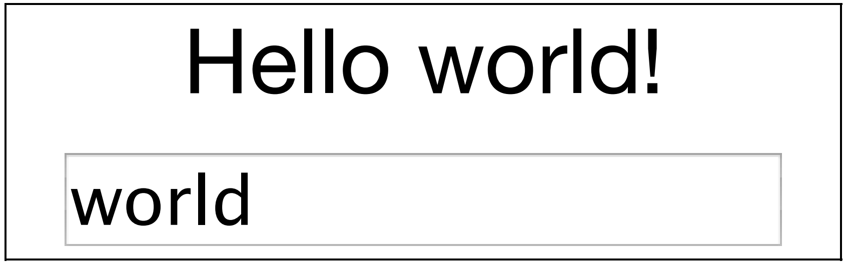
Supercharge AngularJS

Performance Measurement and Tuning

Sebastian Fröstl && Damien Klinnert

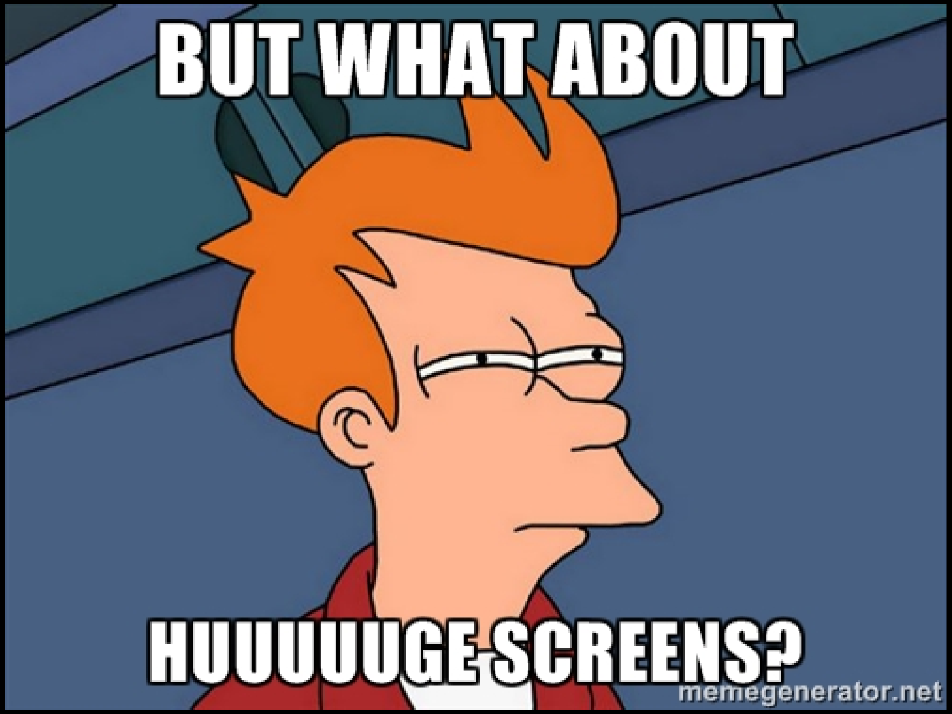
We <3
angular.js

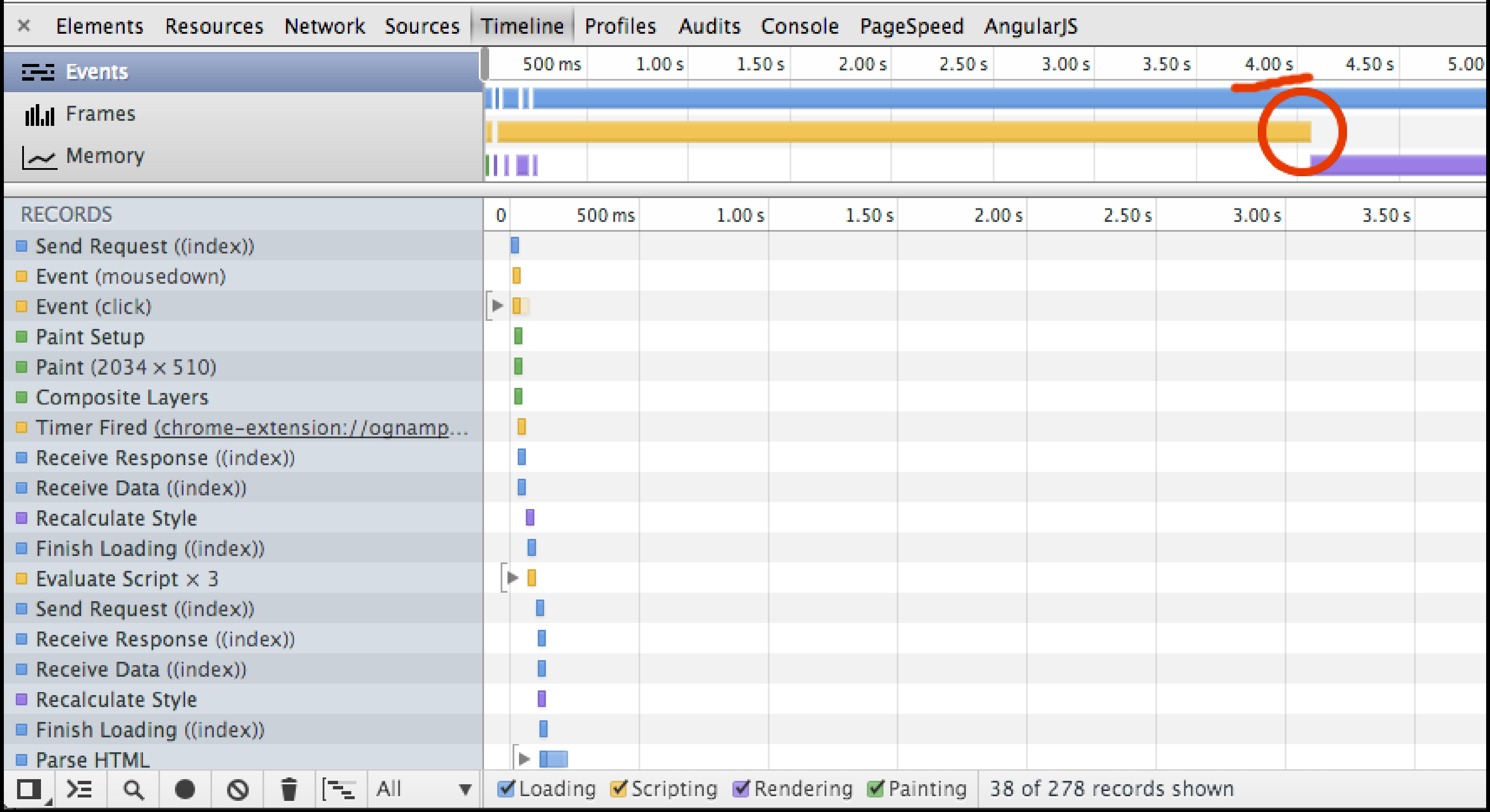


Hello world!

world

```
<h3>Hello {{name}}</h3>
<input ng-model="name">
```


BUT WHAT ABOUT
HUUUUUGE SCREENS?
memegenerator.net

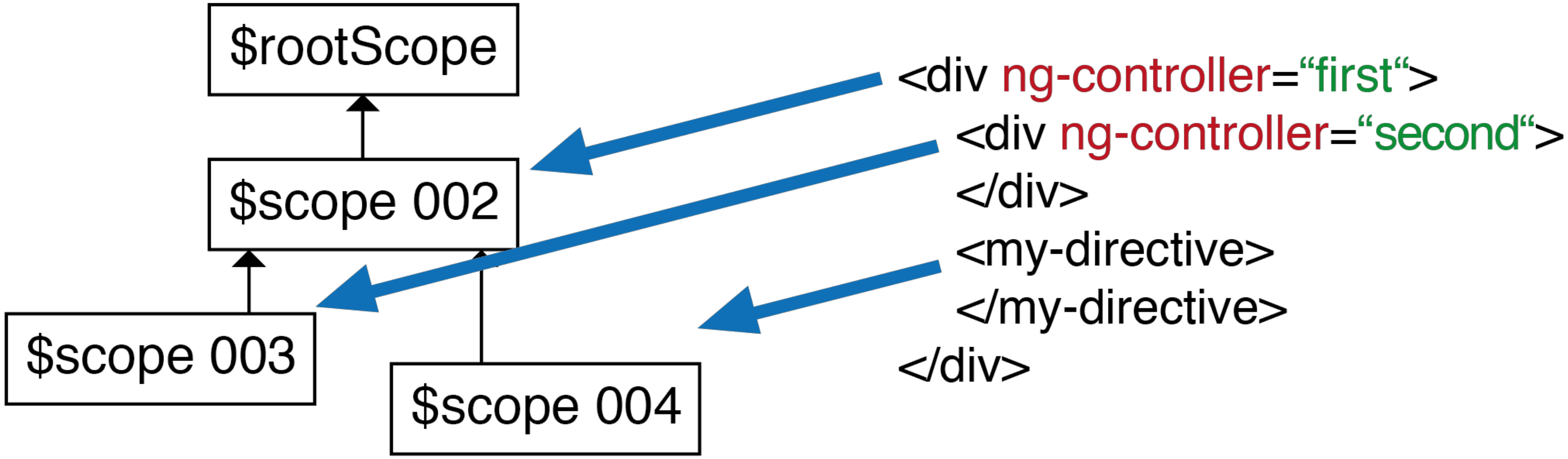


```
Send Request ((index))
Event (mousedown)
Event (click)
Paint Setup
Paint (2034 × 510)
Composite Layers
Timer Fired (chrome-extension://ognamp…
Receive Response ((index))
Receive Data ((index))
Recalculate Style
Finish Loading ((index))
Evaluate Script × 3
Send Request ((index))
Receive Response ((index))
Receive Data ((index))
Recalculate Style
Finish Loading ((index))
Parse HTML
```

☑ Loading ☑ Scripting ☑ Rendering ☑ Painting 38 of 278 records shown

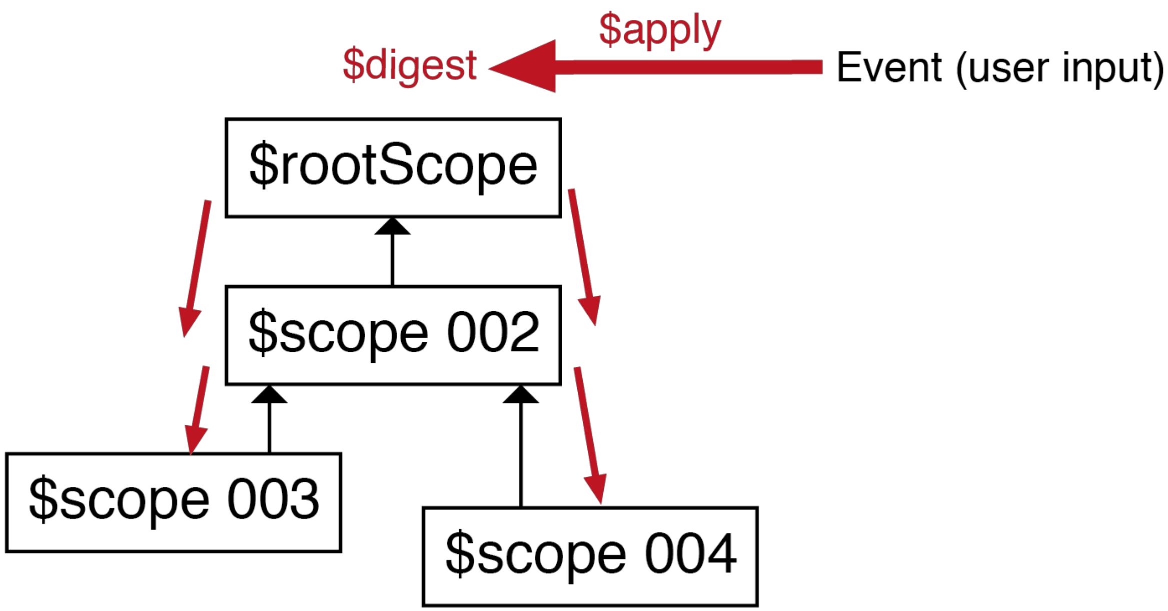
Behind the AngularJS magic

Scope Tree



```
<div ng-controller=“first“>
    <div ng-controller=“second“>
    </div>
    <my-directive>
    </my-directive>
</div>
```

Scope Tree



$apply

$digest

Event (user input)

$rootScope

$scope 002

$scope 003

$scope 004

One:
Address book

Demo | Demo fixed

Avoid watching invisible elements

limit number of watches, use ng-if instead of ng-show

Two:
Bakery

Demo | Demo fixed

No complex watch statements

precalculate and cache results, make watches simple and fast

Three:
Numbers

Demo | Demo fixed

Ensure DOM updates are batched

don't use ng-include in ng-repeat, prefill the template cache

Workflow

“Performance tuning – set expectations, measure, tune, and repeat” – {1}

1. Have a problem (no premature optimizations)
2. Set time/ budget limits
3. Measure correctly
4. Find main bottle necks
5. Fix bottlenecks one by one
   (constantly measure and evaluate new state)

Wrap up

- AngularJS detects model changes by dirty checking. Within the $digest cycle a previous stored value is compared to the actual model value. Process is repeated until models are stable
- One: Don't watch for invisible (ng-if > ng-show)
- Two: Avoid complex watch-expressions
- Three: Ensure DOM updtes are batched
- Workflow: detect problem, set expectations, measure, fix, repeat

References

1. Mastering Web Application Development with AngularJS, Pawel Kozlowski, Peter Bacon Darwin, Packt Publishing, 23. August 2013
2. AngularJS Performance Tuning for Long Lists
3. Optimizing AngularJS: 1200ms to 35ms
4. Analysing Performance of AngularJS Screens
5. Misco Hevery on AngularJS performance
6. AngularJS: 6 Common Pitfalls Using Scopes
7. Bindonce: Zero watches binding for AngularJS