WEB RESEARCH 2016-2017

YARN VS. NPM
VERGELIJKENDE STUDIE

Inleiding

Het NPM registry bevat ondertussen meer dan 300.000 packages voor niet alleen node.js maar zo goed als iedere programmeertaal mogelijk. Het is zonder twijfel de meest populaire oplossing voor package management binnen web development van dit moment. Ook voor open source is NPM een stap in de goede richting. Packages kunnen zonder problemen steunen op packages van andere developers die op hun beurt ook weer steunen op andere packages. Heel deze constructies noemt men de dependency tree.

Uiteraard zijn er ook enkele nadelen aan verbonden, denk maar aan de left-pad chaos waarbij enkele grote frameworks als jenga torens in elkaar storten toen één package gedwongen offline werd gehaald. Ook op vlak van security, consistentie is ruimte voor verbetering om nog maar te zwijgen over de performantie van een simpele `npm install`…

Gelukkig is er binnen Facebook een team ontstaan dat Yarn heeft ontwikkeld heeft met als doel deze kleine problemen op te lossen en de voordelen van NPM te behouden. Zo maken ze het leven van developers net iets dragelijker.

Features

Gelijkenissen

NPM registry

Yarn is absoluut geen complete vervanger van NPM maar louter een andere client voor het NPM registry. Dit wil zeggen dat alle NPM packages gewoon werken binnen Yarn. Dit

is dan ook de meest belangrijke gelijkenis. Wisselen tussen Yarn en NPM kan zonder problemen wat het ideaal maakt voor teams om geleidelijk aan over te stappen.

Open-source

Net als NPM is Yarn volledig open-source te vinden op github. Dit maakt het super gemakkelijk om toevoegingen of bugfixes toe te voegen als externe developer.

Verschillen

Parallelle installatie

Een van de grootste voordelen van Yarn is hoe snel package dependencies worden resolved en gedownload naar de node_modules folder. Dit komt door een iets andere workflow binnen Yarn. Waar NPM vroeger package per package ging downloaden in volgorde ging downloaden voert Yarn meerdere downloads tegelijk uit in parallel. Denk maar aan het verschil tussen single threading en multithreading. Dit resulteert in belachelijk snelle installaties (zie “Snelheid” voor de exacte getallen).

Cache

Deze parallelle downloads komen allemaal terecht in een gedeeld cache geheugen dat offline beschikbaar is. Wanneer je een tweede installatie opnieuw uitvoert wordt eerst dit cache geheugen aangesproken waardoor weer heel wat snelheid gewonnen wordt. Zo worden offline installaties mogelijk, iets wat met NPM niet kon.

Het yarn.lock bestand

Binnen het package.json bestand worden alle nodige packages voor een project vastgelegd, de volgorde van installatie en specifieke versie liggen vaak niet vast. Dit kan zorgen voor enkele vreemde “it works on my machine” bugs die bijna onmogelijk op te sporen zijn zonder een `rm -rf node_modules`.

Voor NPM bestaat er natuurlijk shrinkwrap. Dit legt alle versies van package.json vast binnen een apart bestand. Echter zonder voorkennis van de andere developers die met het project aan de slag gaan wordt dit vaak vergeten en gaat het nut verloren. Daarom zit deze shrinkwrap feature standaard in Yarn door middel van het yarn.lock bestand.

Net als de versies van de pakketten ligt ook de structuur van de dependency tree en de volgorde van package installatie vast. Dit zorgt ervoor dat het mogelijk wordt over verschillende machines exact dezelfde installatie uit te voeren.

Een ander voordeel dat dit yarn.lock bestand met zich meebrengt is dat breaking changes of bugs niet automatisch gepulld worden. Binnen automatische build systemen is dit een enorm voordeel.

Security

Voor het originele team binnen Facebook was security een van de problemen met NPM. Meer specifiek de post-, pre-install scripts, ik heb echter geen informatie gevonden over een mogelijke oplossing hiervoor binnen Yarn.

Wel worden checksums van iedere packet gecontrolleerd binnen de Yarn workflow, beide bij het downloaden naar cache als bij het kopiëren naar node_modules.

Bower support

Of toch niet. Bower werd recent verwijderd uit Yarn… en terecht.

Workflow

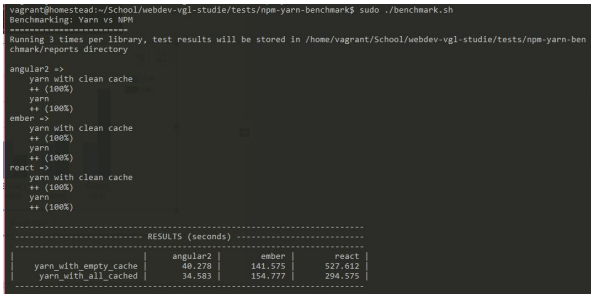
Het grootste voordeel van Yarn naar mijn mening is dat de workflow binnen een team volledig hetzelfde blijft. Yarn en NPM kunnen zonder problemen naast elkaar gebruikt worden. Voor wie toch overstapt zijn er enkele belangrijke commando's die bijna 1 op 1 overeen komen met hun NPM varianten.

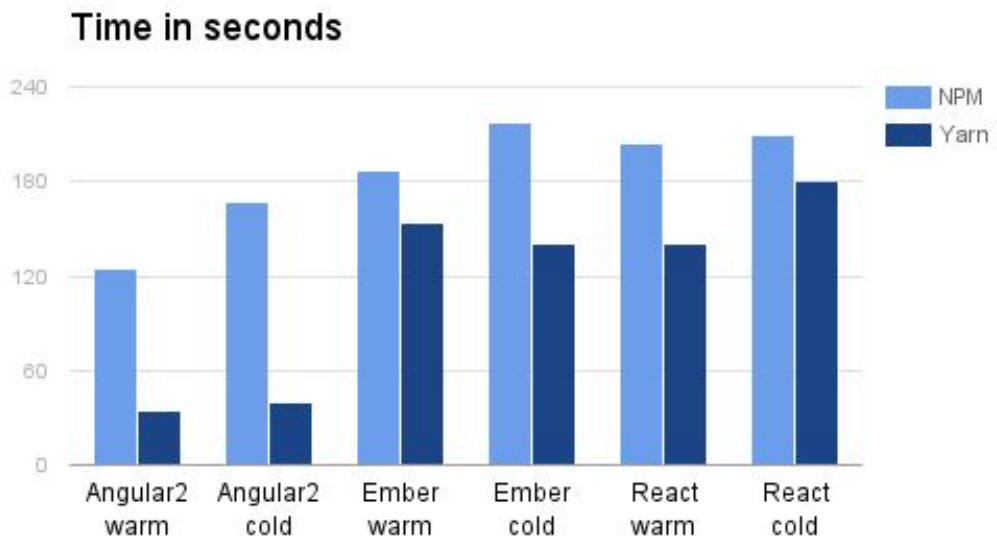
```
npm install                          → yarn
npm install --global                 →yarn global add
npm install --save                   →yarn add
npm cache clean                      →yarn cache clean
// en mijn persoonlijke favoriet:
rm -rf node_modules && npm install   → yarn upgrade
```

Snelheid

Zoals eerder vermeld is er een enorm snelheidsverschil tussen NPM en Yarn. Voor wie het niet geloofd: open een project, `yarn` en be amazed.

Toch wou ik enkele testjes doen met een warm/cold cache en met of zonder de node_modules folder aanwezig. Origineel gebruikte ik de `time` command binnen een Debian virtual machine op v0.16.1 om een gemiddelde snelheid te berekenen van enkele installs. Gelukkig kwam ik erachter dat iemand mijn werk voor mij had gedaan en een benchmarking script had geschreven (zie tests folder voor het script en nog enkele instructies om manueel tests uit te voeren). Enkel de `--no-bin-links` flag moest nog toegevoegd worden.





Conclusie

In het heel kort samengevat is Yarn ideaal op alle vlakken. Het valt perfect te gebruiken binnen teams en bestaande projecten met NPM, het is snel en het is stabieler. Wil je toch liever 5 minuten pauze bij iedere `npm install`? Dan kan je zonder problemen gewoon terug overschakelen op NPM.

Bronnen

http://www.theregister.co.uk/2016/03/23/npm_left_pad_chaos/

https://auth0.com/blog/five-things-you-can-do-with-yarn/

https://github.com/yarnpkg/yarn/pull/1441

https://yarnpkg.com/en/compare

https://yarnpkg.com/en/docs/migrating-from-npm

https://www.berriart.com/blog/2016/10/npm-yarn-benchmark/

https://code.facebook.com/posts/1840075619545360

https://gist.github.com/peterjmit/3864743

https://github.com/artberri/npm-yarn-benchmark

https://www.npmjs.com/