How to get extreme database performance and development speed

Niklas Bjorkman
VP Technology, Starcounter

Could it be magic?

"Any sufficiently advanced technology is indistinguishable from magic."

Arthur C. Clarke

Today's topics

Utilize server side performance of today

Persistent server controlled app

How it is possible

Performance on all levels

Performance – where?

Raw database performance	The enabler Hundreds of thousands of TPS per CPU core Easy to scale up performance
Web server performance	The hub Millions of http RPS DB transaction time included
Development performance	We like less lines of code Shorter time to market Less maintenance costs

Super fast

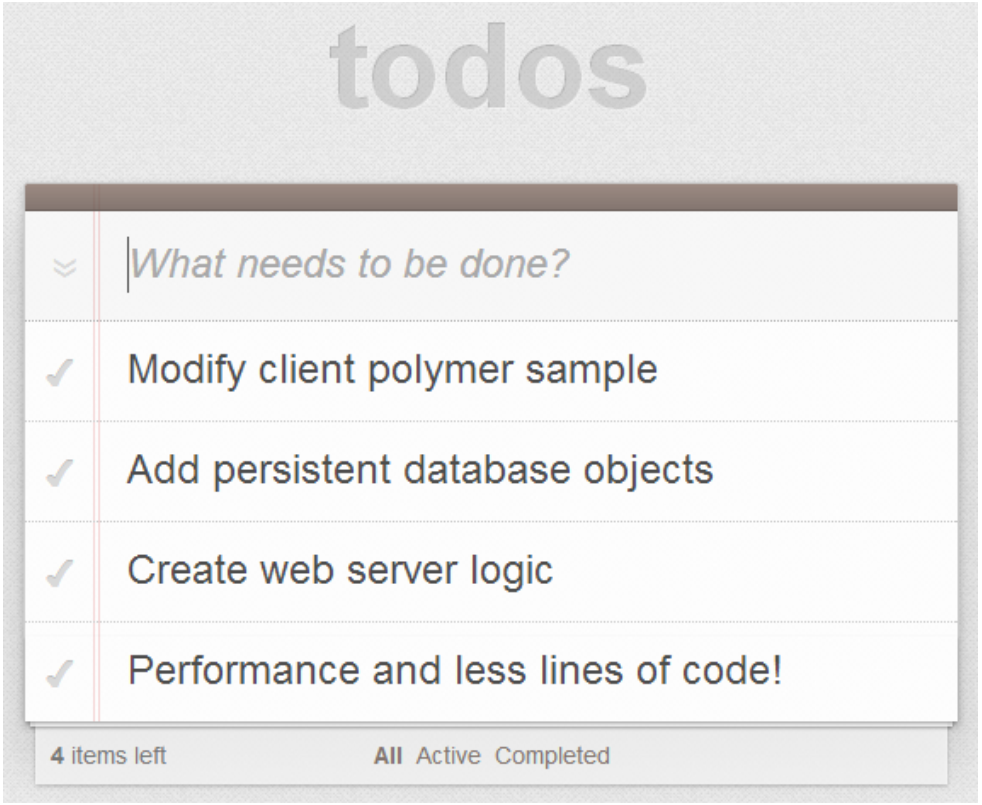
What to do with it?

“Todo” sample

Rewrite to server side controlled

Todo MVC (Web Components)

Web component implementation using Polymer
www.todomvc.com/architecture-examples/polymer/index.html

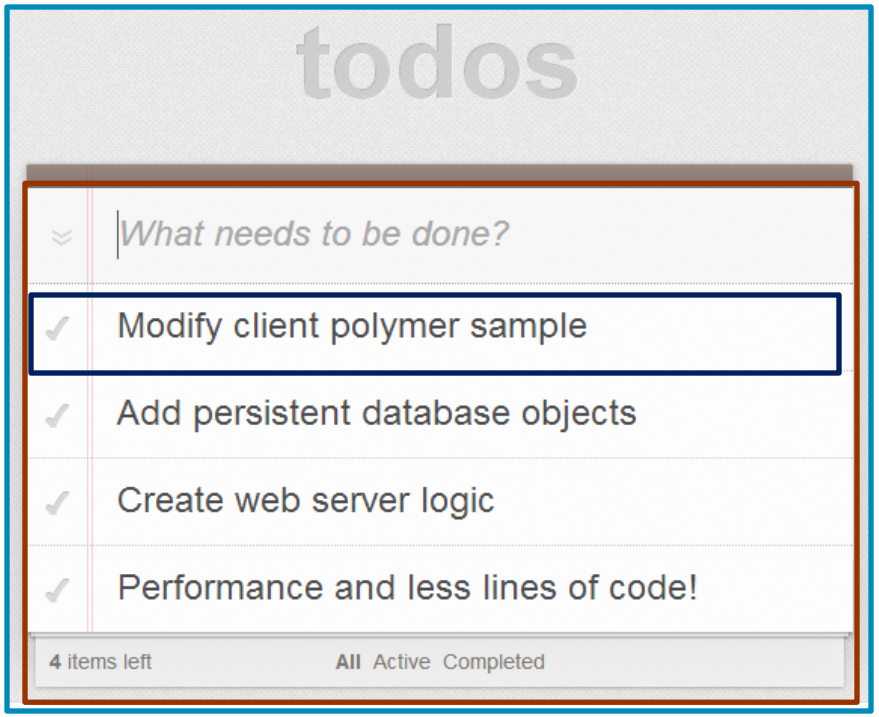


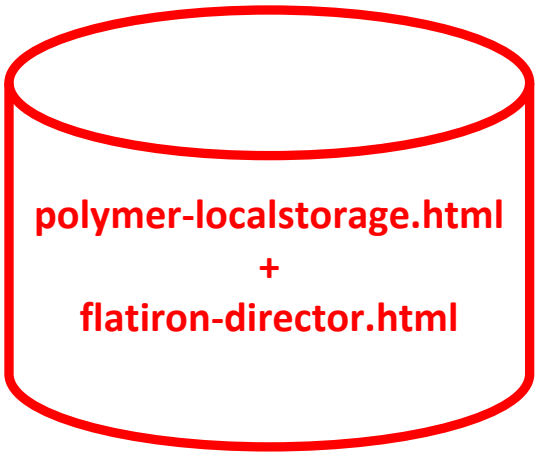
Todo MVC file setup

master.html

td-todos.html

td-item.html





Server side “todo”

All the same features - fully server controlled

Nothing created, stored or updated in local client storage

Persistent

Tamper proof

Less lines of code

Simplified client implementation

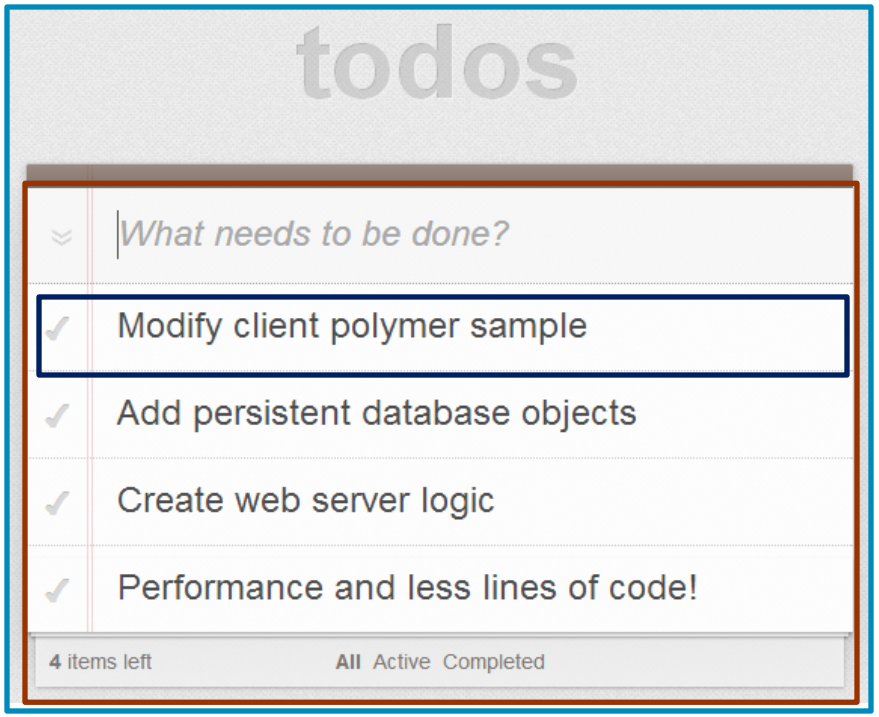
Use JSON Patch (RFC 6902)

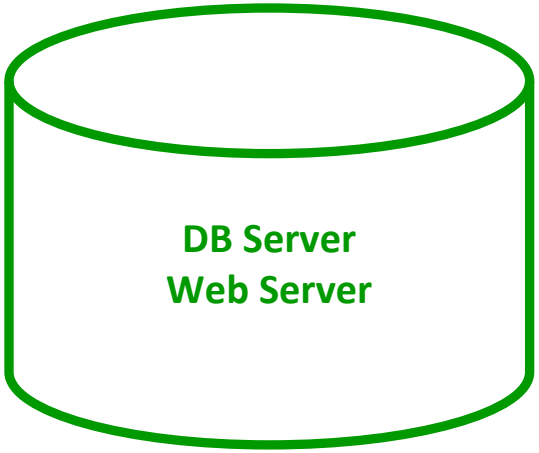
Todo MVC server version setup

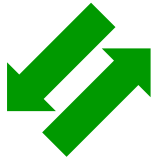
master.html

td-todos.html

td-item.html







“Todo” polymer client

Remove and modify code

Change 1: master.html

```
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <title>Polymer • TodoMVC</title>
  <link rel="stylesheet" href="/app/app.css">
  <link rel="import" href="/lib-elements/polymer-localstorage.html">
  <!-- <link rel="import" href="elements/td-model.html">-->
  <link rel="import" href="/elements/td-todos.html">
  <script src="/bower_components/polymer/polymer.min.js"></script>
  <script src="/json-patch-duplex.js"></script>
  <script src="/puppet.js"></script>
</head>
<body>
  <header>
      <h1>todos</h1>
  </header>
  <polymer-localstorage id="storage" name="todos-polymer"></polymer-localstorage>
  <!--<td-model id="model" storageId="storage"></td-model>-->
  <td-todos storageId="storage"></td-todos>
  <footer id="info">
      <p>Double-click to edit a todo</p>
      <p>Created by <a href="http://www.polymer-project.org">The Polymer Authors</a></p>
      <p>Part of <a href="http://todomvc.com">TodoMVC</a></p>
  </footer>
</body>
```

Change 2: polymer-localstorage.html

```
Polymer('polymer-localstorage', {
    …
    value:{},
    load: function() {
        new Puppet(null, null, this.value);
        // var s = window.localStorage.getItem(this.name);
        // if (s && !this.useRaw) {
        // this.value = JSON.parse(s);
        // } else {
        // this.value = s;
    },
    save: function() {
        // window.localStorage.setItem(this.name,
        // this.useRaw ? this.value:JSON.stringify(this.value));
    }
```

Change 3: td-model.html

```
<polymer-element name="td-model" attributes="filter items storageId">
        <script>
                Polymer('td-model', {
                        filtered: null,
                        completedCount: 0,
                        activeCount: 0,
                        allCompleted: false,
                        ready: function () {
                                this.asyncMethod(function () {
                                        this.items = this.items || [];
                                });
                        },
                        filterChanged: function () {
                                this.filterItems();
                        },
                        itemsChanged: function () {
                                this.completedCount =
                                        this.items.filter(this.filters.completed).length;
                                this.activeCount = this.items.length - this.completedCount;
                                this.allCompleted = this.completedCount && !this.activeCount;
                                this.filterItems();
                                if (this.storage) {
                                        this.storage.value = this.items;
                                        this.storage.save();
                                }
                        },
                        storageIdChanged: function () {
                                this.storage = document.querySelector('#' + this.storageId);
```

Delete file

```
.items;
                        title = String(title).trim();
                        if (title) {
                                var item = {
                                        title: title,
                                        completed: false
                                };
                                this.items.push(item);
                                this.itemsChanged();
                        }
                },
                destroyItem: function (item) {
                        var i = this.items.indexOf(item);
                        if (i >= 0) {
                                this.items.splice(i, 1);
                        }
                        this.itemsChanged();
                },
                clearItems: function () {
                        this.items = this.items.filter(this.filters.active);
                },
                setItemsCompleted: function (completed) {
                        this.items.forEach(function (item) {
                                item.completed = completed;
                        });
                        this.itemsChanged();
                },
                filters: {
                        active: function (item) {
                                return !item.completed;
                        },
                        completed: function (item) {
                                return item.completed;
                        }
                }
        });
        </script>
</polymer-element>
```

Change 4: flatiron-director.html

```
<script src="../bower_components/director/build/director.min.js"></script>
  <polymer-element name="flatiron-director" attributes="route">
      <script>
      (function() {
        var private_router;
        Polymer('flatiron-director', {
          ready: function() {
            this.router.on(/(\w*)/, function(route) {
                this.route = route;
            }.bind(this));
            this.asyncMethod(function() {
              var initialRoute = this.router.getRoute(0);
              this.route = initialRoute || '';
              // flush to here to render the initial route synchronously.
```

Delete file

```
            private_router = new Router();
            private_router.init();
            }
            return private_router;
      },
      routeChanged: function() {
        this.fire('route', this.route);
      }
        });
      })();
  </script>
</polymer-element>
```

Change 5: td-item.html

```
...
<link rel="stylesheet" href="td-item.css">
  <div class="view {{completed: item.completed$; editing: editing}}" ...>
            <input type="checkbox" class="toggle" checked="{{item.completed$}}" ...>
            <label>{{item.title$}}</label>
            <button class="destroy" on-click="destroyAction"></button>
       </div>
...
  <script>
       (function() {
                 ...
                 commitAction: function() {
                      this.title = this.$.edit.value;
                      if (this.editing) {
                           this.editing = false;
                           this.item.title$ = this.title.trim();
                           if (this.item.title$ === '') {
                                this.destroyAction();
                           }
                 ...
  </script>
</polymer-element>
```

Change 6: td-todos.html (1/2)

```
link rel="import" href="../lib-elements/polymer-selector.html
<!-- <link rel="import" href="../lib-elements/flatiron-director.html">-->
<link rel="import" href="td-input.html">
<link rel="import" href="td-item.html">
<polymer-element name="td-todos" attributes="route storageId">
    <template>
        <link rel="stylesheet" href="
        <!--<flatiron-director route="{{route}}"></flatiron-director>-->
        <section id="todoapp">
            <input id="toggle-all" ... checked="{{model.allCompleted$}}">
            ...
            <template repeat="{{model.items}}">
                <li is="td-item" item="{{}}"></li>
            </template>
        </section>
        <footer id="footer" hidden?="{{model.activeCount + model.completedCount == 0}}">
        ...
        <polymer-selector id="filters" selected="{{model.filterOption}}">
            <li name="all">
                <a href="{{model.taskListUrl}}" on-click="fireAction">All</a>
            </li>
            <li name="active">
                <a href="{{model.taskListUrl}}/active" on-click="fireAction">Active</a>
            </li>
            <li name="completed">
                <a href="{{model.taskListUrl}}/completed" on-click="fireAction">Completed</a>...
```

Change 7: td-todos.html (2/2)

```
...
<script>
    (function() {
        var ENTER_KEY = 13;
        var ESC_KEY = 27;
        Polymer('td-todos', {
            storageIdChanged: function() {
                this.model = document.querySelector('#' + this.storageId).value;
            },
            ...
            addTodoAction: function() {
                this.model.newItemTitle$ = this.$['new-todo'].value;
                this.fire('blur');
                Platform.flush();
                this.$['new-todo'].value = '';
            },
            ...
            destroyItemAction: function(e, detail) {
                detail.remove$ = null;
                // var i = this.model.items.indexOf(detail);
                // if (i >= 0) {
                //    this.model.items.splice(i, 1);
                //}
                //this.model.destroyItem(detail);
            }, ...
```

“Todo” sample server

Implement the server side

Here we did some live programming!
Please download the server side project from github.

All classes and methods have helpful comments.

Happy coding!

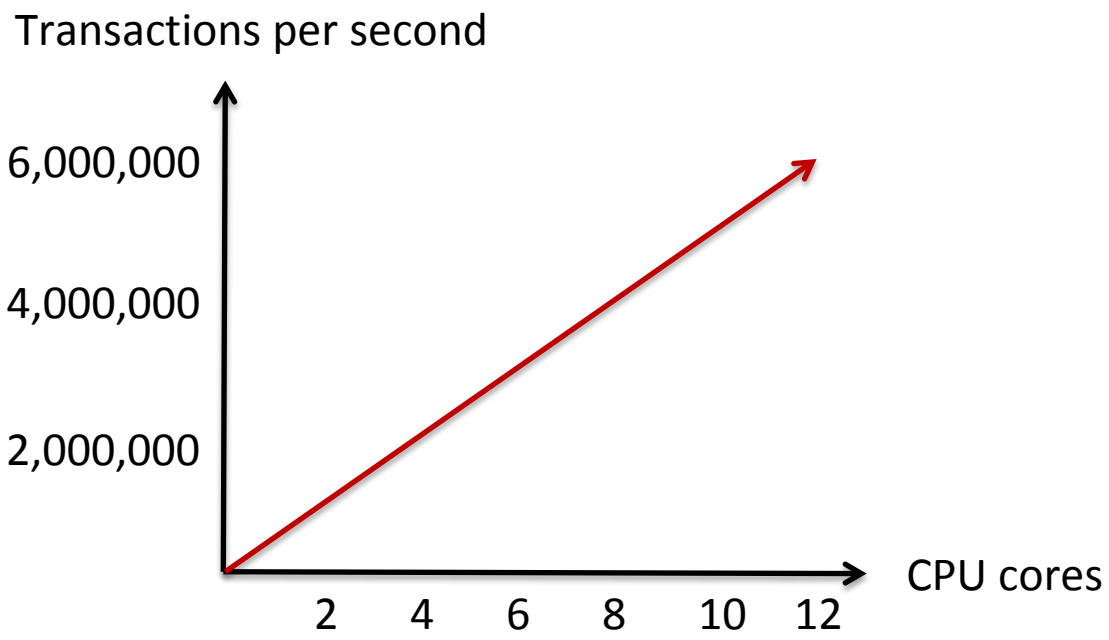
https://github.com/Starcounter/TodoDemo

How is this possible?

(Super fast NewSQL database)

Application performance

A fast NewSQL database would scale read transactions linearly over the number of cores



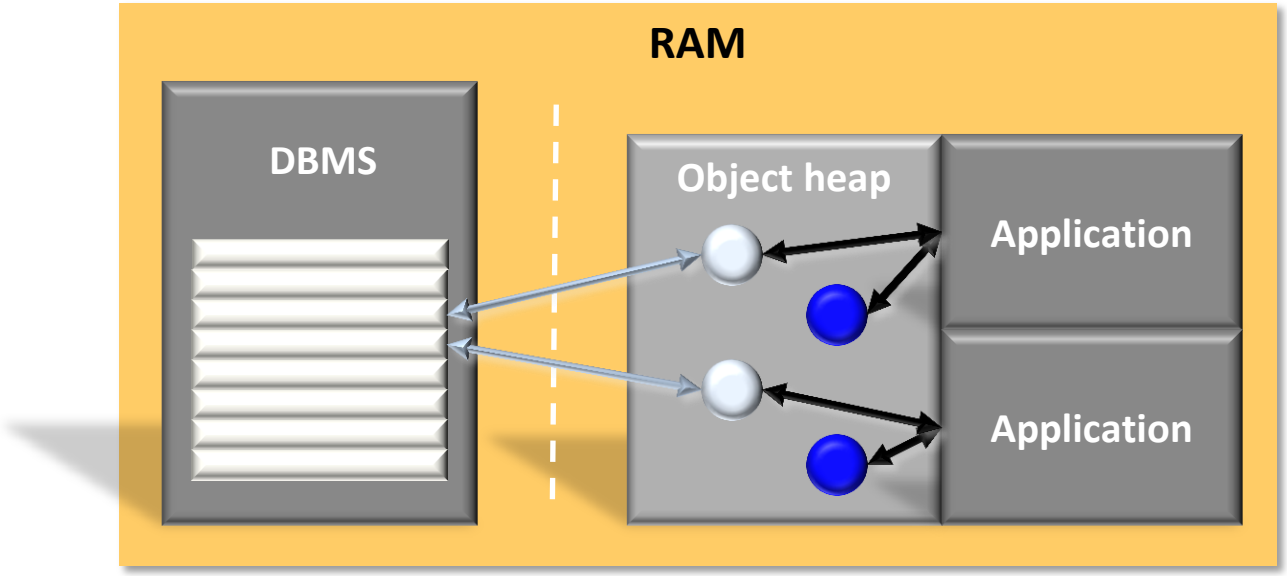
Walking the extra mile

First move everything into RAM

Can we use RAM even further?

Traditionally DB objects and App objects are in different set of RAM

Let the application and database share the object heap



New possibilities

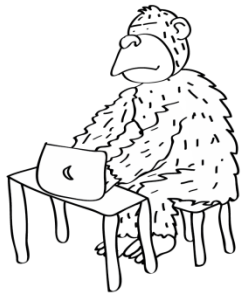
Use classes instead of DB schema

No need for ORMs, faster and less lines of code

POCO objects are your database

Query (SQL) support directly on your POCO

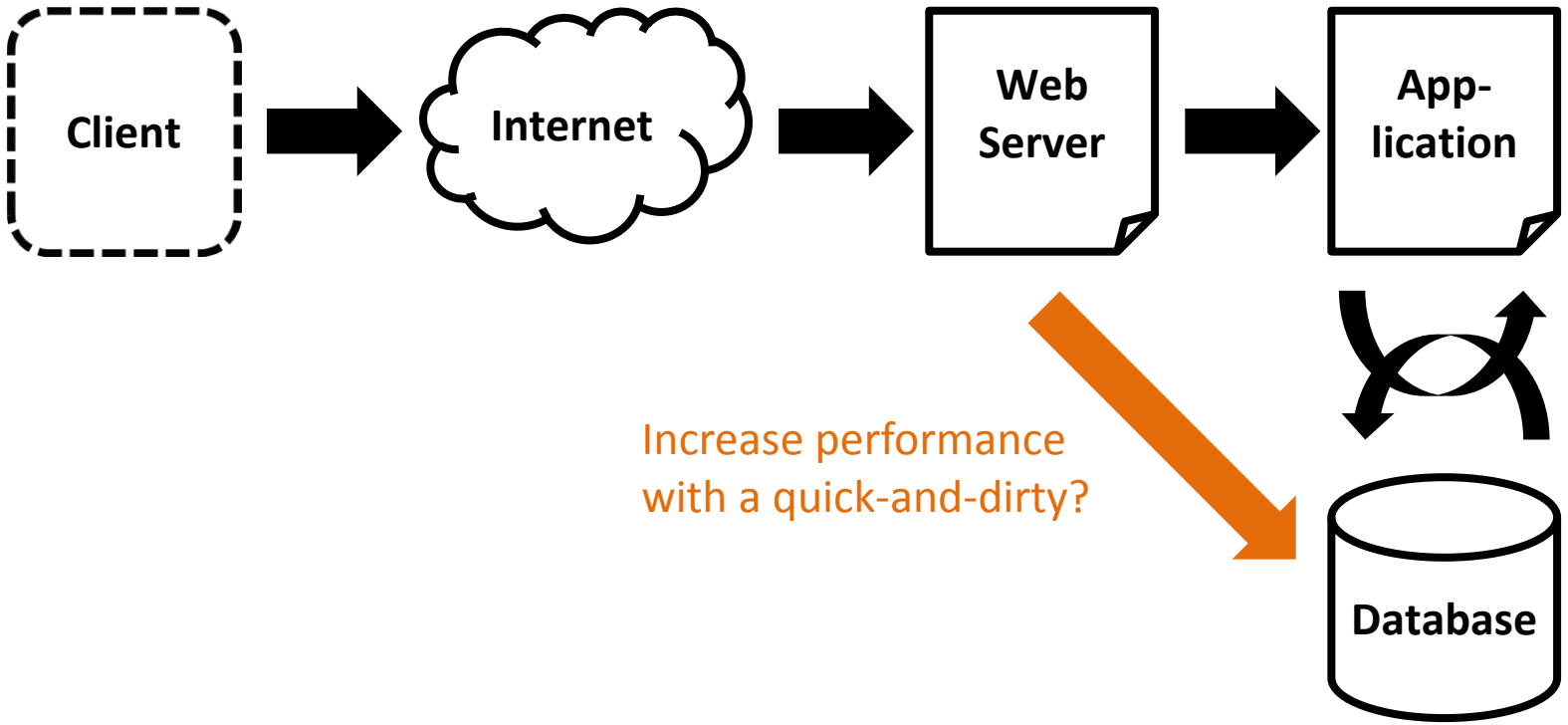
Make life easier for developers and DBAs



Make App development super easy

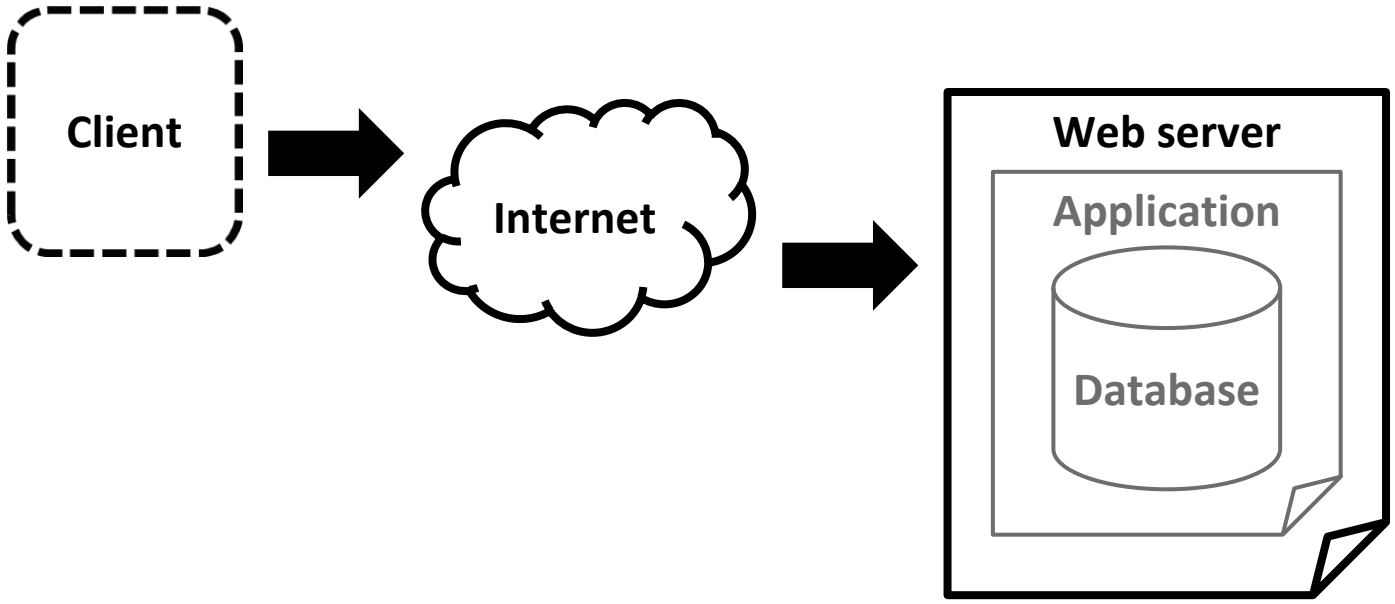
Traditional communication server

Traditional setup for a web based application

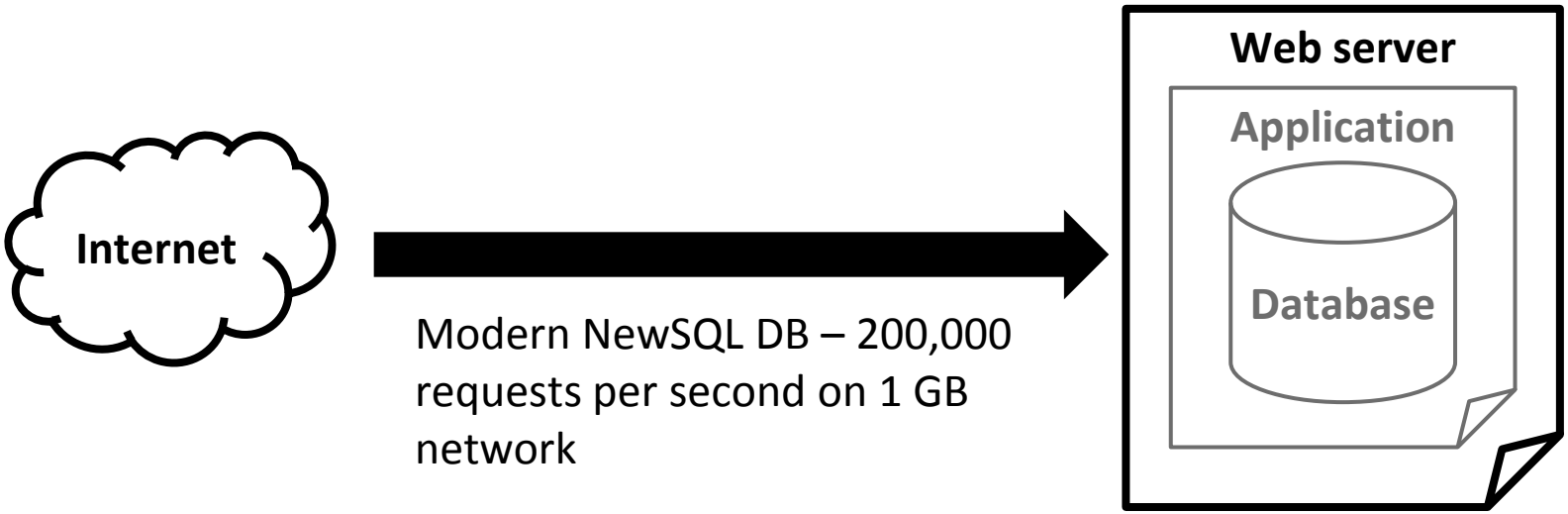


Increase performance

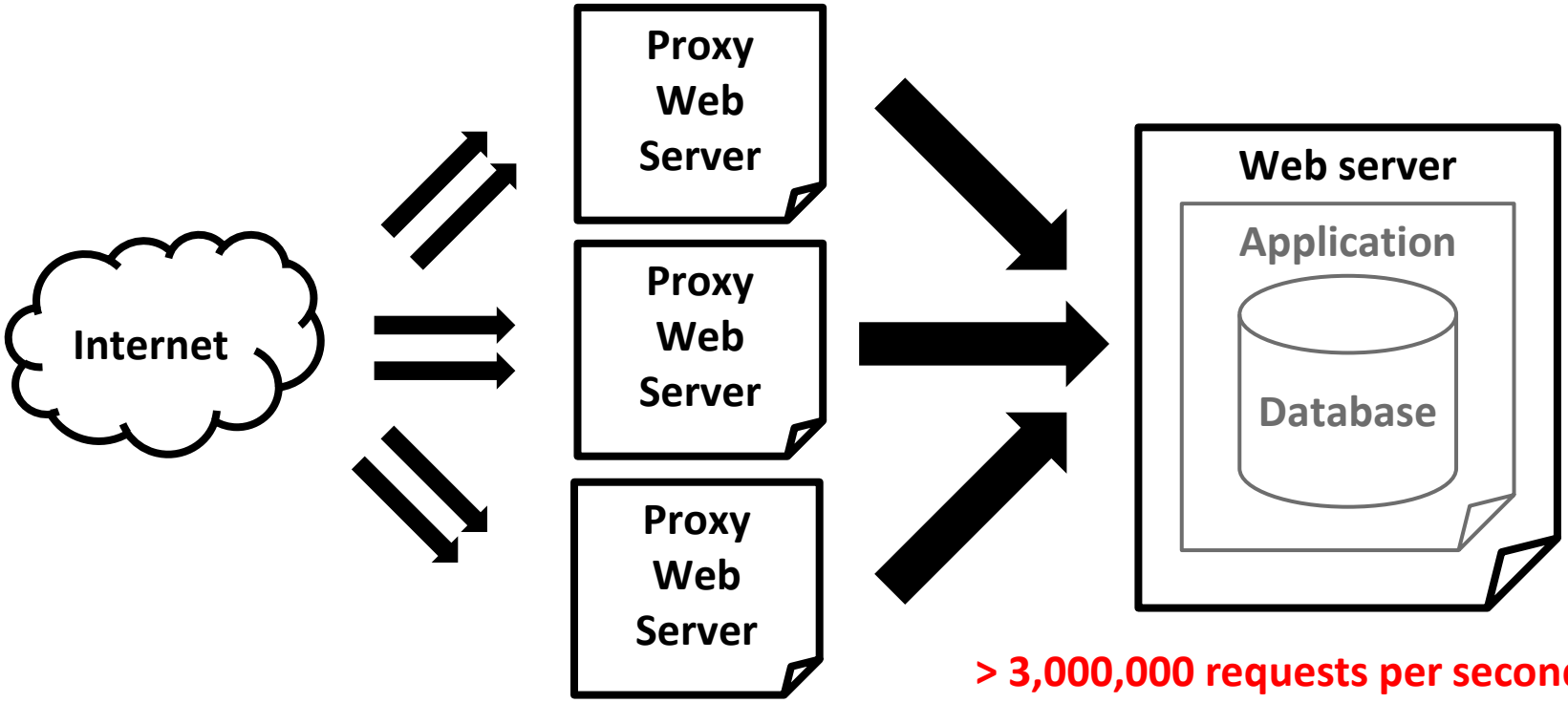
We could let the web server, application and database share the data



Network and OS limitations



Network/OS – solution



> 3,000,000 requests per second

What would ABBA say?

Money, money, money

Save money on

- Hardware – no large data centers needed
- Shorter time to market
- Less maintenance
- Fewer developers
- Faster to learn
- Less DBA costs

Summary

Super fast server side enables new opportunities

Easy server controlled apps development

Less lines of code

Easy to develop

Reduce costs throughout application lifetime!

Magic

References

Original Todo Client
http://todomvc.com/architecture-examples/polymer/index.html

Modified Todo Client
https://github.com/PuppetJs/PuppeteerJs/tree/master/examples/todomvc

Server side Todo project
https://github.com/Starcounter/TodoDemo

JSON Patch
https://github.com/Starcounter-Jack/JSON-Patch

Puppet JS
https://github.com/PuppetJs/PuppetJs

Thanks for listening!

Niklas Bjorkman
VP Technology, Starcounter