

# Influences of Node

Benevolent Dictator Emeritus

July 2, 2012

Node was a long progression of attempts to build websites. It was influenced by a lot of software:

CouchDB, Ebb, Flow, FUSE, libebb, libeio, libev, libircclient, Merb, Mongrel, NGINX, nginx-ey-balancer, Ragel, Ruby on Rails, SqueezeBox, timber lang, XUpload

And probably others. This is a story of what I was doing before Node was first released in June 2009.

2004-11-10. Rochester, New York.



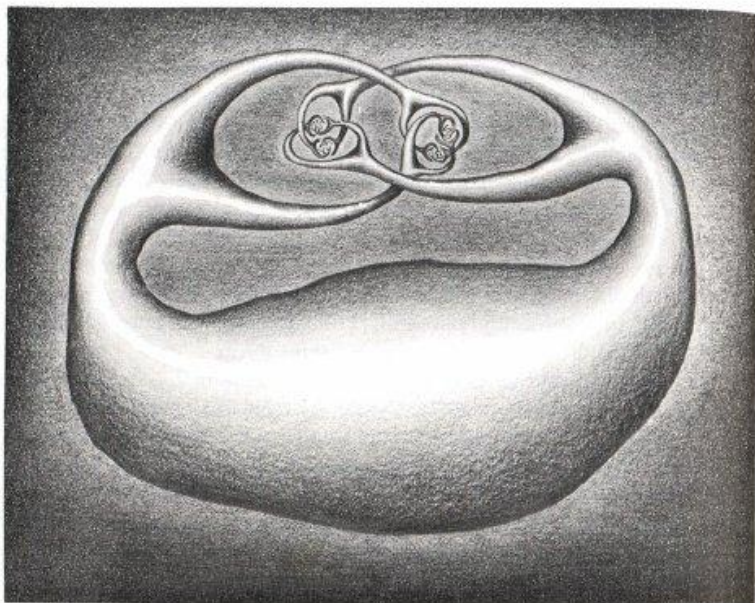
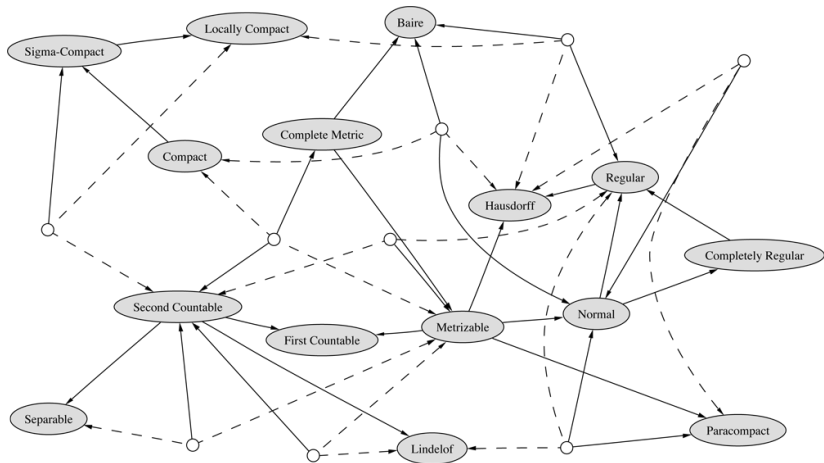


FIG. 4-11. The Alexander horned sphere.



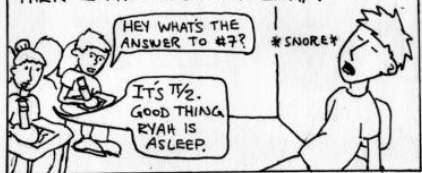
by ryan dahl < rld@math.rochester.edu >. generated on wed may 26 19:52:46 2004 (gmt)

<http://tinyclouds.org/topology.pdf>

I DID A BUNCH OF GRADING.



THEN I PROCTORED AN EXAM

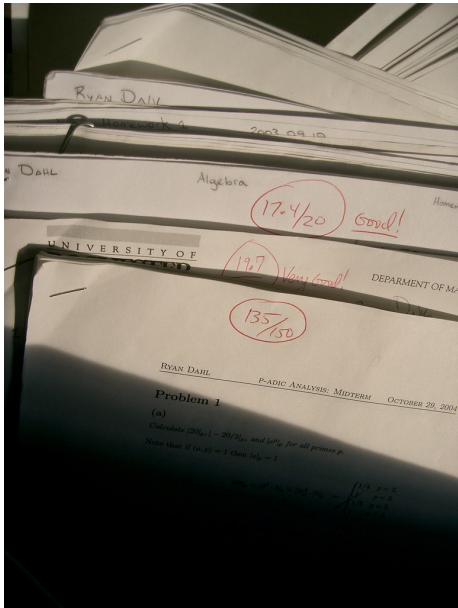


AND I GRADED LATE INTO THE NIGHT



I'VE DEFINITELY ENTERED THE "POST-QUALS SLUMP"



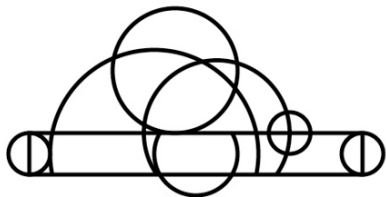


2006-08-20. Valparaíso, Chile.





2006-08-10. Valparaiso, Chile. Each night I've been working until the cold winter morning in order to discover a method of communicating via clouds. As of yet, I am unsuccessful.



2006-08-15. Valparaiso, Chile. One project is almost done and I can put it online soon. It's a theorem database and automated proof thingy for topological spaces. I made it so that I could learn Ruby on Rails - which is hot.

	3	4	5	6	7	8
To	1	1	1	1	1	1
T <sub>1</sub>	0	1	1	1	1	1
Hausdorff	0	0	1	1	1	1
T <sub>3</sub>	0	0	0	0	0	1

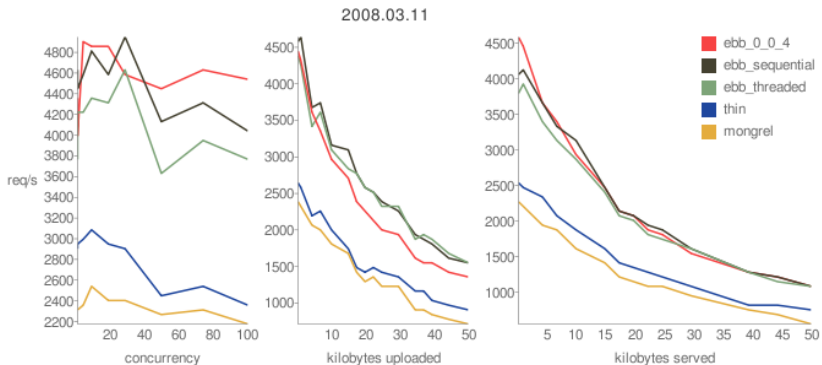
2006-12-09. Buenos Aires. I just got a contract to do a big website. \$ + work = travels to tierra del fuego in January.

2006-12-18. Buenos Aires. HTTP upload progress bars are rather obfuscated- they typically involve a process running on the server keeping track of the size of the tempfile that the HTTP server is writing to, then on the client side an AJAX call is made every couple seconds to the server during the upload to ask for the progress of the upload. This is pretty ridiculous.

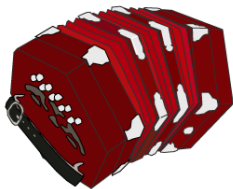
<http://four.livejournal.com/730831.html>



## Ebb web server. Like Thin but using libev instead of EventMachine



<http://ebb.rubyforge.org/>



# SqueezeBox

## Yet Another Web Framework

SqueezeBox extends the [Mongrel web server](#) to provide an effortless web framework. Designed after the Model-View-Controller pattern, SqueezeBox separates presentation and back-end cleanly but slightly differently than contemporary systems. The major difference between Rails, Merb, or the other Ruby frameworks is that SqueezeBox routes HTTP requests based on file system organization. (Like `mod_php`!) While some may scoff at this simplistic design, filesystem routing significantly lessens the complexity of usage.

<http://squeezebox.rubyforge.org/>

Precursor to libuv—the just the C parts of Ebb. Unix only.

```
GET /favicon.  
n.ico HTTP/  
1.1 Connec  
tion: keep-  
alive
```

## libebb

libebb is a lightweight HTTP server library for C. It lays the foundation for writing a web server by providing the socket juggling and request parsing. By implementing the HTTP/1.1 grammar provided in RFC2612, libebb understands most most valid HTTP/1.1 connections (persistent, pipelined, and chunked requests included) and rejects invalid or malicious requests. libebb supports SSL over HTTP.

The library embraces a minimalistic single-threaded evented design. No control is removed from the user. For example, all allocations are done through callbacks so that the user might implement in optimal ways for their specific application. By design libebb is not thread-safe and all provided callbacks must not block. libebb uses the [high-performance](#) libev event loop, but does not control it. The user of the library may start and stop the loop at will, they may attach thier own watchers.

<http://tinyclouds.org/libebb>

2007-12-06. Cologne. I'm working on a new HTML templating language using javascript. The language uses indention white space to build a XML hierarchy, the Haml syntax. In and around the XML one may use javascript for loops, conditionals, and subrutines. The attributes of a XML node are javascript objects. [...] I've written a proof of concept in Ruby using Ragel and SpiderMonkey. [...] It will allow front-end developers to more work independently of server-side programmers.

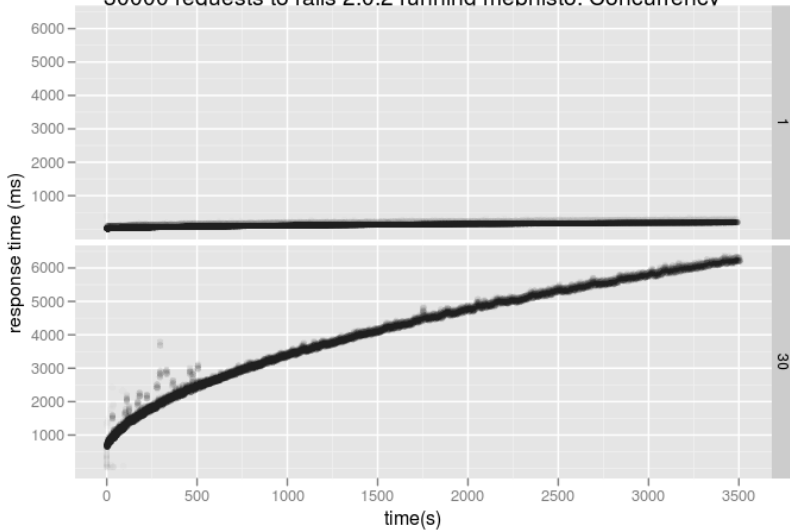
<http://four.livejournal.com/820601.html>



EngineYard contracted me to make NGINX load balance in a similiar way to haproxy. Before I began I demonstrated the need for this:

2009-01-26. Cologne. The result is what every system admin knows intuitively: Rails gets worse over time if loaded down with connections. The culprit is most likely MRI's crappy thread implementation. While Rails serves 1 request, 29 threads are there sitting idle sucking up resources. <http://four.livejournal.com/955817.html>

### 30000 requests to rails 2.0.2 running mephisto. Concurrency



2009-01-30. Cologne. i'm making a file system with only one thread!

2009-02-19. Cologne. i'm listening an album which is stored on the file system i wrote.

EngineYard had contracted me to build a FUSE file system on top of a nascent (and now dead) DynamoDB clone. It was killed after only a few weeks.

However the origin of the JavaScript “comma-first style” began with how I styled the C in this project. <http://four.livejournal.com/964924.html>

```

( char *key
)
{
    block_t *block;
    block = tst_search(key, block_tree, 0, NULL);
    if(block) return block;

    block = malloc(sizeof(block_t));
    memset(block, 0, sizeof(block_t));
    block->key = strdup(key);
    block->load = NULL;
    block->size = 0;
    block->base = 0;
    block->allocated = 0;
    tst_insert(key, block, block_tree, 0, NULL);

    return block;
}

block_t * block_find_from_file
( file_t *file
, int index
)
{
    // The block key looks like this
    // ^7717aadedb149b92a5d1f91779c3df7a:45^
    char key[strlen(file->key)+1+20];
    sprintf(key, "%s:%d", file->key, index);

    block_t *block = block_find(key);
    if(!block) return NULL;

    assert(block->file == file || block->file == NULL);
    block->file = file;
    block->index = index;

    return block;
}
/*

```

block.c

61,3

12%

```

#include "file_mem.h"
#include "db.h"

#define BLOCK_KEY_SIZE 40

typedef struct block_t block_t;
typedef struct block_load_t block_lo

typedef void (*block_done_cb)
( block_t *block
, int error
, void *userdata
);

struct block_t {
    char *key;
    block_load_t *load;

    file_t *file;
    int index;

    char *base;
    size_t allocated;
    size_t size;
};

void block_init
( struct db_ops
);

block_t* block_new
( void
);

block_t* block_find
( char *key
);

block_t * block_find_from_file

```

block.h

40,1

12%

2009-02-09. Cologne. I'm going to

- ▶ write a special thin web server tied to the V8 javascript interpreter (Mongrel for V8 if you will)
- ▶ write a special evented TCP library for V8 (probably by wrapping liboi). Does this exist in public already? Certainly google has such a thing
- ▶ bundle them with a HTTP API for modifying server-side documents

[...] I think this design will be extremely efficient and support very high loads. <http://four.livejournal.com/963421.html>

Ryan (painfully hungover) and Tim at Curucamp 2009 in Vienna.



Date: Sun, 1 Mar 2009 02:46:33 +0100

Subject: solution

From: Ryan Dahl

To: Tim Becker

okay. going to do it in javascript. main thing is tcp. all will follow from there. maybe we'll just use databases instead of file systems. (we can cache image data inside javascript if need be)

So. TCP API:

```
TCP.connect(host, port, onConnect,  
            onRead, onDrain, onClose);
```

Example:

```
function myOnConnect () {  
  this.write('GET /index.html HTTP/1.1');  
  this.write('\r\n\r\n');  
}
```

```
TCP.connect("google.com", 80,  
            myOnConnect);
```

I will implement DOM's timer thing `setTimeout()` so that it uses libev timers.

thoughts? i think this is extremely simple and will solve the world's problems.



Date: Sun, 1 Mar 2009 09:40:29 +0100  
Subject: Re: solution  
To: Ryan Dahl  
From: Tim Becker

It is extremely simple, feels like it's missing something, but I can't think of what at the moment apart from `close`.

Anyway need to pack, see you! -tim

Date: Sun, 1 Mar 2009 14:49:06 +0100

Subject: Re: solution

From: Ryan Dahl

To: Tim Becker

So. The idea is shifting in my head. Instead of presenting this primarily as a web server with some extension libraries for TCP lets present it as a set of evented libraries.

Initially we will provide

```
TCP.connect() // described above
```

```
HTTP.serve('localhost', 80, onRequest)
```

But later we'll provide

TCP.serve, HTTP.connect, MySql.connect ?

This will be the framework that we eventually build our HTTP-only web server (where you upload javascript to change the behavior through HTTP). But we'll release this set of libraries initially. Maybe call it netv8?

```
./netv8 myscript.js
```

Would be nice to have some way of loading external js files (perhaps through TCP?)

Date: Sun, 1 Mar 2009 14:58:05 +0100

Subject: Re: solution

From: Ryan Dahl

To: Tim Becker

> Maybe call it netv8?

> ./netv8 myscript.js

better name: "node" ?

./node irc\_client.js

WHEREAS, The usage of threads has complicated computer programming; and

WHEREAS, V8 javascript comes free of I/O and threads; and

WHEREAS, Most operating systems do not provide asynchronous file system access.

Now, therefore:

This set server and client libraries were made to build simple but fast servers. They are provided free of charge under a permissive simple license.

Submitted by Ryah Dahl, Tim Becker, March 1, 2009



Menu http://localhost:7000/ keypress event tab javascript

The New York Times - Breaki... irc Google Mail - Inbox - coldred...

```
02:40 nopcode_ if you use one width on one platform and another one on another platform
02:40 dizzy you can reach the fastest speed possible in a portable way
02:40 dizzy when you have access to the fastest integer type fo rthat platform
02:40 nopcode_ unless you ever need to port data ;)
02:40 kadoban you can port data in a portable way
02:40 dizzy of course
02:41 dizzy networking and binary formats are a common occurance in my programs
02:41 FauxFaux Unlucky.
02:41 dizzy but that's not relevant to the discussion at hand
02:41 nopcode_ ACTION thinks System.Xml.Serialization
02:41 nopcode_ *sigh*
02:41 dizzy you clearly have all your needs satisfied by C# in which case you shoul djust use C#
02:42 dizzy my needs are not satisfied by C# so I need to use something else
02:42 FauxFaux No! Must convert the non-beleEVERS!
02:42 nopcode_ nah i do scientific visualization and number crunching, i need c++
02:42 dizzy having something as flexible as C++ that allows to write both high level code and low level ones
    makes C++ one of the best solutions for my needs
02:42 nopcode_ i just wish i didnt ;P
02:42 FauxFaux If only boost::serialisation::xml existed. Oh wait.
02:43 dizzy IMO the C low level features of C++ shouldn't even be taught in C++ books
02:43 dizzy it shopol dsome kind of "secret knowledge"
02:43 dizzy only for those people that read the standard and find them
02:43 FauxFaux Ac++ tries.
02:43 dizzy so they say :)
02:43 nopcode_ like unsafe{} in c#? ;)
02:44 ville FauxFaux: The XML format is slightly bothersome to serialize say containers: <element>1</element>
    <element>2</element>... Maybe see where I am going with this
02:44 Math` dizzy: there's one thing that looks like that in C99
02:44 FauxFaux Oh, I wasn't advocating XML. No siree.
02:44 #hello #javascript #c++ #ruby-lang #c #haskell blah
```

## Initial Announcemnet:

2009-05-31. Cologne. Node is a new server-side javascript project. It provides a purely event-based interface to I/O:

- ▶ TCP server and client
- ▶ Standard setTimeout() setInterval() timers
- ▶ Asynchronous file I/O
- ▶ HTTP server and client

Node's main focus is on performance and efficiency.

<http://four.livejournal.com/1003191.html>