Performance through caching

Aurélien Broszniowski
@AurBroszniowski

Aurélien Broszniowski

Lead Engineer at Terracotta/Software AG

Ehcache!

Initiated the Rainfall framework.
Testing JSR107 caches.

Extension to a generic stress and performance testing framework.

What we'll cover

- JCache (jsr107)
- Cache Aside
- Cache-Through
- Performance testing
- Testing jsr107 providers
- Statistics
- Beyond JCache… capacity
- Beyond JCache… topology

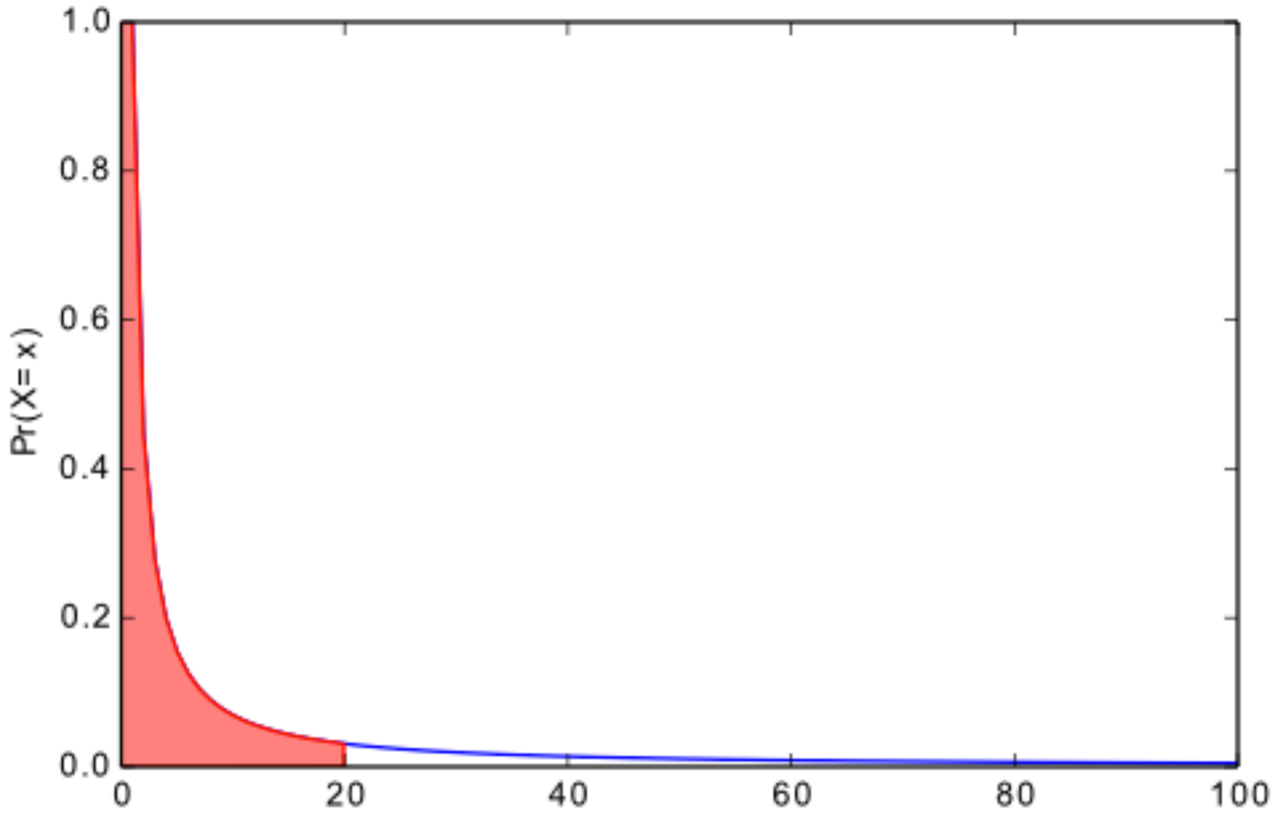
Introductory poll

- Who knows nothing about caching?
- Who already uses caching in production?
- Who had caching related problems in production?
- Who knows about JSR 107?

What is a cache?

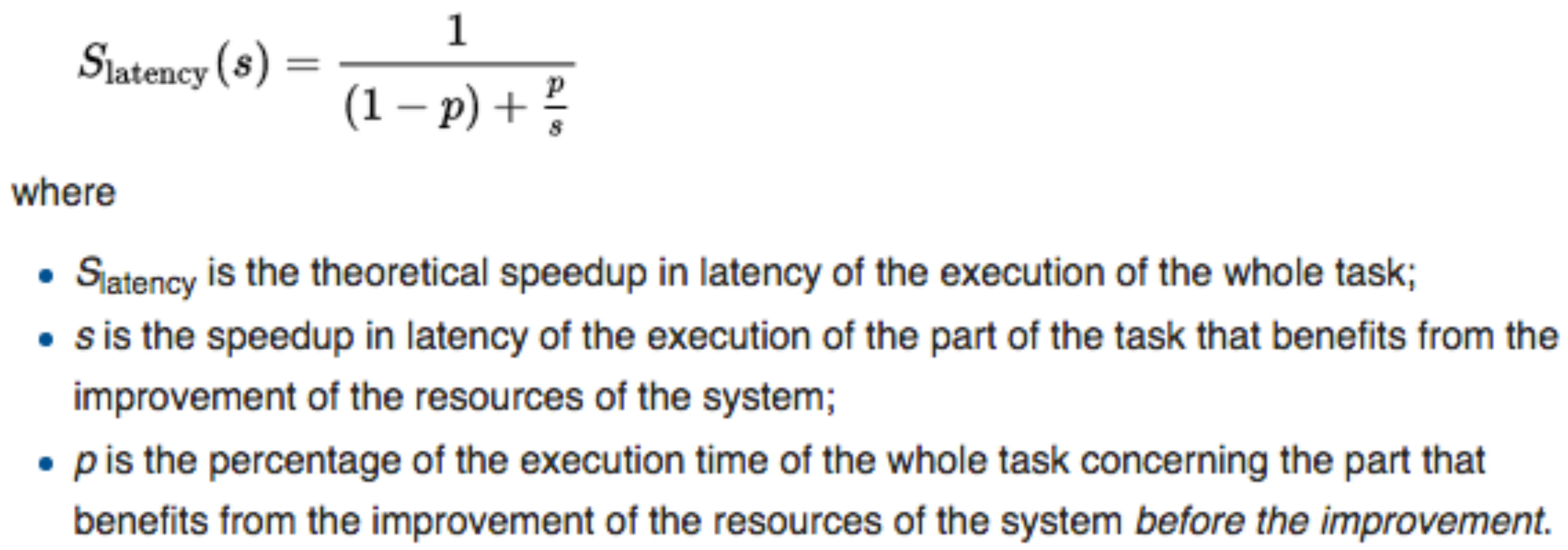
- Data structure holding a ***temporary*** copy of some data
- Trade off between ***higher memory*** usage for ***reduced latency***
- Targets:
  - Data which is reused
  - Data which is expensive to compute or retrieve

Why does caching work?



- Pareto distribution : The long tail
- 80/20

Why does caching work?



$$S_{\text{latency}}(s) = \frac{1}{(1-p)+\frac{p}{s}}$$

where

- S_{latency} is the theoretical speedup in latency of the execution of the whole task;
- s is the speedup in latency of the execution of the part of the task that benefits from the improvement of the resources of the system;
- p is the percentage of the execution time of the whole task concerning the part that benefits from the improvement of the resources of the system *before the improvement*.

- Amdahl's law

Possible usages

- CPU bound applications
  - Normal speedup through algorithm improvement or parallelisation
  - Cache can help by storing computation results
- I/O bound applications
  - Normal speedup through disk or network upgrades
  - Cache can help by storing data locally

JSR 107
JCache

JSR 107

- Java Community Process driven standard
  - Specifies API and semantics for **temporary**, **in-memory caching** of Java objects, including object creation, shared access, spooling, invalidation, and consistency across JVM's

javax.cache API

`CacheManager` repository,

```
CachingProvider provider = Caching.getCachingProvider();

CacheManager cacheManager = provider.getCacheManager();

Cache<Long, String> myCache = cacheManager.getCache("myCache", Long.class, String.class);
```

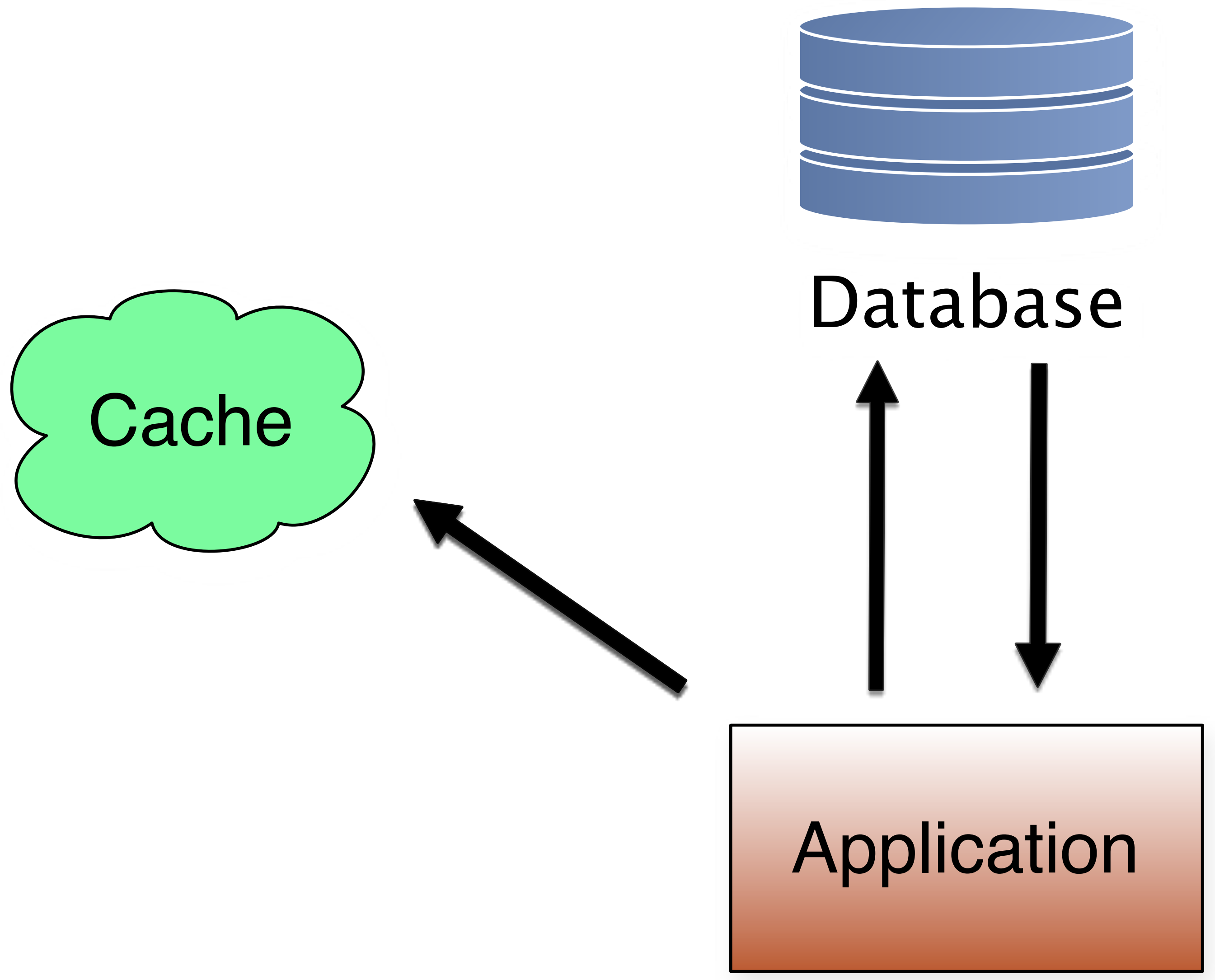
Named `Cache` repository, handles their lifecycle

`ConcurrentMap<K, V>` sibling, major interaction

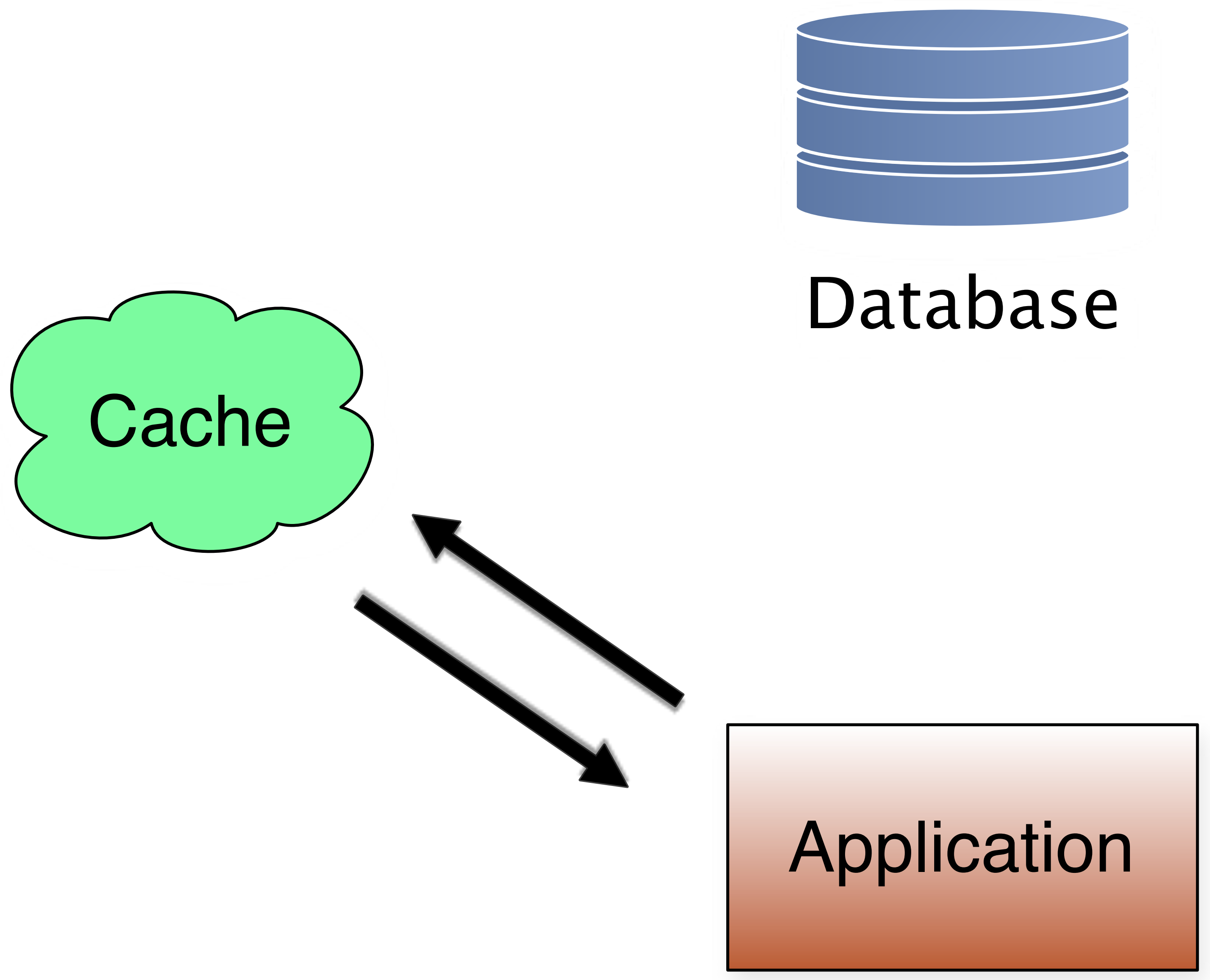
Cache patterns

Cache aside

Cache aside - miss



Cache aside - hit



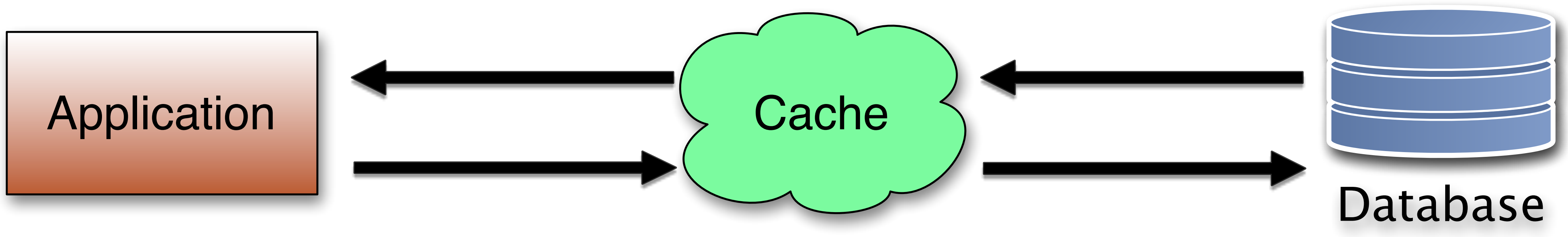
Exercise

https://github.com/aurbroszniowski/jbcn2016

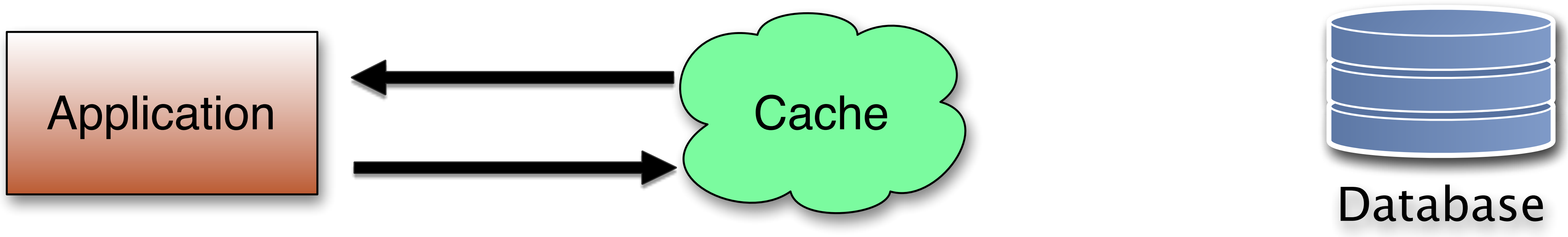
Cache patterns

Cache through

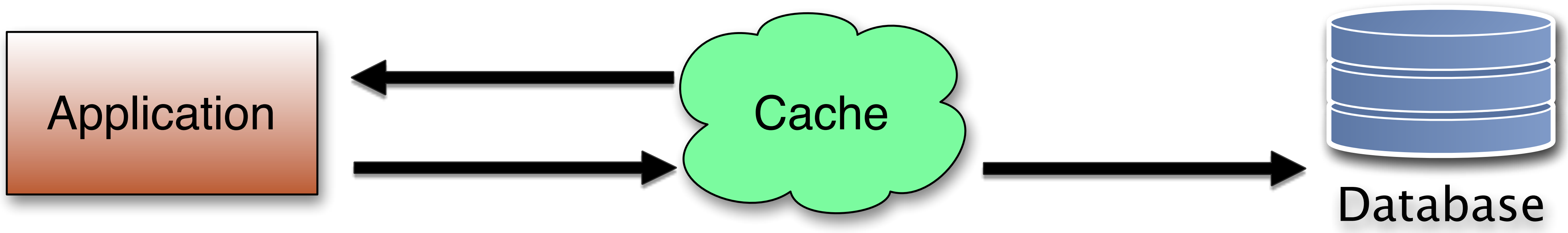
Cache through - miss



Cache through - hit



Cache through - write



Exercise

https://github.com/aurbroszniowski/jbcn2016

Performance testing

Pitfalls

- Warm-up phase (JIT optimisation!)
- Run reasonable length of time
- Do not average results!
- Think about amortization
- Pay attention to variability

Common actions

- Defining a Scenario of operations
- Executing the Scenario
- Gathering statistics
- Reporting results

Rainfall framework

```
Runner.setUp(
    Scenario.scenario("load test")
        .exec(new Operation() {
          @Override
          public void exec(…) throws TestException {

            long start = getTimeInNs();
            // This is what we measure
            service.someLogic(id);
            //
            long end = getTimeInNs();
            statisticsHolder.record("READ", (end - start), READ);
          }
       }))
    .warmup(during(45, TimeDivision.seconds))
    .executed(during(1, TimeDivision.minutes))
    .config(report(Results.class).log(text(), html()))
    .start();
```

Exercise

https://github.com/aurbroszniowski/jbcn2016

JSR 107 providers

JSR 107

- Ehcache
- Hazelcast
- Infinispan
- Apache Ignite
- more…

Exercise

https://github.com/aurbroszniowski/jbcn2016

Statistics

Statistics

- **HIT or MISS ? 50%!**
- **Statistics MBean**

```
javax.cache:type=CacheStatistics,CacheManager=urn.X-
ehcache.jsr107-default-config,Cache=yourcache
```

Exercise

https://github.com/aurbroszniowski/jbcn2016

Beyond JCache

Beyond JCache

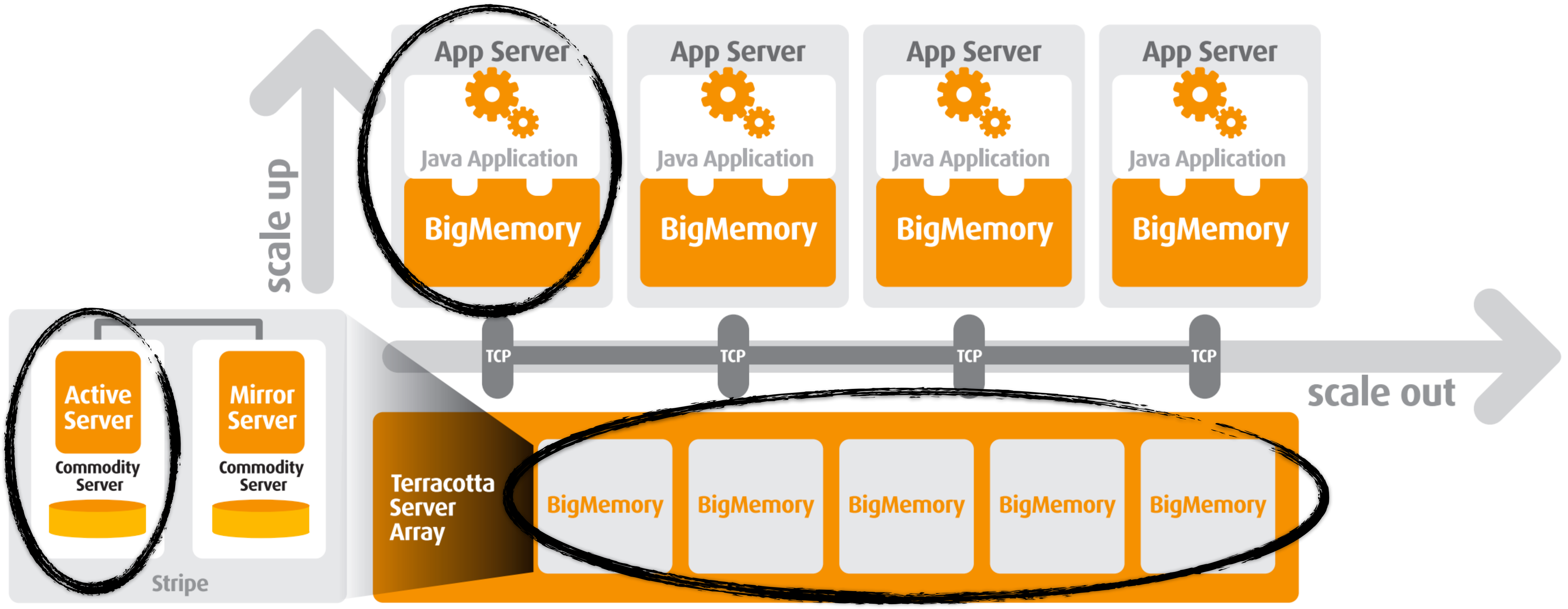
```
CacheConfiguration<String, String> cacheConfiguration =
  CacheConfigurationBuilder.newCacheConfigurationBuilder(
      String.class, String.class,
      ResourcePoolsBuilder.heap(10000))
  .build();

cache = cacheManager.createCache("someCache",
     Eh107Configuration.fromEhcacheCacheConfiguration(
        cacheConfiguration));
```

Exercise

https://github.com/aurbroszniowski/jbcn2016

Beyond JCache : Cache topologies

Terracotta clustering

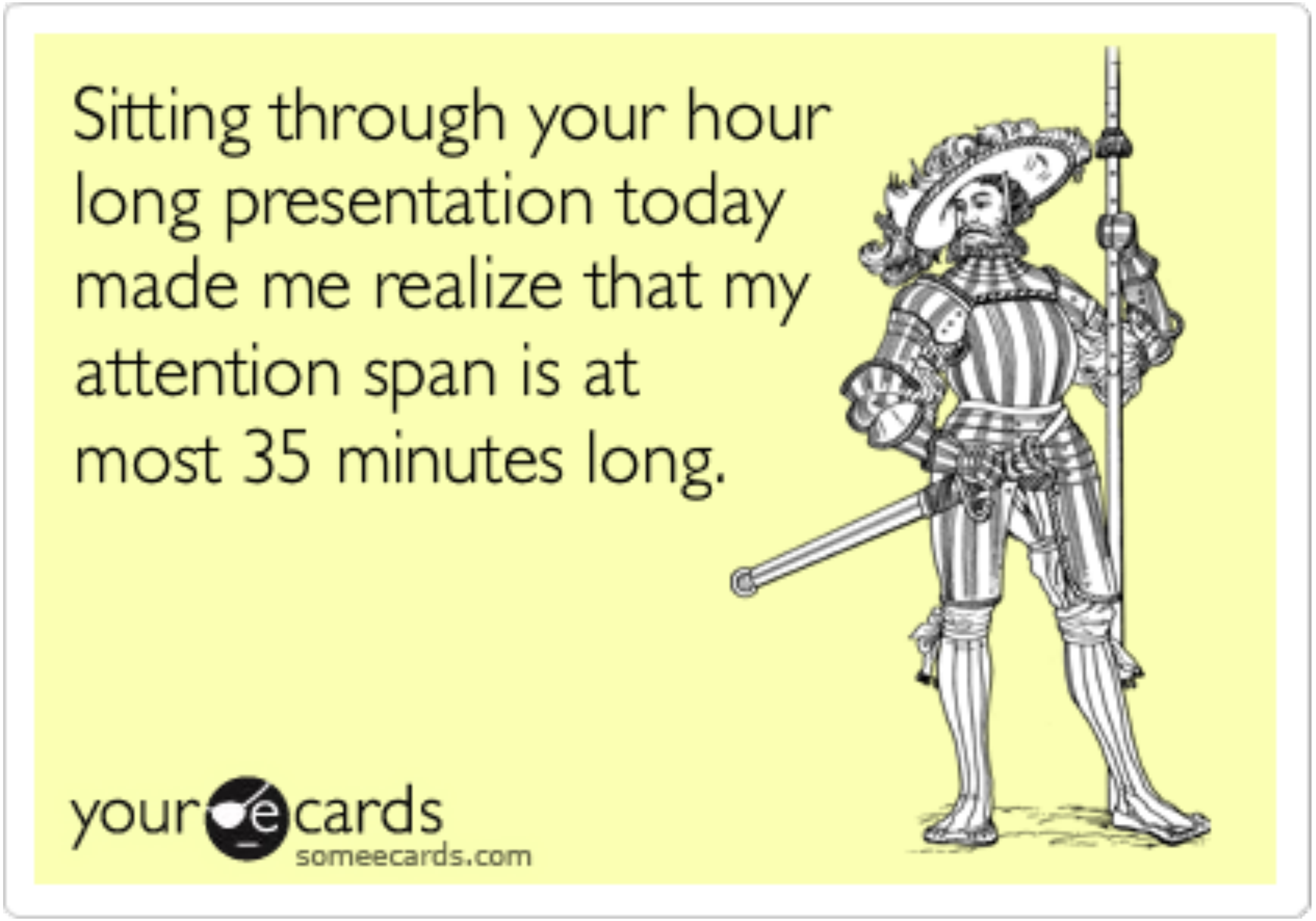


www.ehcache.org

www.rainfall.io

Thanks!

@AurBroszniowski


Sitting through your hour long presentation today made me realize that my attention span is at most 35 minutes long.
your ecards
someecards.com