react server side rendering

react boston february 2015
@nickdreckshage

isomorphic is a pretty sexy word…

- …for shared code. spectrum: templates -> meteor.
- got into it with rendr / spikes talk (http://bit.ly/1MoAzjZ)
- experimented making 'isomorphic' on npm
- frontend trend / direction (backbone, react, ember)
- word not cool anymore? oh well (http://bit.ly/1CxzIFy)

react made it extremely easy

http://bit.ly/1yzOaul

workflow

- node as a ui layer (http://bit.ly/1bjOXb3)
- not too many abstractions (webpack, superagent, etc)
- careful with context (http://bit.ly/1EewJEn)
- react router; react nexus
- relay / graphql

smart server side rendering

- 1 lookup; 1 event binding; 0 repaints
- base10 numbering; adler32 checksum
- high level (virtual dom; synthetic events)
- low level (jquery killer bees)
- larger html payload. html + json data + async js

sounds perfect...

server side performance

- not great!
- simple experiment (http://bitly.com/1yzOaum); 1mb json; 1000 items; a few components / partials.
- ab -c 1 -n 100; mean response time on slow macbook air

react (node)	react* (node)	hogan (node)	mustache (go)
1250ms	650ms	450ms	30ms

*using best practices

q&a @ react conf

- http://bit.ly/1DxGdfx
- “funny story about server rendering - it wasn’t actually designed that way” (Sebastian Markbåge)
- “we don’t use it that heavily, which is why we haven’t really invested strong in it” (Sebastian Markbåge)
- improvements: autobinding, ignoring state

a secondary concern

- primary concerns: architecture + performance
- server side rendering a bonus (with great architecture)
- facebook doesn’t use server side rendering in prod (tmk)
- instagram did, then abandoned it (http://bit.ly/1FPekyY)
- transparency > collective ignorance

growing importance

- netflix switching all platforms to react
- facebook mobile (server rendered; relay / graphql)
- e-commerce? non-single page apps?
- bbc mobile (http://bit.ly/1Ab0uoD)
- flipboard (http://bit.ly/1zTrFnK)
- ssr performance as final barrier. then no downside?

best practices

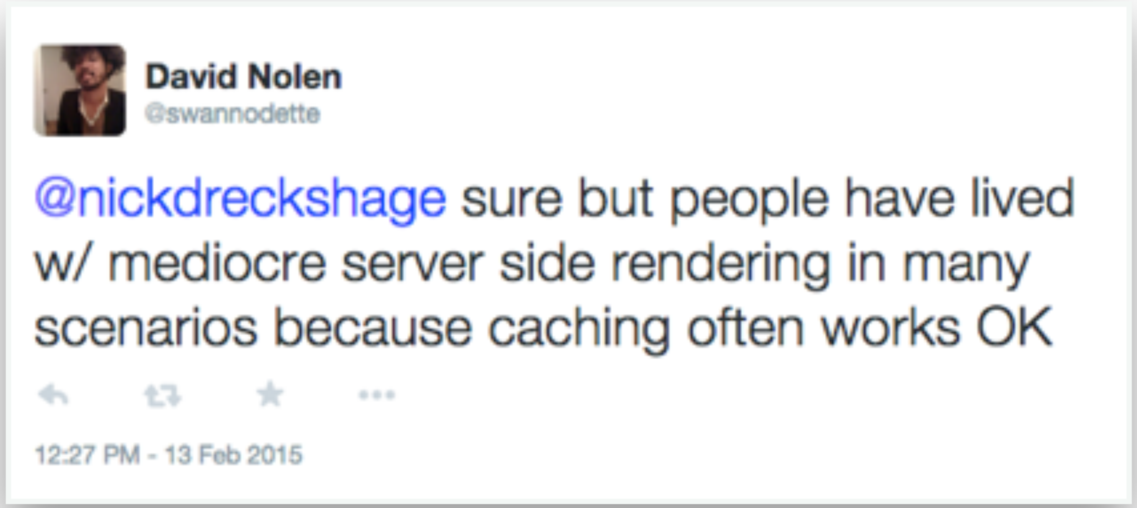
- use node. not ruby racer / php v8js
- compile jsx / webpack for server?*
- NODE_ENV=production*
- require('wrapped-react')*
- cluster.fork()
- cache (um...)

*http://bit.ly/1AF4yiP

easy	1250ms
+ compile jsx	1240ms*
+ NODE_ENV	990ms
+ wrapped	650ms
+ cluster	**
+ cache	30ms!

* increases with # of components

** basic load balancing


David Nolen
@swannodette

@nickdreckshage sure but people have lived w/ mediocre server side rendering in many scenarios because caching often works OK

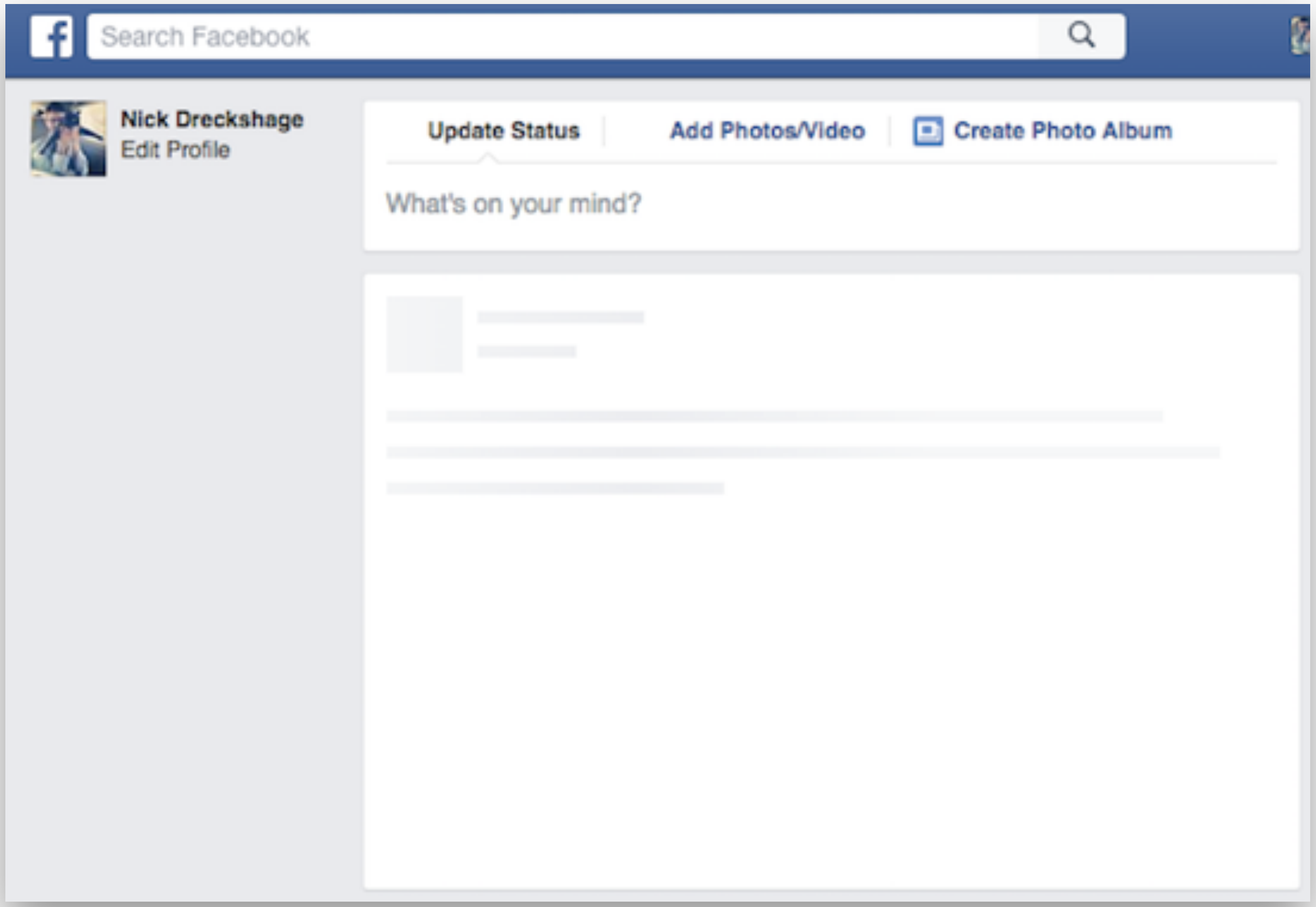
12:27 PM - 13 Feb 2015

crazy ideas

- node limitations, value in portability
- jsxtache (dont use) (http://bit.ly/1Modxtm)
- compile to js? clojure? hhvm/hack?

perceived performance

- preview templates
- optimistic updates
- first paint?



lets stop using the phrase perceived performance
when talking about server side rendering

...so should i still use it? yes.
react is awesome.