

Baseline Gatsby Benchmark: fs + markdown + images

This is a baseline benchmark site in the Gabe project.

This site in particular tracks Markdown performance for individual files per page that also have an image (not part of the markdown).

The site can generate an arbitrary amount of super simple pages. Each page has a small header, a quote, and two small paragraphs of random text. No images, because we want to benchmark Markdown.

The results of this benchmark can be compared to the results of the `gabe-fs-markdown` benchmark, to see a tentative impact of using images in markdown.

Install

Run `yarn` or `npm install`

Usage

Unlike most other gabe benchmarks, the generation part is a little more complex because it will generate image file pools first and then copy images from those pools into their destination.

Image generation

Image generation is rather expensive. The default size for 128k can take 2 hours single threaded. For that reason, the image generation can use workers instead.

Recommended way for larger pages is to first generate all the images up to the amount you're going to use. These pools will persist across benchmarks so it's a one time cost:

For example; to generate 128k 100x100 images using 8 worker threads:

```
C=8 W=100 H=100 N=128000
```

This will require an up to date node because workers aren't available in node 10.13, you'll get a warning if that's the case.

The files will be generated in `generated_image_pools/jpg/wxh`. If `C` is not set then it will only add images and assume the existing images are already properly incrementally numbered, without gaps.

If `C` is set (and used) then it will regenerate all images regardless and use that many workers to divide the work.

Image usage

When you run the benchmark, or generate the random content files, it will first check whether the pools have a sufficient amount of images. If they don't then the image pool is amended (see above).

Once the pool contains enough images for a given type/dimension, the random `.md` files are generated and for each file an image is copied from the pool as well. The copying of images is a lot faster.

It's important to note that the pool will persist between benchmark runs, while the randomly generated content does not.

Running the benchmark

Either way, you can start a benchmark run using the following. If the pool doesn't exist or does not have enough images, images will be generated:

```
W=100 H=200 N=1000 M=2 yarn bench
```

- `N=1000` : instructs the run to build a site of 1000 pages
- `M=2` : instructs nodejs to use up to 2gb of memory for its long term storage
- `W=100` : use images that are 100px wide
- `H=200` : use images that are 200px high
- `C=8` : (optional) force regenerate the image pool for given size and use 8 worker threads while doing so.
Only need to do this once per image type+dimension.
- Deletes generates files from previous run
- Generates `N` pages with pseudo-random content, copies one image from pool per page generated
- Runs `gatsby clean`
- Runs `gatsby build`

The default `yarn bench` will build 512 pages with 1gb memory.