Caching

Hva er caching

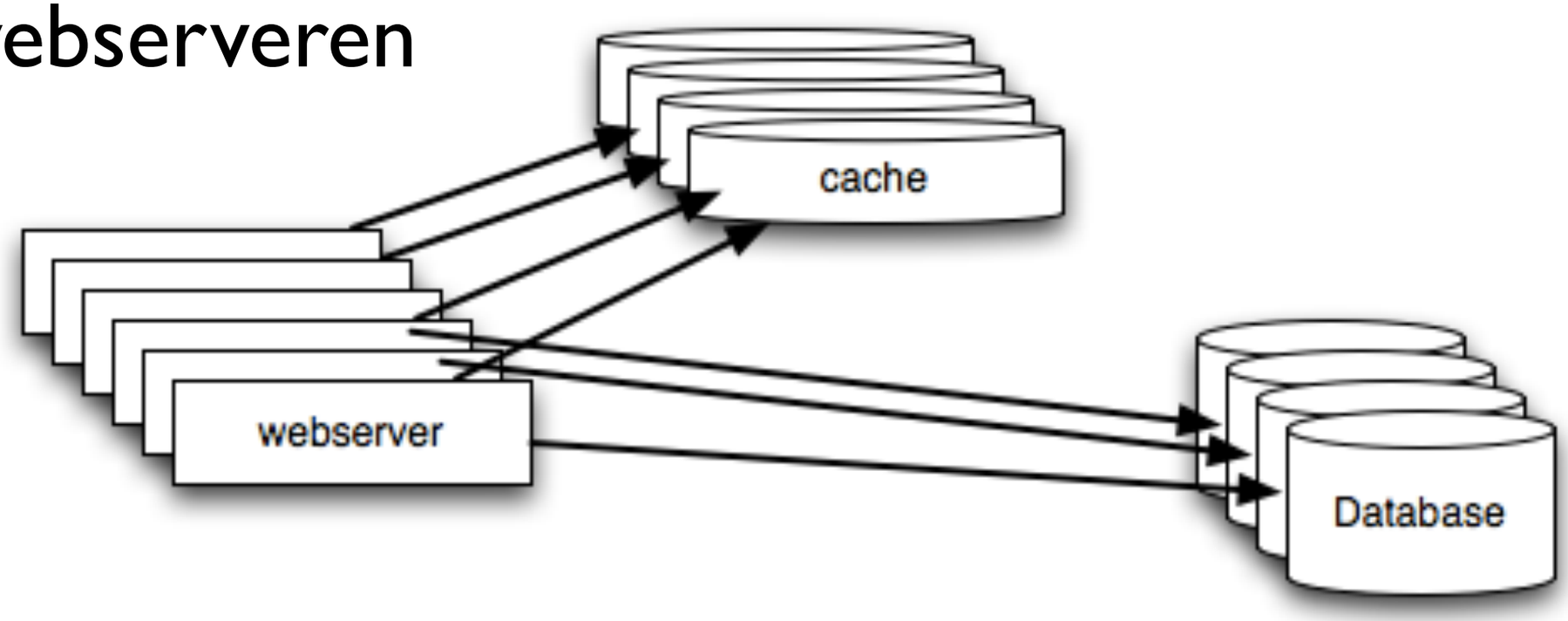
- Godt kjent i hardware og operativsystemer
- Mindre kjent i webtjenester
  - Blitt veldig populært de siste to årene
- Kan gjøres på forskjellige måter:
  - Applikasjonsnivå
  - Tjenestenivå

Caching på applikasjonsnivå

- Eksempel: memcache
  - Memcache er en enkel, distribuert "nøkkel -> verdi" tjeneste
  - Applikasjonen sjekker alltid memcache først før den gjør et oppslag i databasen
  - Dersom resultatet finnes i cache, blir det brukt, og databasen blir ikke kontaktet
  - Dersom noen gjør en endring i databasen, fjerner man cache-objektet fra memcache
    - Dette må gjøres av applikasjonen og kan fort føre til krøll

Memcache arkitektur

- Memcache er med andre ord "database-blind"
- Ingen redundans i lagringen, objektene blir fordelt mellom memcahe serverne
- Kan brukes i mange programmeringsspråk
- En variasjon er å kjøre memcache på selve webserveren



Fordeler / Ulemper

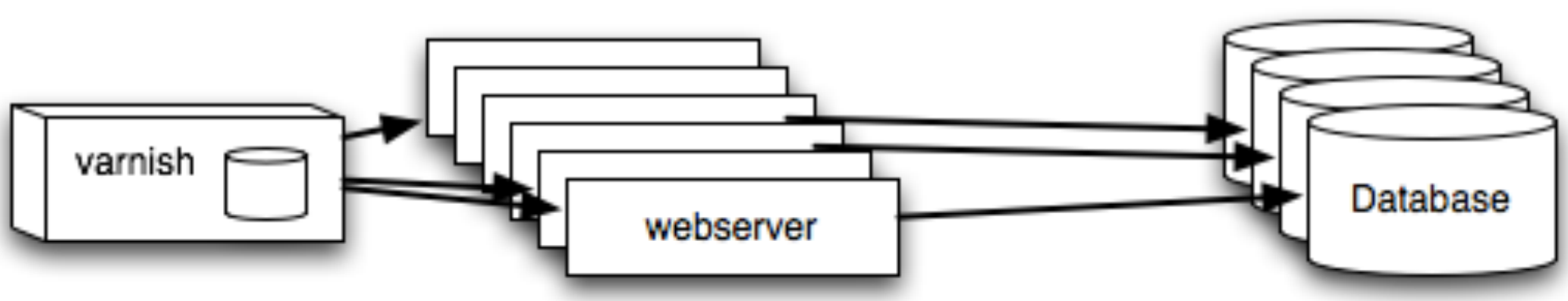
- Kan relativt enkelt implementeres i etterkant i en ny versjon av tjenesten
  - Kan føre til dramatisk ytelsesøkning og responstid, spesielt ved svake databaser og få oppdateringer
- Man må holde tungen veldig rett i munnen
  - Glemmer man f.eks å slette objekter som er blitt endret eller sletter unødvendig ofte, kan tjenesten bli svært langsom

Caching på HTTP-nivå

- Eksempel: Varnish
  - En norskutviklet proxy som sitter foran webserverne og mellomlagrer websider som kan bli gjenbrukt
  - Fungerer med vilkårlige lastbalanserere
  - Kan brukes som innholds-basert lastbalansering, men er sterkest om den samtidig mellomlagrer
  - Har et eget "programmerinsgsspråk" som kan brukes til å beskrive hvordan div. HTTP spørringer skal håndteres

Varnish arkitektur

- Er kun en web-proxy, kan ikke brukes i andre sammenhenger
- Man kan ha flere varnish servere
  - Ingen redundans i cachingen her heller



Fordeler / Ulemper

- Potensielt veldig kraftig løsning hvor man ikke trenger endre noe på selve web-siden
- Man må allikevel programmere varnish selv for å få det resultatet man vil
  - Krever innsikt i varnish og HTTP protokollen, ofte mer ukjent enn PHP eller .NET
  - Enkelte ganger må man endre på websidene også, men da dreier det seg om å plante "hint" i HTTP-headerne

Memcache Installasjon

IMT3441 kyrre.Begnum@ntnu.no

Memcached

- Memcached - memcache daemon
  - memcached.org
- Linux-basert daemon som brukes av bl.a
  - Twitter, YouTube, Facebook, Craigslist, Digg, Wordpress, Flickr, LiveJournal
- Følger enkelt prinsipp:

```
function get_foo(foo_id)
    foo = memcached_get("foo:" . foo_id)
    return foo if defined foo

    foo = fetch_foo_from_database(foo_id)
    memcached_set("foo:" . foo_id, foo)
    return foo
end
```

Installasjon

- Installer pakken:
  `apt-get install memcached`
- Editer konfigurasjonsfilen: /etc/memcached.conf
  - Hvilken port skal man lytte på? Kun intern trafikk er default. Endre til:
    `-l 0.0.0.0`
  - Hvor mye minne skal brukes til caching?
    `-m 128`
- Restart tjenesten:
  `service memcached restart`

Apache + memcache

- Egentlig vet apache selv ingenting om memcache. Man må imidlertid installere biblioteket til PHP

  `apt-get install php5-memcache libmemcached10 libmemcache-dev`

- Deretter må man restarte apache for å reloade PHP

  `service apache2 restart`

Memcache + munin

- Laste ned plugins
  - En zip-fil ligger i fronter under “Resources” med navn memcache_plugins.tar.gz
- Munin plugins’ene er skrevet i Perl, så vi trenger å installere Perl-memcache bibliotek

  `apt-get install libcache-memcached-perl`
- Restart munin-node

  `service munin-node restart`

Bookface og memcache

- Bookface v2 er allerede programmert til å kunne bruke memcache, men det er skrudd av
- Man må legge til tre linjer i `/var/www/html/config.php`

```
$memcache_enabled = 1;
$memcache_enabled_pictures = 1;
$memcache_server = “IP til memcached”;
```