The following examples cover some typical cache configurations. Note that for your individual projects, you may (and mostly will) need to alter them to some extent.

You can use similar configuration also for smaller projects or projects with lower load, in general:

- The more content the site has, the shorter caching period should be used to reduce load on the system's memory.
- The higher load you expect on the site, the higher levels of caching and longer caching intervals you should use.
- When there is a higher load to a specific part of the site, the section should use longer cache intervals. Balance this well with the first point.
- Configure shorter cache intervals if the process has less memory available.
- When you need to keep the content updatable, use either short enough caching intervals or levels of cache with properly configured flushing (cache dependencies).

Heavily accessed corporate sites

Websites of this type are mostly static, with lower amount of content, but a heavy load.

- Use static external cache or output cache based on the amount of load to handle.
- Use persistent output cache with large intervals.
- Use larger cache intervals on all the enabled caches as the content is mostly static.
- Simulate first loads to identify weak spots, use API caching and bulks loads to provide efficient first load times when cache is not available.
- Use client caching with larger interval, revalidation is usually not necessary.
- Use file caching with larger file size and larger cache interval.
- If possible, disable per user and other contextual caching.

Heavily accessed news sites

News sites are an example of frequently updated websites with heavy load.

- Use static external cache, preferably a Content delivery network (CDN) with shorter time interval (for example, 1 minute), make sure you use it for both pages and files.
- Use output cache and carefully set up output cache dependencies to allow efficient flushing of the cache. Use shorter time intervals as there is a large number of pages (for example, 10 minutes).
- Use client file caching with revalidation if you expect images and files to update, or without revalidation if you don't expect the published content to change. Use file server caching with shorter interval and file size to not spam the server memory too much.
- Simulate first loads to identify weak spots. It is likely that a such site will share content between multiple sections. This means that you may need to configure content caching with some meaningful time interval so that you don't spam memory too much (for example, 1 hour). The assumption is that the shared content won't have many unique instances.
- If possible, disable per user and other contextual caching.

Heavily accessed sites with login

For example, websites with member-only areas.

- If the only information which changes based on login is the username in header, configure output caching to NOT cache per user and leverage output substitutions to populate user name.
- If login influences the availability of content, use output caching configured to store content per user with shorter time period (more users = more similar cache items).
- If only smaller portions of an otherwise large website are influenced by login, consider using partial caching to lower the memory footprint.

Heavily accessed e-commerce sites

- Use output cache with shorter time interval (for example, 1 minute). Product prices may change based on factors other than discounts, and this helps you keep the customer updated.
- Use output caching per user as the content may vary by user.
- Do not use caching for checkout process pages, these need to be loaded with new information every time.

- Use partial caching on master page components and static components of checkout process to make it as efficient as possible (even when its internals do not use caching). Use larger time interval on the components as they will be static most of the time.
- Use file client caching without revalidation, as customers will probably visit same listings/products several times during their session.
- Use file server caching with longer time interval (this assumes that images used on the site won't change that much) and lower file size limit (most of the files will be product images).
- When there are a lot of products in the E-shop, lower the caching intervals to reduce load on the system's memory.

- Use partial caching on master page components and static components of checkout process to make it as efficient as possible (even when its internals do not use caching). Use larger time interval on the components as they will be static most of the time.