9 Challenges in Node.js

ryan@joyent.com

September 26, 2010

Node is gaining popularity.

To encourage contributors I want to highlight what I see as the biggest problems facing Node

Problem 1: TLS / SSL

TLS is increasingly important. “Infrastructural” protocol. Needs very good support.

Current bindings to OpenSSL are

- Incomplete (E.G. No sessions)
- Buggy
- Tied into socket code – should be independent.

Ultimately unusable.

Problem 1: TLS / SSL

Would like to have with *really* good support for both the client and server for:

- Server Name Indication
- Sessions
- OCSP stapling
- False Start
- Snap start (!)

`http://www.imperialviolet.org/2010/06/25/overclocking-ssl.html`

Problem 2: Continuous Integration Perf Testing

Development is mostly blind with respect to performance.

HTTP hello world server is tested regularly and sometimes start up time.

Mostly performance is unknown.

Problem 3: Debugging

Not all is sunshine in the world of callbacks:

```
1 function f () {
2   throw new Error('foo');
3 }
4
5 setTimeout(f, 10000*Math.random());
6 setTimeout(f, 10000*Math.random());
```

From which line does the error arise?

Problem 3: Debugging

```
% node x.js

x.js:2
  throw new Error('foo');
        ^
Error: foo
    at Timer.f [as callback] (x.js:2:9)
    at node.js:266:9
```

No reference to either of the **`setTimeout`** lines in the stack trace.

Hard to debug when you're always destroying your stack. Often difficult to discover which path lead you to an exception.

Problem 3: Debugging

Would like to dynamically turn on the V8 debugger (via a signal?).

- Must have zero performance impact when off
- Useful for LiveEdit features

Similarly: I would like to be able to trigger heap profiles at anytime.

Problem 3: Debugging

DTrace probes all over Node:

- Function calls in V8 (getting args impossibly hard?)
- Settle for an is-enabled probe?
- EventEmitter firing (at least off the event loop)
- Thread pool queue size

Problem 4: Windows

Currently there is a CYGWIN port, but that's not very satisfying.

Would like to compile on MinGW or MSVC but much work needs to be done.

Problem 4: Windows

A lot of **`libev`** will not work on Windows. (E.G. **`ev_async`**–AKA the pipe trick–probably needs to be rewritten.)

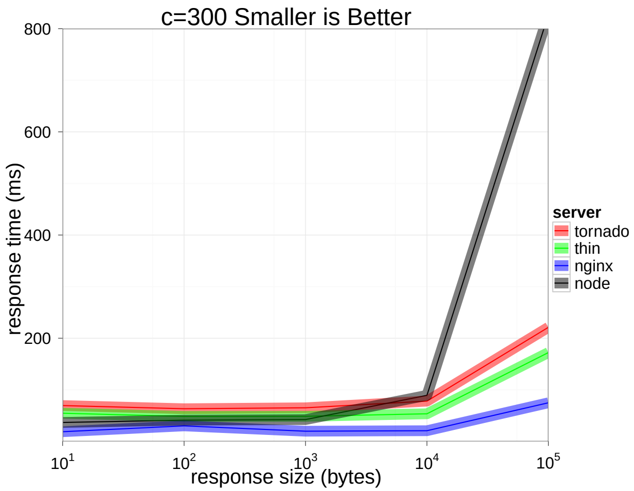
Desirable to use IOCP instead of **`select()`**... (AFAICT Impossible to do on libev...)

Perhaps porting **`libev`** is not impossible–but it will use **`select()`**.

Problem 5: V8 on x64 limited to 1 GB heap

Somewhat unreasonable to expect all server-side processes to run under 1 gig.

Problem 6: Copying Strings out of V8 heap



(Warning: dated benchmark - but behavior should still hold)

Problem 6: Copying Strings out of V8 heap

Writing strings to a socket or file currently requires two copies:

1. copying the data out of the V8 heap into an externally allocated buffer.
2. moving the data to the device on the **`write`** syscall

This needs to be faster for at least a few common cases - writing out a JSON string for example.

Problem 7: Bunching Writes

In many cases, multiple **`write()`** calls are being made in a single iteration of the event loop (for a single file descriptor.)

Leads to smaller packets.

Outgoing data should be queued during the iteration until just before Node goes to sleep (**`kevent`**, **`epoll`**, etc) and sent out at once.

Problem 8: File System Event Notification

`fs.watchFile` uses inotify on Linux but polls with **`stat()`** on all other operating systems.

This could be much better. Almost every OS has some FS event notification mechanism–Node should take advantage of it.

Problem 9: Kernel AIO

`libeio` uses pthreads to execute blocking system calls. Where possible, it would be useful to write a new **`libeio`** backend for kernel AIO (e.g. on Solaris).

Contributions Welcome

New low-traffic mailing list for Node development:

`http://groups.google.com/group/nodejs-dev`

Joyent `no.de` Service

Second preview version!

1. If you don't already have a Joyent account sign up at `http://no.de/`
2. Get a coupon code:
```
curl -k -u user:pass https://api.no.de/heart -X POST
```
3. Provision a machine:
```
curl -k -u user:pass
https://api.no.de/smartmachines/node -F
"coupon=123456789abcdefghijk" -F "subdomain=myname"
```

The first 150 people can generate a coupon code for creating a `no.de` Smart Machine.

http://nodejs.org/

ryan@joyent.com