Web performances

@k33g_org
@nicolasleroux
!@RemiForax
@jeffmaury
@ehsavoie

DEVOXX™ France

DEVOXX™ France

Agenda

- Why ?
- The application
- The selected stacks/frameworks
- Infrastructure and tools
- Tests
- Conclusion

DEVOXX™ France

The team

- Philippe Charriere (@k33g_org)
- Remi Forax
- Nicolas Leroux (@nicolasleroux)
- Jeff Maury (@jeffmaury)
- Emmanuel Hugonnet (@ehsavoie)

DEVOXX™ France

Many thanks to

Goals

DEVOXX™

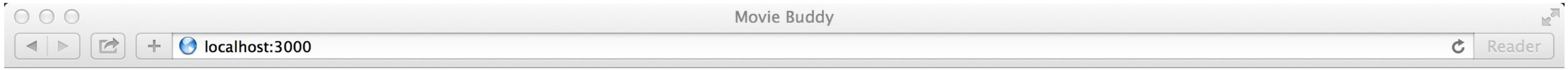
DEVOXX™ France

Goals

- Feedback on http://blog.shinetech.com/2013/10/22/performance-comparison-between-node-js-and-java-ee/
- Show how simple can it be
- No backend, useless slowdown
- Production oriented, standard stacks are used
- Cloud is our friend
- State of the art on used technologies

The application

- Recommandation system
- 10000 users, 800 movies
- User, movie search
- Users vote for movies
- User similarity : compute distance (euclidean) between 2 users



Movie Buddy

Users | Movies | ...

Thu Apr 17 2014 16:01:34 GMT+0200 (CEST)

Search users by name

bob Search

id	name	
2164	Bobby Stamm	select
3902	Dr. Bobbie Veum	select
4806	Bobbie Miller	select
5707	Bobby Veum	select
5965	Bobby Harªann	select
6250	Dr. Bobby McLaughlin	select
7056	Bobbie Strosin	select

Movie Buddy

Users | Movies | ...

Dr. Bobbie Veum has just rated Police Academy : 4

Search movies

title Search Comedy Search actors Search

Title	Genre	Actors	Rating
Rick Santorum Aborts Presidential Campaign	Short, Comedy	Ashley Judd, Michelle Trachtenberg, Katy Mixon, Eliza Coupe	0 1 2 3 4 5
Gary and the Store Button, Howard Addresses Tonight Show Rumor, Howard and Fred Write a Song, and Scott the Engineer's Bowling Ring	Comedy, Talk-Show	Gary Dell'Abate, Robin Quivers, Scott Salem, Howard Stern	0 1 2 3 4 5
Wataru to himiko no achi achi adobenchâ	Animation, Adventure, Comedy	N/A	0 1 2 3 4 5
Police Academy	Comedy, Crime	Steve Guttenberg, Kim Cattrall, G.W. Bailey, Bubba Smith	0 1 2 3 4 5
The Adding Machine	Comedy, Fantasy	Milo O'Shea, Phyllis Diller, Billie Whitelaw, Sydney Chaplin	0 1 2 3 4 5
Acupressure	Short, Comedy	Steve Exeter, Mike Lukey	0 1 2 3 4 5

The application

- REST/JSON services:
  - Users : list, read, search, distance
  - Movies : list, read, search
  - Vote : vote with ranking
- Front-end :
  - Bootstrap/JQuery/Backbone/React.JS

APIs : movies

- GET /movies : returns the list of movies
- GET /movies/search/genre/criteria/limit
  - Returns the list of movies satisfying genre criteria with a limit

APIs : vote

- POST /rates {« userid » : userid, « movieId »:movieid, « rate » : rate}
  - Records the vote of a user for a movie

APIs : users

- GET /users/distance/id1/id2 :
  - Compute the distance between two users

Implementations

- JEE/JAX-RS (Java)
- NodeJS/Express (Javascript)
- Play2/Scala (Scala)
- Undertow (Java)
- Vertx (Java)
- Finatra (Scala)

J2EE/JAX-RS

- Endpoints REST based on JAX-RS annotations
- Tomcat container
- JAX-RS implementation : Jersey

J2EE/JAX-RS code

```
@GET
@Path("/")
@Produces(MediaType.APPLICATION_JSON)
public void getMovies(@Context HttpServletResponse response) throws IOException {
  PrintWriter writer = response.getWriter();
  writer.write("[");
  for(Movie movie : movieService.listMovies()) {
    writer.write(movie.toString());
    writer.write(",");
  }
  writer.write("]");
}
```

J2EE/JAX-RS code

```
@GET
@Path("search/genre/{genre}/{limit}")
@Produces(MediaType.APPLICATION_JSON)
public void searchByGenre(@Context HttpServletResponse response, @PathParam("genre") String genre,
        @PathParam("limit") String limit) throws IOException {
  PrintWriter writer = response.getWriter();
  writer.write("[");
  for(Movie movie : movieService.findByGenre(genre, Integer.parseInt(limit))) {
    writer.write(movie.toString());
  }
  writer.write("]");
  writer.flush();
}
```

Play 2.2 overview

- Playframework is a full-stack framework.
- Support for Form, Validation, Redirection, etc..
- Json → Object Java / Scala
- Library for manipulation data flow
- Elegant code

Play 2.2 – Get all movies

Routes file

```
GET     /movies                 controllers.Application.movies
```

File Application.scala

```
def movies = Action {
        Ok.chunked(Enumerator.fromStream(Play.resourceAsStream("movies.json").get))
         .as("application/json")
}
```

Play 2.2 – Get movies per genre

```
GET   /movies/search/genre/:genre/:limit
controllers.Application.movieByGenre(genre: String, limit: Int)
```

```
def movieByGenre(genre: String, limit: Int) = Action {
        val pattern = Pattern.compile(genre.toLowerCase);
  val movies:Seq[Movie] = Repository.movies
        .filter(movie => pattern.matcher(movie.genre.toLowerCase)
          .find())
        .take(limit)
        }
  Ok(Json.toJson(movies))
}
```

DEVOXX™ France

Express + Node

- Node : platform based on Chrome V8 Javascript Engine
- JS plateform from back to front
- Ease of exchange with the browser
- Express : Framework for developping Webapps
- Pros : native JSON
- Cluster module...

Express + Node / Code

```
1  app.get("/movies", function(req, res) {
2    res.send(movies);
3  });
4
5  app.get("/movies/search/genre/:genre/:limit", function(req, res) {
6    res.send(movies.filter(function(movie) {
7      return movie.Genre.toLowerCase().
8        search(new RegExp(req.params.genre.toLowerCase()),"g") != -1;
9    }).slice(0,req.params.limit));
10 });
```

Finatra

- Web framework from Twitter, Scala based inspired by Sinatra
- Looks like Express, with typing
- Easy to setup, even for a Scala beginner
- Scala standard library for JSON parsing

  POST :`JSON.parseFull(request.getContentString()).get.asInstanceOf[Map[String, Double]]`

Finatra / Code

```
1  // moviesList : List[Map[String,Any]]
2
3  get("/movies") { request =>
4    render.json(moviesList).toFuture
5  }
6
7  get("/movies/search/genre/:genre/:limit") { request =>
8    val genre = request.routeParams.getOrElse("genre","?")
9    val limit = request.routeParams.getOrElse("limit",1).toString()
10   val searchedMovies = moviesList.filter(movie => {
11     movie("Genre").toString().toLowerCase().contains(genre)
12   }).slice(0, limit.toInt)
13   render.json(searchedMovies).toFuture
14 }
```

Vertx in one slide

- Polyglot application, heavily inspired by node.js but multi-threaded
- Based on Netty for Networking, Jackson for Json
- Deploy sources, no Maven, no Gradle, no Ant ??
- Based on Java 7

Vertx + Java 8 code

```
route.get("/movies", req -> {
  req.response().sendFile("/db/movies.json");
});

route.get("/movies/search/genre/:genre/:limit", req -> {
  Pattern pattern = compile(req.params().get("genre").toLowerCase());
  req.response().end(
    movies.stream()
      .filter(movie -> pattern.matcher(movie.genre).find())
      .map(Movie::toString)
      .limit(parseInt(req.params().get("limit")))
      .collect(joining(",", "[", "]")));
});
```

Vertx @ Jackson vs Java 8

Vertx can use lambda out of the box
    All handlers are interfaces with one method

But not support of new Java 8 classes
  (java.util.Stream)

For Jackson, no automatic mapping json file -> Stream
    only 120 lines of code

For Vertx, async read but sync-like write
    the Vertx response should take a Stream as parameter

Undertow / in one slide

Use non standard NIO (XNIO)

Flexible

full Java EE servlet 3.1 container

low level specific non blocking handler API

Embeddable

Undertow Servlet / Code

```
DeploymentInfo servletBuilder = Servlets.deployment()
  .setClassLoader(SearchMoviesServlet.class.getClassLoader())
  .setContextPath(MYAPP)
  .addWelcomePage("index.html")
  .setResourceManager(new ClassPathResourceManager(SearchMoviesServlet.class.getClassLoader()))
  .setDeploymentName("moviebuddy.war")
  .addServlets(
      Servlets.servlet("Movies", SearchMoviesServlet.class)
      .addMapping("/movies")
      .addMapping("/movies/*")
      .setAsyncSupported(true),...
);

DeploymentManager manager = Servlets.defaultContainer().addDeployment(servletBuilder);
manager.deploy();
PathHandler path = Handlers.path(Handlers.redirect(MYAPP))
        .addPrefixPath(MYAPP, manager.start());

Undertow server = Undertow.builder()
  .addHttpListener(port, hostName).setHandler(path)
  .setBufferSize(1024 * 16).setWorkerThreads(50).build();
server.start();
```

Declaring a servlet

Starting the server

Undertow Handler / Code

```
Undertow server = Undertow.builder()
  .setHandler(Handlers.header(Handlers.path(Handlers.resource(
    new WebappClassPathResourceManager(MYAPP,
    SearchMoviesHandler.class.getClassLoader())))
      .addPrefixPath(MYAPP + "/rates", new RatesHandler())
      .addPrefixPath(MYAPP + "/users", new SearchUsersHandler())
      .addPrefixPath(MYAPP + "/movies", new SearchMoviesHandler()),
        Headers.SERVER_STRING, "U-tow")).build();
server.start();
```

Undertow Handler / Code

```
protected void doGet(HttpServerExchange exchange) throws Exception {
    final String[] params = URLParser.parse(PREFIX, exchange);
    String result = RateService.INSTANCE.findRateByUser(Integer.parseInt(params[0]));
    //Sending the response in an nio way, ByteBuffer can be used.
    if (result == null) {
        exchange.setResponseCode(StatusCodes.NOT_FOUND);
    } else {
        exchange.getResponseHeaders().add(Headers.CONTENT_TYPE, "application/json");
        exchange.getResponseSender().send(result);
    }
    exchange.endExchange();
}
protected void doPost(HttpServerExchange exchange) throws Exception {
    exchange.startBlocking(); //We are going to read from the request inputstream
    List<JsonItem> items = JsonLoader.load(exchange.getInputStream());
    exchange.setResponseCode(StatusCodes.MOVED_PERMENANTLY);
    exchange.getResponseHeaders().put(Headers.LOCATION, "http://" + exchange.getHostAndPort()
            + MYAPP + RateService.INSTANCE.rateMovie(items.get(0)));
    exchange.endExchange();
}
@Override
public void handleRequest(HttpServerExchange exchange) throws Exception {
    //Only method to implement
    if (Methods.POST.equals(exchange.getRequestMethod())) {
        doPost(exchange);} else { doGet(exchange); }
}
```

Infrastructure

- Server cloud hosted (Cloudbees)
- Cloudbees provides most of the containers : Tomcat, Vertx, Play, Node
- Finatra and Undertow via the Java standalone container
- nginx frontend for load balancing

Load testing tool

- Gatling : gatling-tool.org
- Scala DSL for scenarios
- Very reactive support (thanks S LANDELLE)
- Reactive architecture for large load scenarios on commodity machines
- Tests are run as Jenkins jobs : Maven plugin, Jenkins Jenkins for tests results archival

Scenarios

- Three scenarios :
  - MovieLoading : load the list of movies
  - Vote : movies search, then vote
  - Distance : compute the distance between two pairs of users
- Scenarios simulate 100 simulatenous users, without pause looping 5000 times

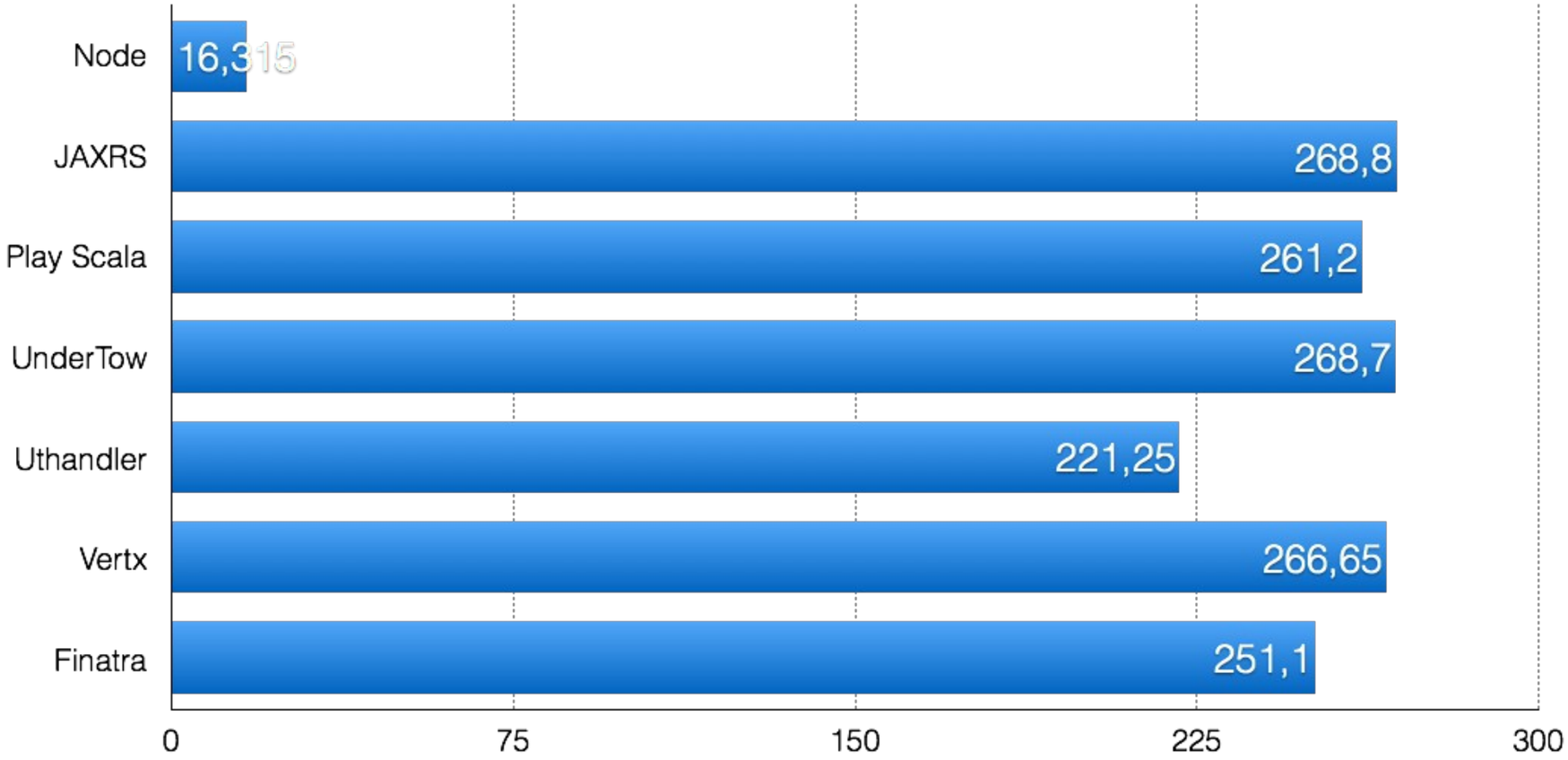
Scenario

```
class DistanceScenario extends Simulation {
  val server = System.getProperty("buddyserver", "http://localhost:8080")
  val totalUsers = Integer.getInteger("gatling.users", 100).toInt
  val loops = Integer.getInteger("gatling.loops", 1000).toInt
  val protocol = http.disableCaching.disableFollowRedirect

  val scn = scenario(s"Distance ($totalUsers users/$loops loops)")
    .repeat(loops) {
      exec(
        http("Distance 3022 <-> 9649")
          .get(server + "/users/distance/3022/9649")
          .check(status.is(200))).
      exec(
        http("Distance 2349 <-> 496")
          .get(server + "/users/distance/2349/496")
          .check(status.is(200)))
    }
  setUp(scn
    .inject(rampUsers(totalUsers) over (totalUsers seconds)).protocols(protocol))
}
```

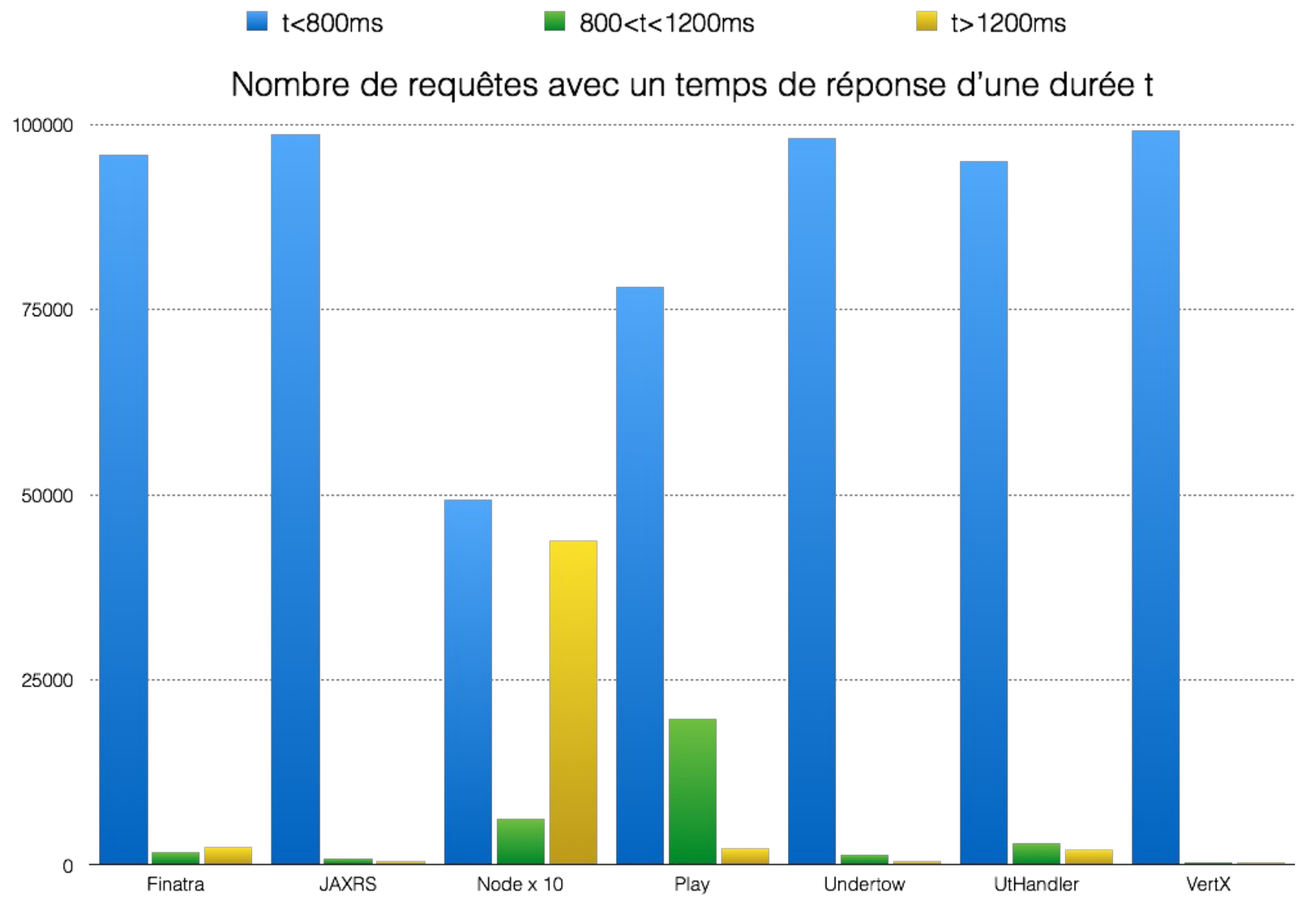
MovieLoading scenario

- Loading of the movies database (400Ko)
- IO bound
- Focus on framework IO management

Nb Reqs/sec : All Movies



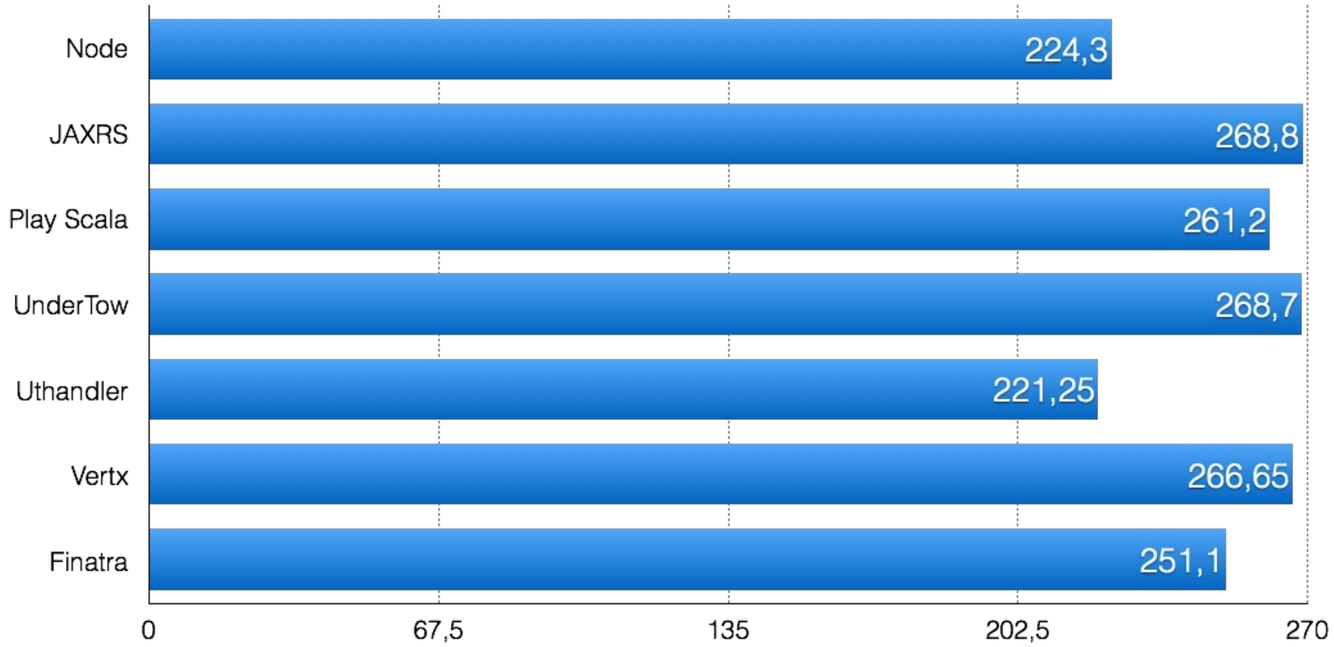
Response times



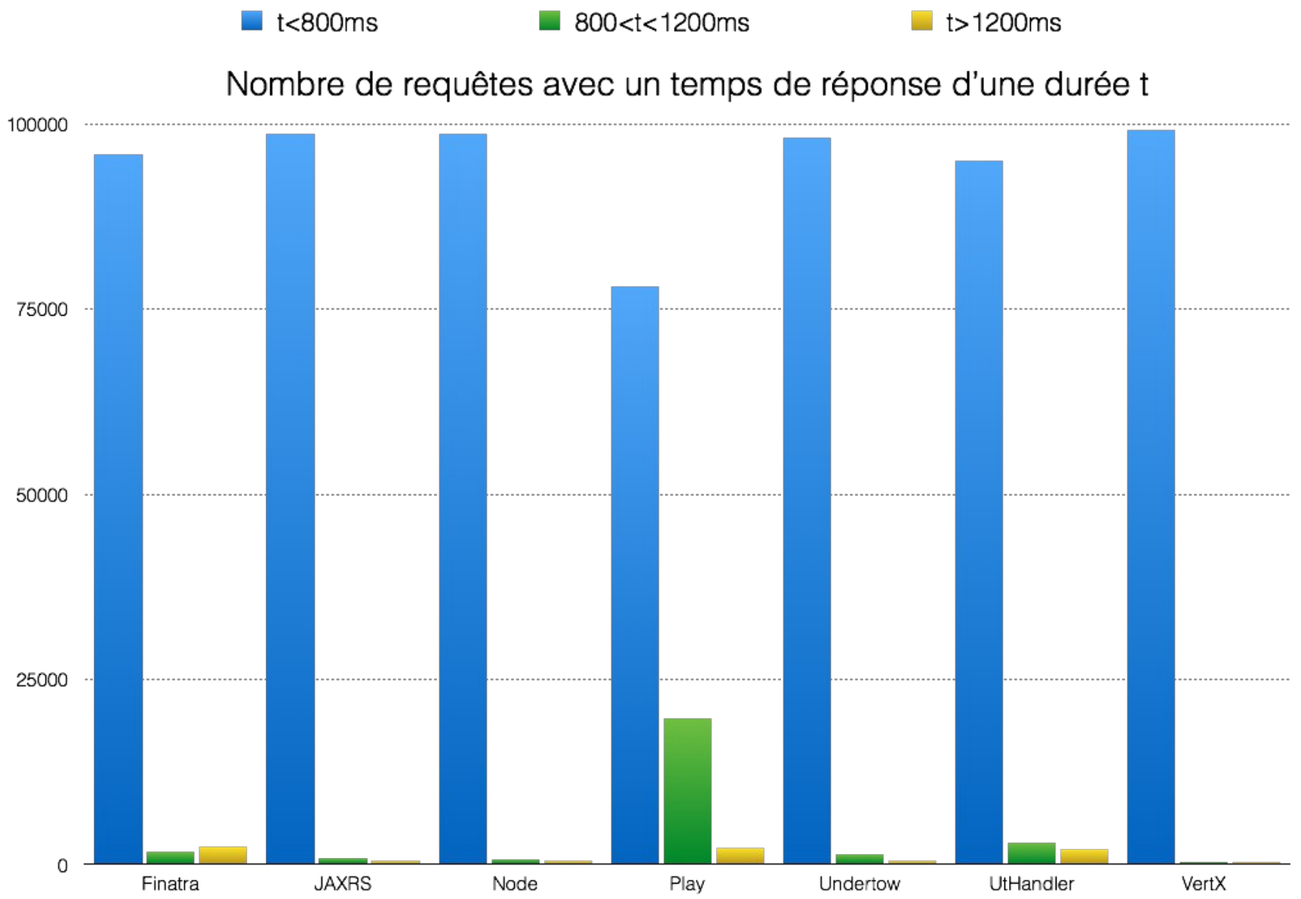
Code optimization

```
// avant

app.get("/movies", function(req, res) {
  res.send(movies);
});

// après

app.get("/movies", function(req, res) {
  res.sendfile("./db/movies.js", "utf8");
});
```

Then ...


Node
JAXRS
Play Scala
UnderTow
Uthandler
Vertx
Finatra
224,3
268,8
261,2
268,7
221,25
266,65
251,1
0
67,5
135
202,5
270

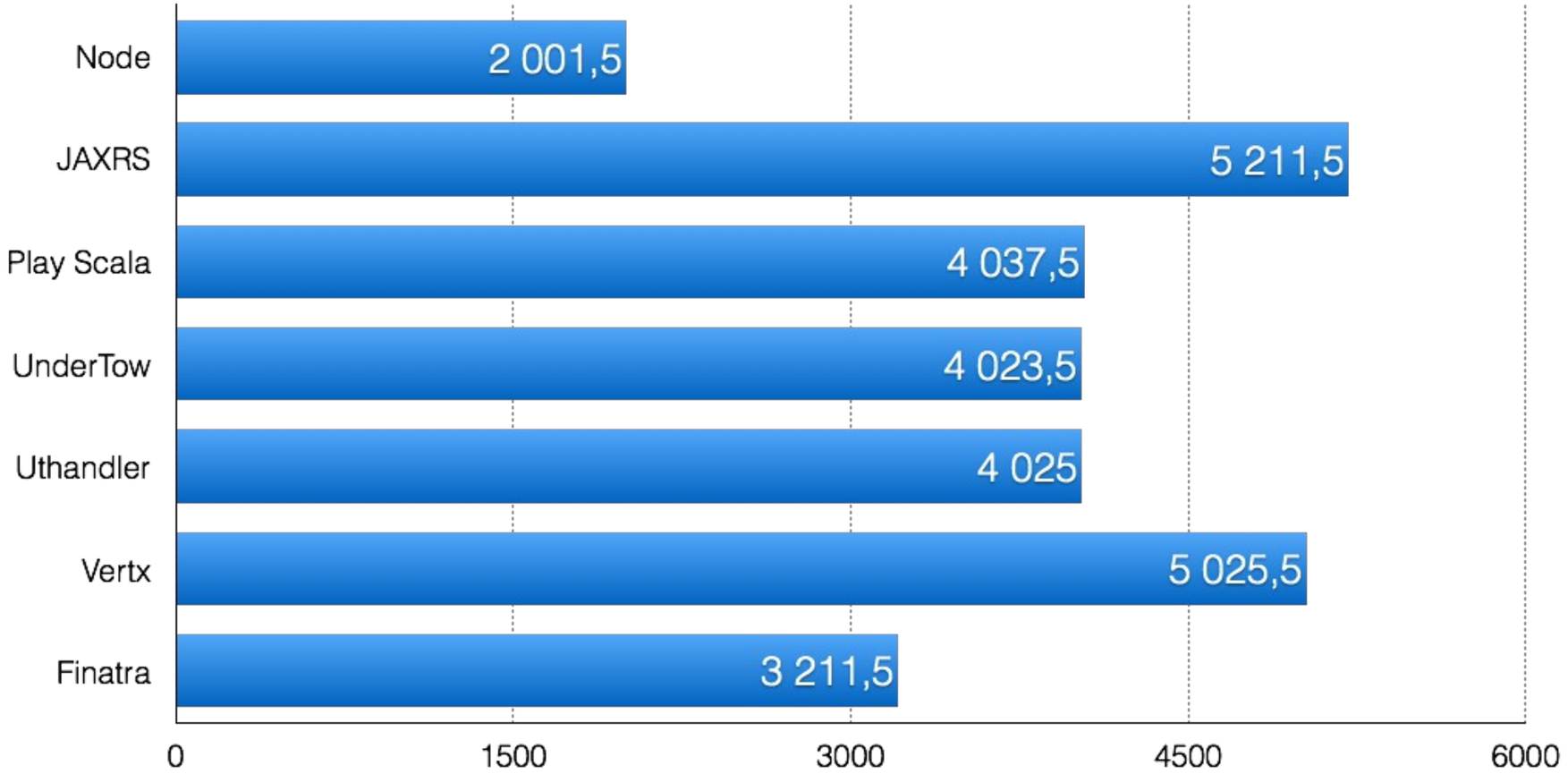
Then ...


t<800ms
800<t<1200ms
t>1200ms
Nombre de requêtes avec un temps de réponse d'une durée t
100000
75000
50000
25000
0
Finatra
JAXRS
Node
Play
Undertow
UtHandler
VertX

Vote scenario

- Light payload
- Much CPU bounded (search via regular expressions)

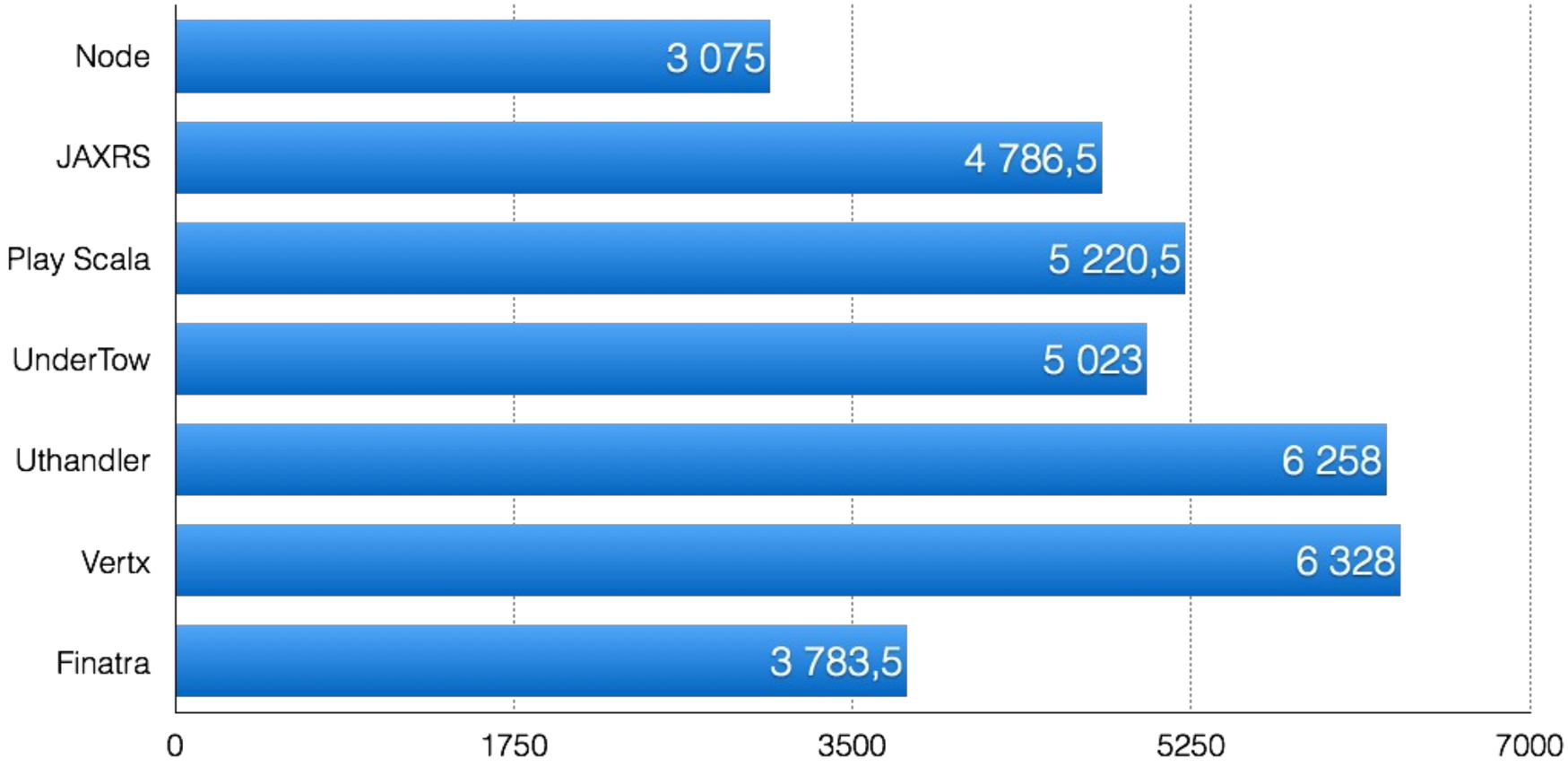
Nb Reqs/sec : Votes



Distance scenario

- Nul payload
- CPU bound : compute distance

Nb Reqs/sec : Distances




Summary

Conclusion

- Do performance testing, it is not that complex
- Developer is key
- Undestand the architecture
- Validate your architectual choice ASAP
- To be continued...


Q & A