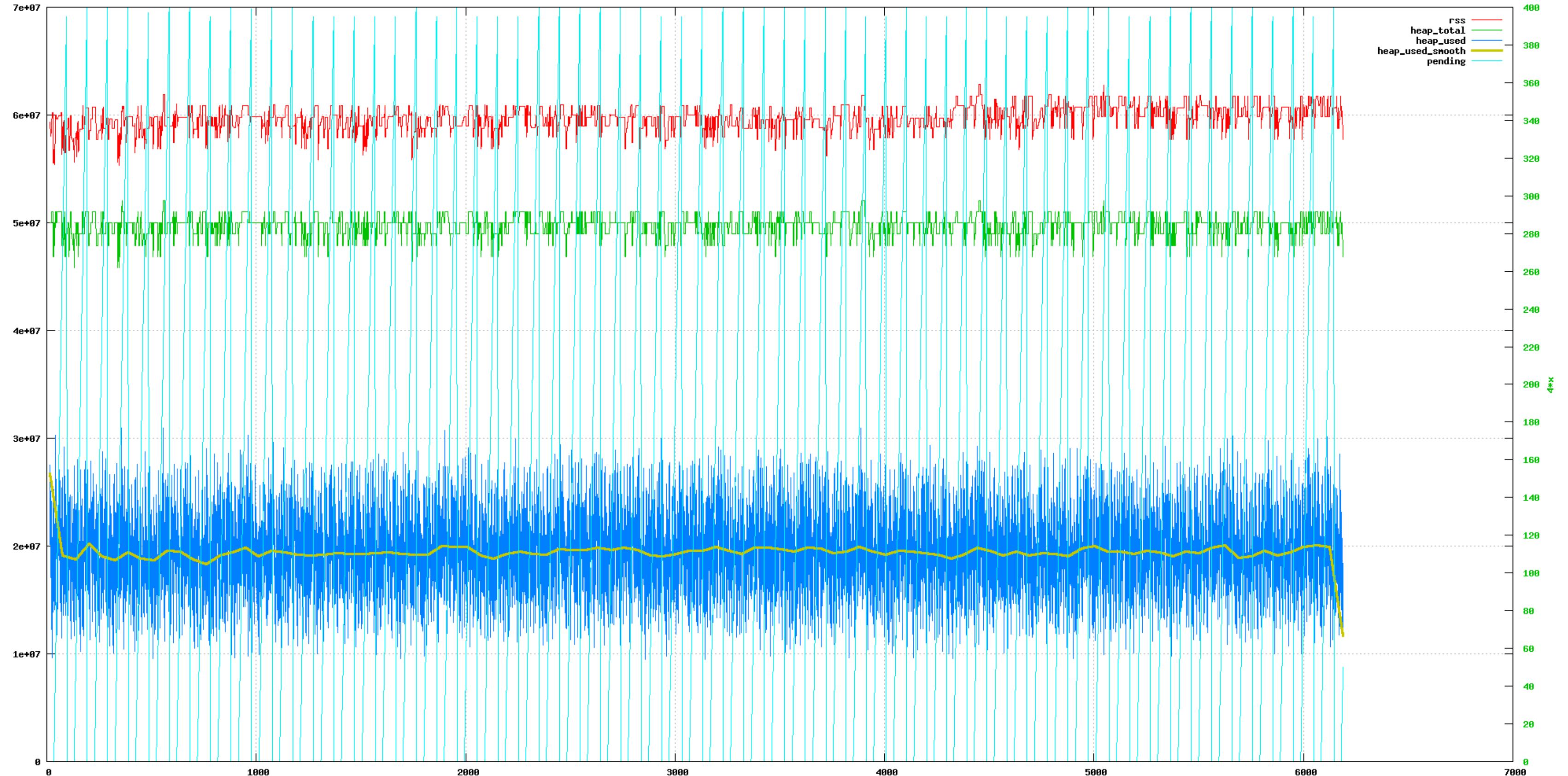
100

Walmart

The 'handleScope' Saga

handlescope

keeps track of the objects on stack



libumem.so.1`..		libumem.s..
libumem.so.1`..		libumem.s..
libumem.so.1`..		libumem.s..
libumem.so.1`.. libumem.so.1` umem_cache_alloc_de..		libumem.so.1` u.. libumem.s..
libstdc libumem.so.1` umem_cache_alloc		libumem.so.1` u.. libstdc
libstdc libumem.so.1` umem_alloc		libumem.so.1` u.. libstdc
_ZN2v88intern.. libumem.so.1` malloc		libumem.so.1` m.. _ZN2v88in..
_ZN2v88intern.. libstdc		libstdc _ZN2v88in..
_ZN2v88intern.. libstdc		libstdc _ZN2v88in..
_ZN2v88intern.. _ZN2v88internal11HandleScope6Ext..		_ZN2v88internal.. _ZN2v88in..
_ZN2v88intern.. _ZN2v88internal8JSObject20SetPro..		_ZN2v88internal.. _ZN2v88in..
<< internal c.. _ZN2v88internal10JSReceiver11Set..		_ZN2v88internal.. << intern..
<unknown> _ZN2v88internal7StoreIC5StoreENS..		_ZN2v88internal.. (anon) as..
<< adaptor >> _ZN2v88internal12StoreIC_MissENS..		_ZN2v88internal.. << adapt..
<unknown> << internal code >>		<< internal co.. (anon) as..
<< adaptor >> <unknown>		(anon) as Clie.. << adapt..
<unknown>		(anon) as finish at /hom.. <unknow..
<unknown>		(anon) as <unknown>
<< internal >>		<< internal >>
<unknown>		g at events.js pos 5670
<< adaptor >>		<< adaptor >>
<unknown>		(anon) as EventEmitter.emit at e..
<< adaptor >>		<< adaptor >>
<unknown>		parserOnIncomingClient at http.j..
<unknown>		parserOnHeadersComplete at http.js..
<< internal >>		<< internal >>
<< entry >>		<< entry >>
_ZN2v88internalL6InvokeEbNS0_6HandleINS0_10JSFunctio..		_ZN2v88internalL6InvokeEbNS0_6Handle..
_ZN2v88internal9Execution4CallENS0_6HandleINS0_6Obje..		_ZN2v88internal9Execution4CallENS0_6..
_ZN2v88Function4CallENS_6HandleINS_6ObjectEEEiPNS1_I..		_ZN2v88Function4CallENS_6HandleINS_6..
_ZN4node6Parser19on_headers_completeEP11http_parser		_ZN4node6Parser19on_headers_complete..
http_parser_execute		http_parser_execute
_ZN4node6Parser7ExecuteERKN2v89ArgumentsE		_ZN4node6Parser7ExecuteERKN2v89Argum..
<unknown>		<< internal code >>
<unknown>		socketOnData at http.js pos 45315
<unknown>		onread at net.js pos 12838

150MB leak per day!

down to 22...

**but still too much
for the volume of Walmart!**

3 months of debugging...

**make
or suffer!**

and Dtrace...
or die!

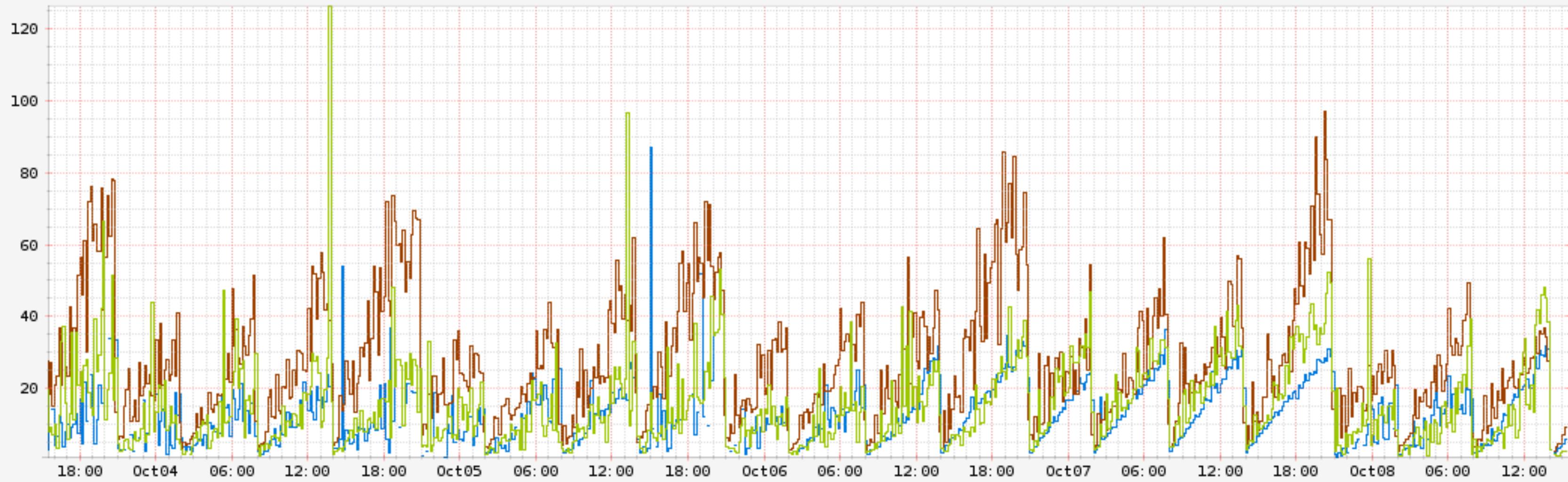
**Yep...and our workers
were leaking too! ;)**

Thank you Walmart!



2
netfix

Node.js in 'Flames'



v8 perf_events

Flame graphs

and

**Route handlers stored
in 1 global array**

**Express.js recursively iterates and invokes
all handlers
until it finds the right one!**

```
next();

function next(err) {
  if (err && err === 'route') {
    return done();
  }

  var layer = stack[idx++];
  if (!layer) {
    return done(err);
  }

  if (layer.method && layer.method !== method) {
    return next(err);
  }

  if (err) {
    layer.handle_error(err, req, res, next);
  } else {
    layer.handle_request(req, res, next);
  }
}
```



the answer
is...

Restify

**First class resident
traceability**

but how?

Bunyan

Dtrace probes or die!

Thank u Netflix!

and chill...



3.
joyent



PostgreSQL autovacuum

github.com/joyent/pglockanalyze

Thank you Joyent!

in progress...

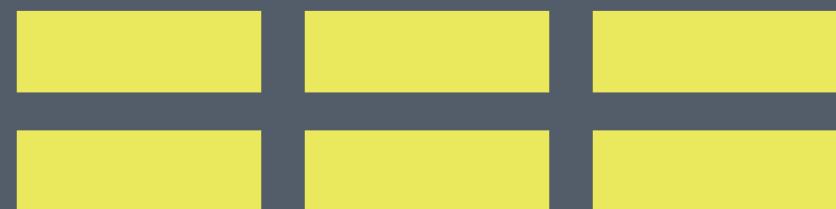
actually...
I just heard...

quote_literal()

vs.

quote_ident()

wrong quote



∞ LOOP

D'OH!



**Production
operability**



Traceability

Post-mortem

github.com/nodejs/tracing-wg
github.com/nodejs/post-mortem

Dapper, a Large-Scale Distributed Systems Tracing Infrastructure

Benjamin H. Sigelman, Luiz André Barroso, Mike Burrows, Pat Stephenson,
Manoj Plakal, Donald Beaver, Saul Jaspan, Chandan Shanbhag

Abstract

Modern Internet services are often implemented as complex, large-scale distributed systems. These applications are constructed from collections of software modules that may be developed by different teams, perhaps in different programming languages, and could span many

because large collections of small servers are a particularly cost-efficient platform for Internet services workloads [4]. Understanding system behavior in this context requires observing related activities across many different programs and machines.

A web-search example will illustrate some of the challenges such a system needs to address. A front-end ser-



Zipkin is a distributed tracing system that helps us gather timing data for all the disparate services at Twitter. It manages both the collection and lookup of this data through a Collector and a Query service. We closely modelled Zipkin after the [Google Dapper](#) paper. Follow [@ZipkinProject](#) for updates.

Collecting traces helps developers gain deeper knowledge about how certain requests perform in a distributed system. Let's say we're having problems with user requests timing out. We can look up traced requests that timed out and display it in the web UI. We'll be able to quickly find the service responsible for adding the unexpected response time. If the service has been annotated adequately we can also find out where in that service the issue is happening.

--abort-on-uncaught-exception

unhandledRejection does not have the correct stack trace.

#5084

ⓘ Open

Raynos opened this issue 6 days ago · 60 comments



Raynos commented 6 days ago

Given the recent PR ([#5020](#)) I wanted to see if promises could enable core dump based debugging.

It looks like currently when an `UnhandledRejection` occurs the stack is gone. This is very similar to how the stack is gone when an `UncaughtException` occurs.

We've added `--abort-on-uncaught-exception` flag to allow us to abort the process with the correct stack in C++. We need to implement `--abort-on-unhandled-rejection` flag as well.

what's next?

**post-mortem analysis
cross platform!**

made like
everywhere!

Linux support for mdb #17

ⓘ Open

Raynos opened this issue 5 days ago · 7 comments



Raynos commented 5 days ago



There was a conversation about post mortem on nodejs/node where an engineer did not know how mdb worked because he's never run a smartos vm. This started me thinking about Linux support.

one of the uber engineers ported a subset of mdb to Linux.

Document the v8 Heap Snapshot Format #14

! Open

yunong opened this issue 14 days ago · 0 comments



yunong commented 14 days ago

Node.js Foundation member



Documentation already exists today: <https://bmeck.github.io/snapshot-utils/doc/manual/overview.html> but lacks semantic information. Produce more complete documentation, including semantics on the v8 heap snapshot format.

Proposal for Common Heap Dump Format #13

! Open

yunong opened this issue 14 days ago · 8 comments



yunong commented 14 days ago

Node.js Foundation member



From the meeting on 1/28/16, the group aligned on producing a standardized common heap dump format. The current V8 heap dump format is a good starting point, but it has a few drawbacks.

@davepacheco has volunteered to take a stab at a proposed common file format.

github.com/indutny/llnode
@indutny

Community driven
enterprise effort

***“One reason I think debugging is so hard:
you need to know much more about how
computers work to debug than you do to
program.”***

Dave Pacheco(Twitter - 18 August 2015)

Thank you all!

do u like Epic Shit?

JOIN #IceMobile!