Orchestrating Robot Swarms with Java

ocado TECHNOLOGY


PICCOLO'S DIP IT!
Indian Rice
Halloumi with mint
UMAMI PASTE
Colman's
kallø ORGANIC VEGETABLE
the Coconut COLLABORATIVE
Uncle Ben's MEXICAN CHILLI PERFECT FOR LUNCH
MAILLE
Jamie Oliver
Ostara
LOYD GROSSMAN Check it!
zest
OLD EL PASO Hot
ocado 6 Medium free range
PG tips
HELLMANN'S
DOLMIO
gluten free oatcakes
TATE & LYLE
ORGANIC SPICED DAMSON
McVitie's Jaffa Cakes
NIVEA Crème
Doritos
CARDAMOM PODS
Balsamic Onion & Cheddar Quiche
Colgate
beond
ORGANIC APPLE CINNAMON BAR
the original organic baby food co.
COVENT GARDEN REMEDIES
Tomato salsa dip
Waitrose Zesty, buttery lemon curd yogurt
Gü
Cadbury Dairy Milk Chunks
Knorr GRAVY POT CHICKEN GRAVY
PRONTO STIR-IN PASTA SAUCE
VINTAGE ORGANIC CHEDDAR


ocado
30% OFF
your first grocery shop
+ FREE deliveries for a year
2 FOR £5
EDWARDS

ocado TECHNOLOGY

Matthew Cornford

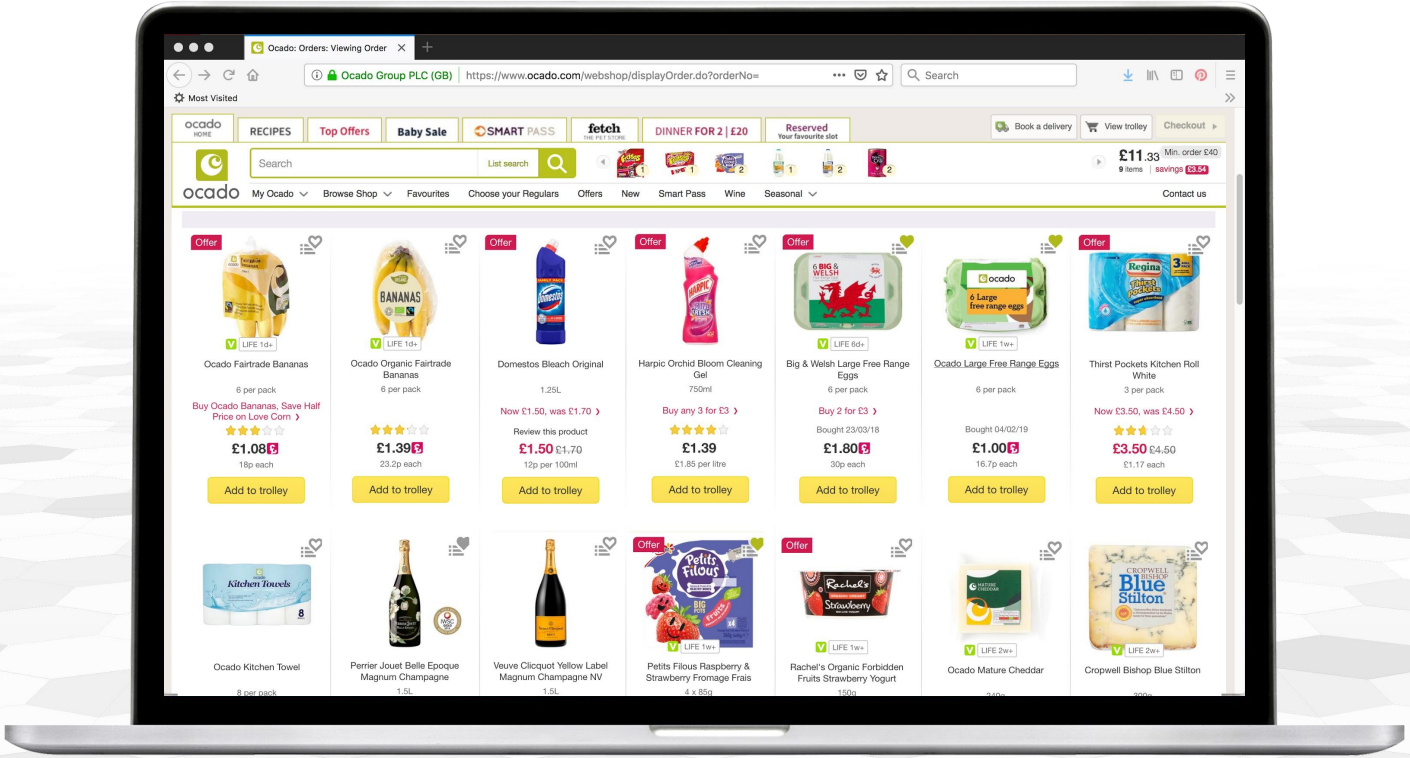
Technology Lead
Ocado Technology

ocado TECHNOLOGY

@OcadoTechnology

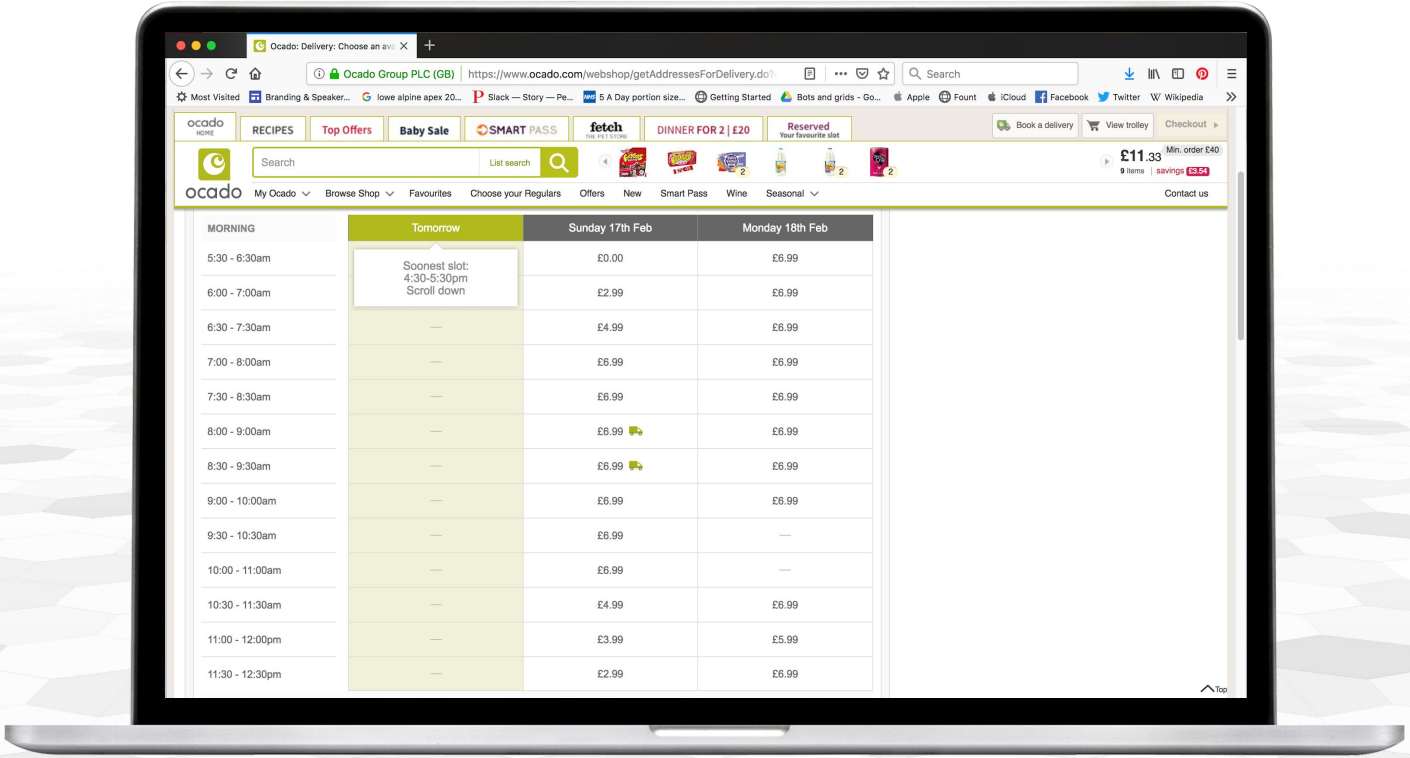
@bofalot

The Problem to Solve



/ OcadoTechnology

The Problem to Solve

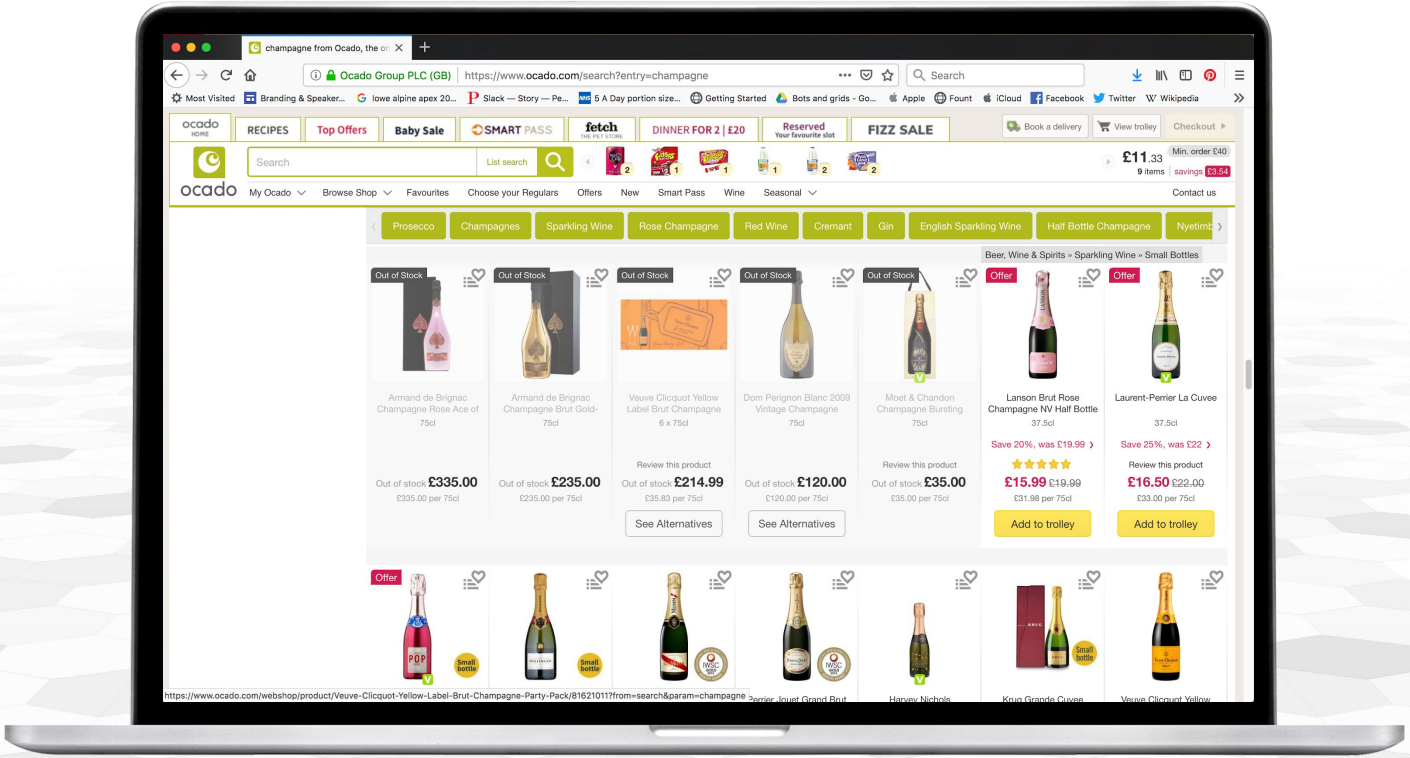


/ OcadoTechnology


Hamburger Press
Hamburger Press
Sunnyside
8 Plain Donuts
price drop
10 FOR $10
2.99
3.59
4.29


1
2
3
4
2
7
15
5
1
12
11
10
3
16
13
6
8
9
14

Customer Fulfilment Centres



/ OcadoTechnology


Aisle
7

Robot Stats

Grid the size of 3 football pitches

Up to 3000 robots at any one time

4m/s bot speed

5mm clearance

35kg load

Communicating 10 times a second over an unlicensed part of the 4G spectrum

System Overview

Why Java?

Simulation

Determinism

Low Latency Communication

ocado TECHNOLOGY

System Overview

Why Java?

Simulation

Determinism

Low Latency Communication

ocado TECHNOLOGY

System Overview

Why Java?

Simulation

Determinism

Low Latency Communication

ocado TECHNOLOGY

System Overview

Why Java?

Simulation

Determinism

Low Latency Communication

ocado TECHNOLOGY

System Overview

Why Java?

Simulation

Determinism

Low Latency Communication

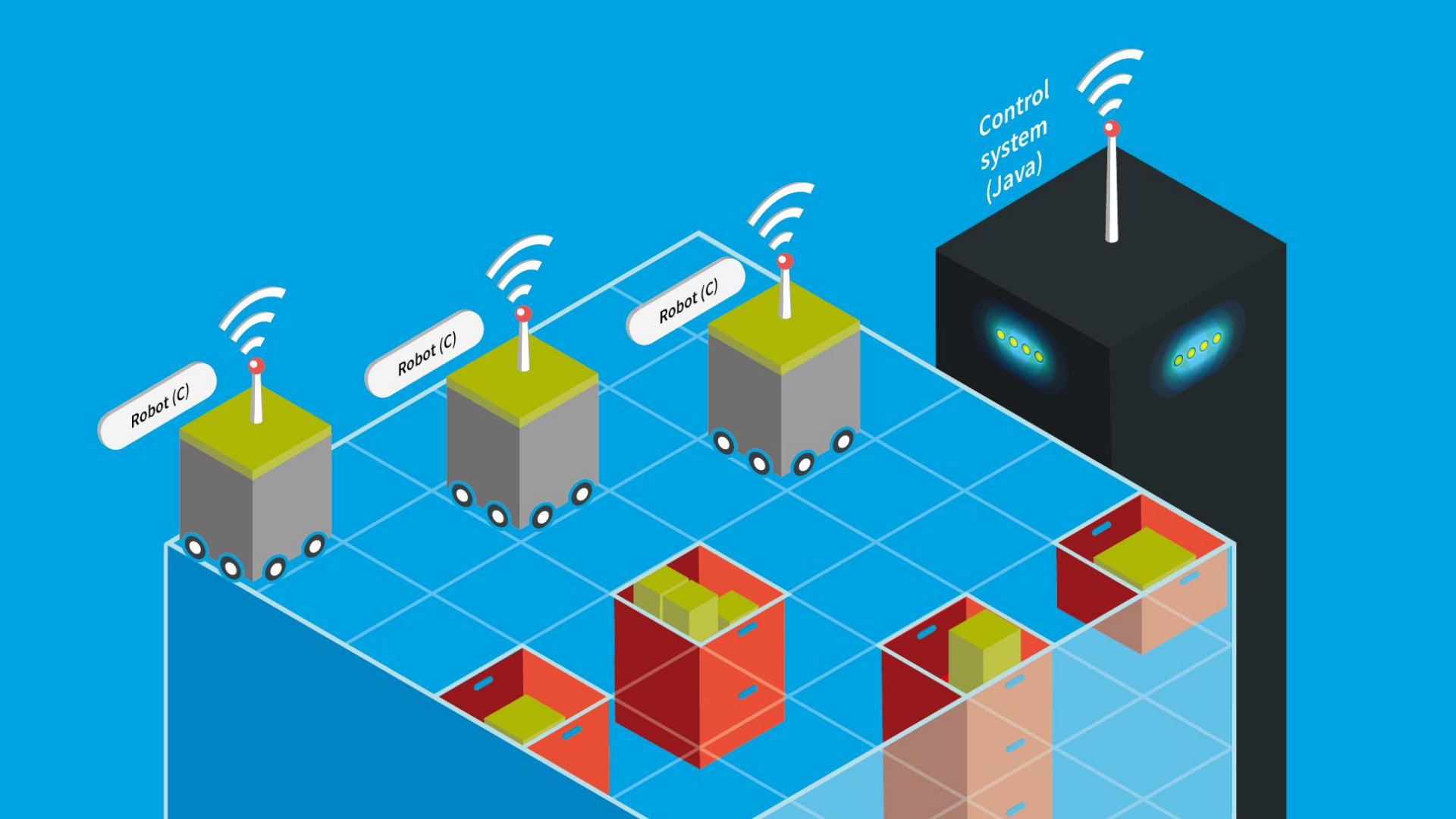
System Overview

Why Java?

Simulation

Determinism

Low Latency Communication


Control system (Java)
Robot (C)
Robot (C)
Robot (C)

System Overview

ocado TECHNOLOGY

> **“Premature optimization is the root of all evil”**
>
> *Donald Knuth*

ocado TECHNOLOGY

System Overview

Why Java?

Simulation

Determinism

Low Latency Communication

Why Java?

Speed of Development

vs

Performance


ocado
TECHNOLOGY


AdoptOpenJDK

ocado TECHNOLOGY

System Overview

Why Java?

Simulation

Determinism

Low Latency Communication

Simulation

A simulation is an **approximate imitation** of the operation of a **process or system**; the act of simulating first requires a **model is developed**

Simulation

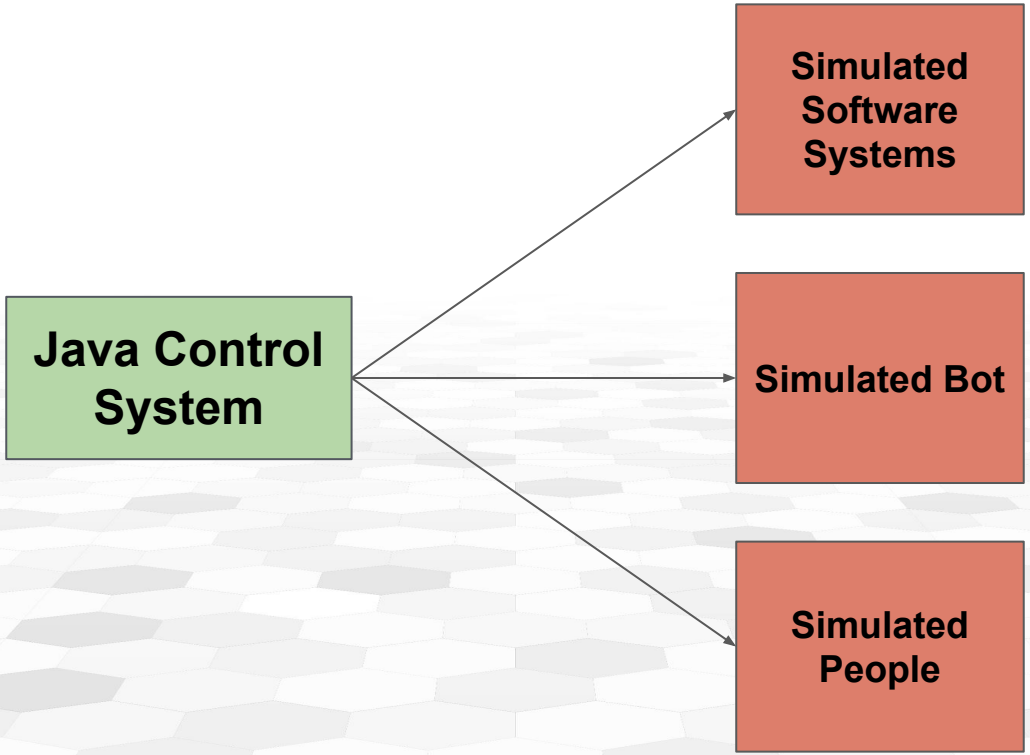
ocado TECHNOLOGY

Microsoft Flight Simulator

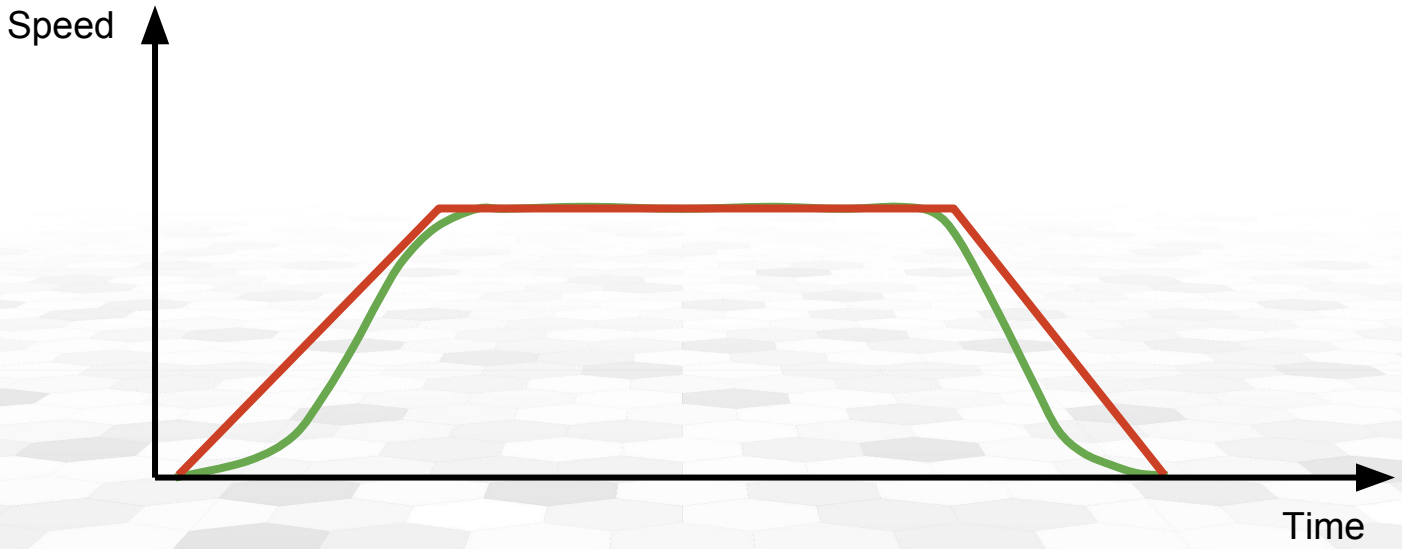
The SIMS™

/ OcadoTechnology

What Do We Simulate?



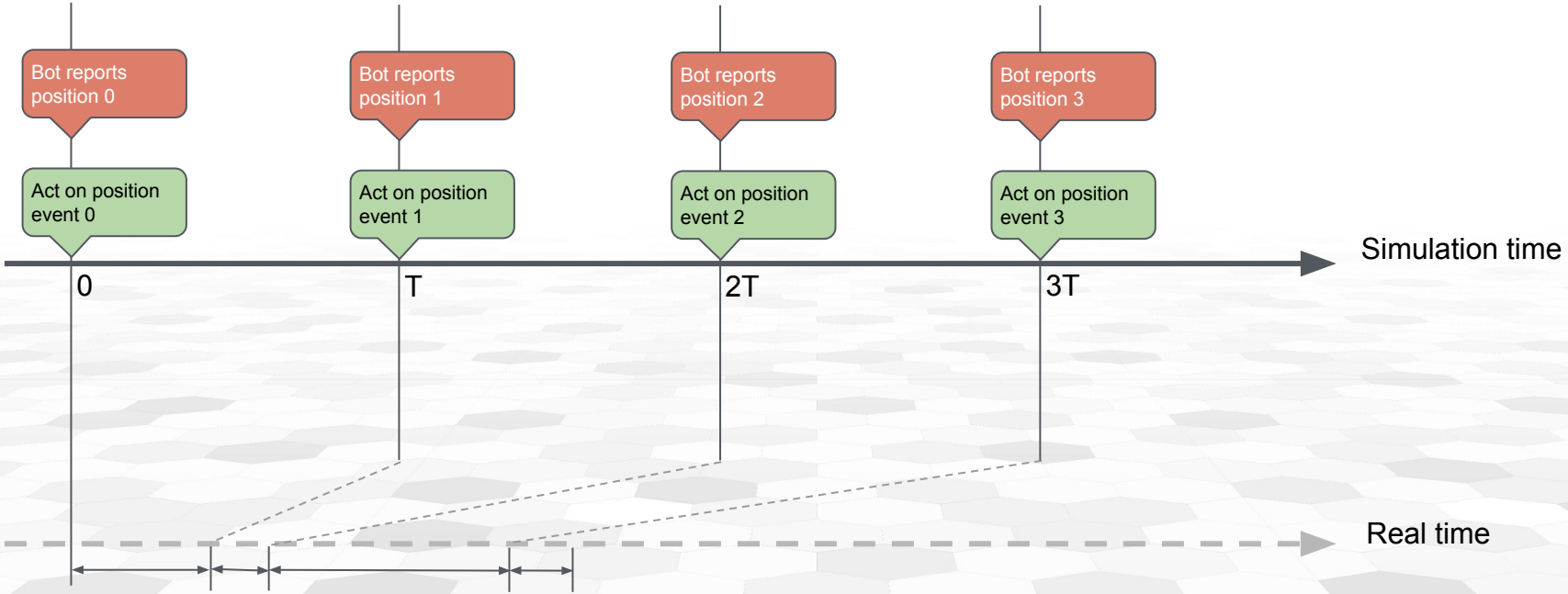
Simulation Example



Discrete Event Simulation

A discrete-event simulation (DES) models the operation of a system as a **discrete sequence of events in time**. Each event occurs at a particular **instant in time** and marks a **change of state** in the system. Between consecutive events, no change in the system is assumed to occur; thus the simulation can directly **jump in time** from one event next.

Discrete Event Simulation



ocado TECHNOLOGY

System Overview

Why Java?

Simulation

Determinism

Low Latency Communication

Determinism

- Real-time systems are not deterministic
- We want determinism in our discrete event simulations
- We test for it in our CI pipeline
- Three areas:
  - Time
  - Scheduling
  - Iteration

Determinism - Time

```
@FunctionalInterface
public interface TimeProvider {
    long getTime();
}
```

Determinism - Time

```
public class AdjustableTimeProvider implements TimeProvider {
    private long currentTime;

    @Override
    public long getTime() {
        return this.currentTime;
    }

    public void setTime(long time) {
        this.currentTime = time;
    }
}
```

Determinism - Time

```
public class SystemTimeProvider implements TimeProvider {
    @Override
    public long getTime() {
        return System.currentTimeMillis();
    }
}
```

Determinism - Scheduling

```
public interface Event {
    long getTime();
    void run();
    void cancel();
}

public interface EventScheduler {
    Event doNow(Runnable r);
    Event doAt(long time, Runnable r)
}
```

Determinism - Scheduling

```
public class DiscreteEventScheduler implements EventScheduler {
    private final AdjustableTimeProvider timeProvider;
    private final EventQueue queue;

    private void executeEvents() {
        Event nextEvent = queue.getNextEvent();
        while (nextEvent != null) {
            timeProvider.setTime(nextEvent.getTime());
            nextEvent.run();
            nextEvent = queue.getNextEvent();
        }
    }
}
```

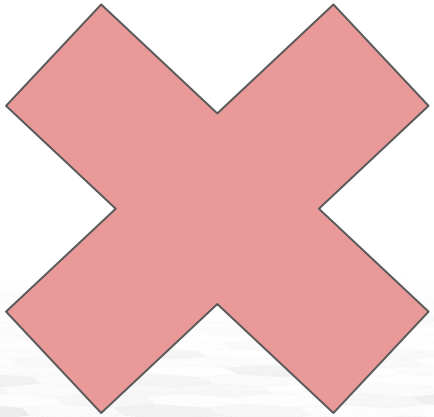
Determinism - Scheduling

```
public class RealTimeEventScheduler implements EventScheduler {
    ...
}
```

Determinism - Iteration

```
private Set<String> mySet = Set.of("a", "b", "c");

for (String entry : mySet) {
    doStuff();
}
```

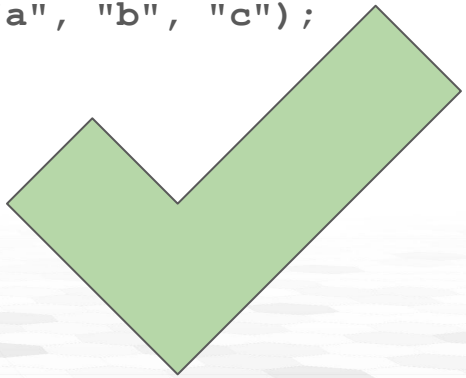


“The iteration order of set elements is unspecified and is subject to change.”

Determinism - Iteration

```
private ImmutableSet<String> mySet = ImmutableSet.of("a", "b", "c");

for (String entry : mySet) {
    doStuff();
}
```



“Except for sorted collections, order is preserved from construction time.”

ocado TECHNOLOGY

System Overview

Why Java?

Simulation

Determinism

Low Latency Communication

Event Scheduling

Requirements:

- To schedule events for specific times
- Individual events can’t be arbitrarily delayed
- The system can’t allow the events to arbitrarily backup

Module java.base

Package java.util

Class Timer

java.lang.Object
    java.util.Timer

```
public class Timer
```

Module java.base

Package java.util.concurrent

Class ScheduledThreadPoolExecutor

java.lang.Object
    java.util.concurrent.AbstractExecutorService
        java.util.concurrent.ThreadPoolExecutor
            java.util.concurrent.ScheduledThreadPoolExecutor

Event Scheduling - Busy Loop

```
public class RealTimeEventScheduler implements EventScheduler {
    private final TimeProvider timeProvider;
    private final EventQueue queue;

    private void executeEvents() {
        Event nextEvent = queue.getNextEvent();
        while (true) {
            if (nextEvent.getTime() <= timeProvider.getTime()) {
                nextEvent.run();
                nextEvent = queue.getNextEvent();
            }
        }
    }
}
```

Event Scheduling - Busy Loop

ocado TECHNOLOGY

Advantages

- Lower latency for individual events - from <5ms down to effectively 0
- Supports up to 3 times higher throughput of events

Disadvantages

- 100% CPU utilisation
- Can reduce clock speed due to the processor heating up

Memoization

We use two main flavours:

- “Standard”
- Object caching

Garbage Collection

GC is a primary source of application pauses

- Remove `java.util.Optional` from APIs that are heavily used
- Use `for` loops instead of the `Streams` API
- Use an `Array` backed data structure instead of `java.util.HashSet` or `java.util.LinkedList`
- Avoid primitive boxing, especially in unexpected places such as log lines, for example: `log.debug("{}", d)`

Garbage Collection Tips

- Enable GC logs by default!
- Understand the different collectors
- Don’t just take the latest and “greatest” garbage collector

G1GC vs ZGC

ocado TECHNOLOGY

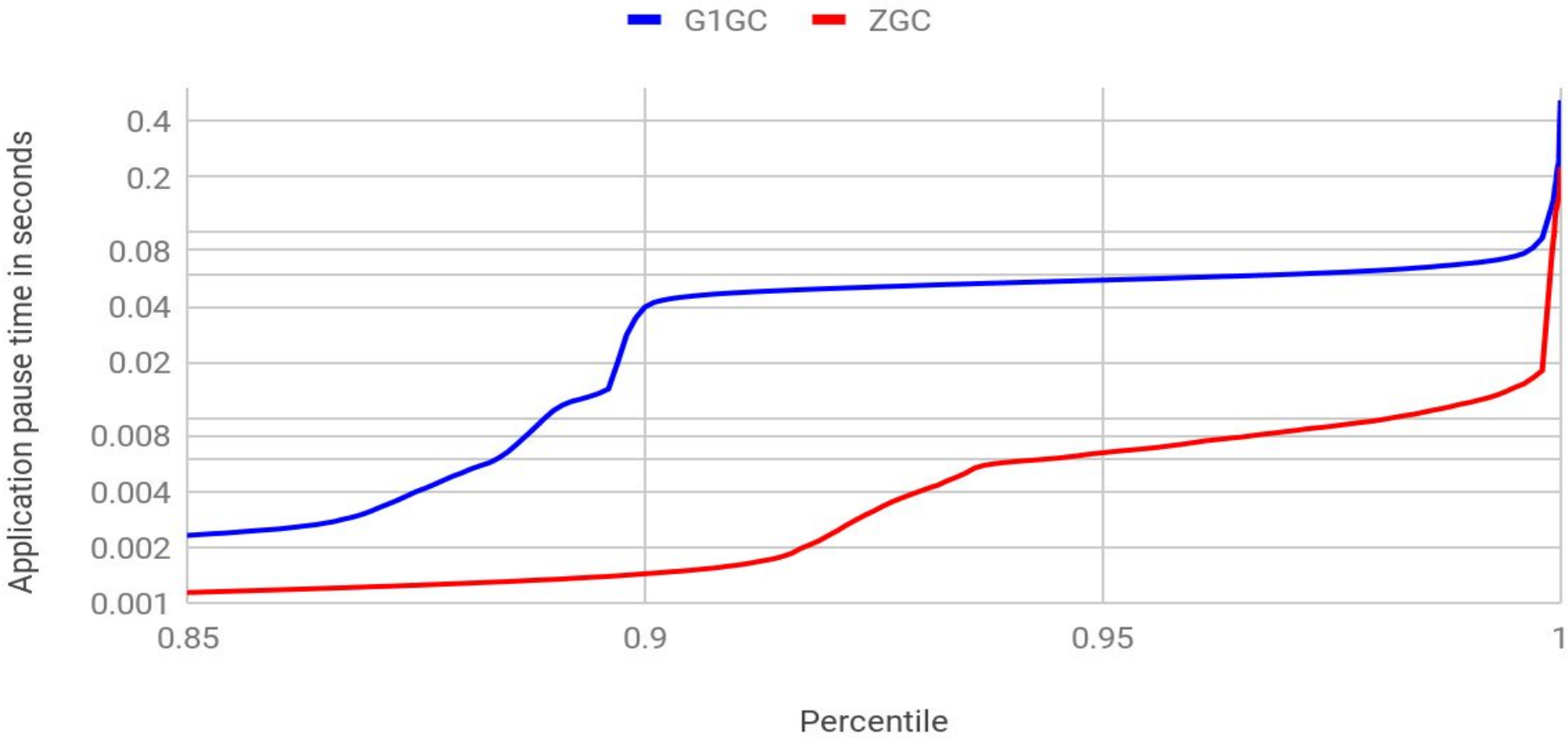
G1GC

- Used in production
- Can specify target pause time with `-XX:MaxGCPauseMillis=200`
- Trade-off with lower throughput and higher use of CPU by GC

ZGC

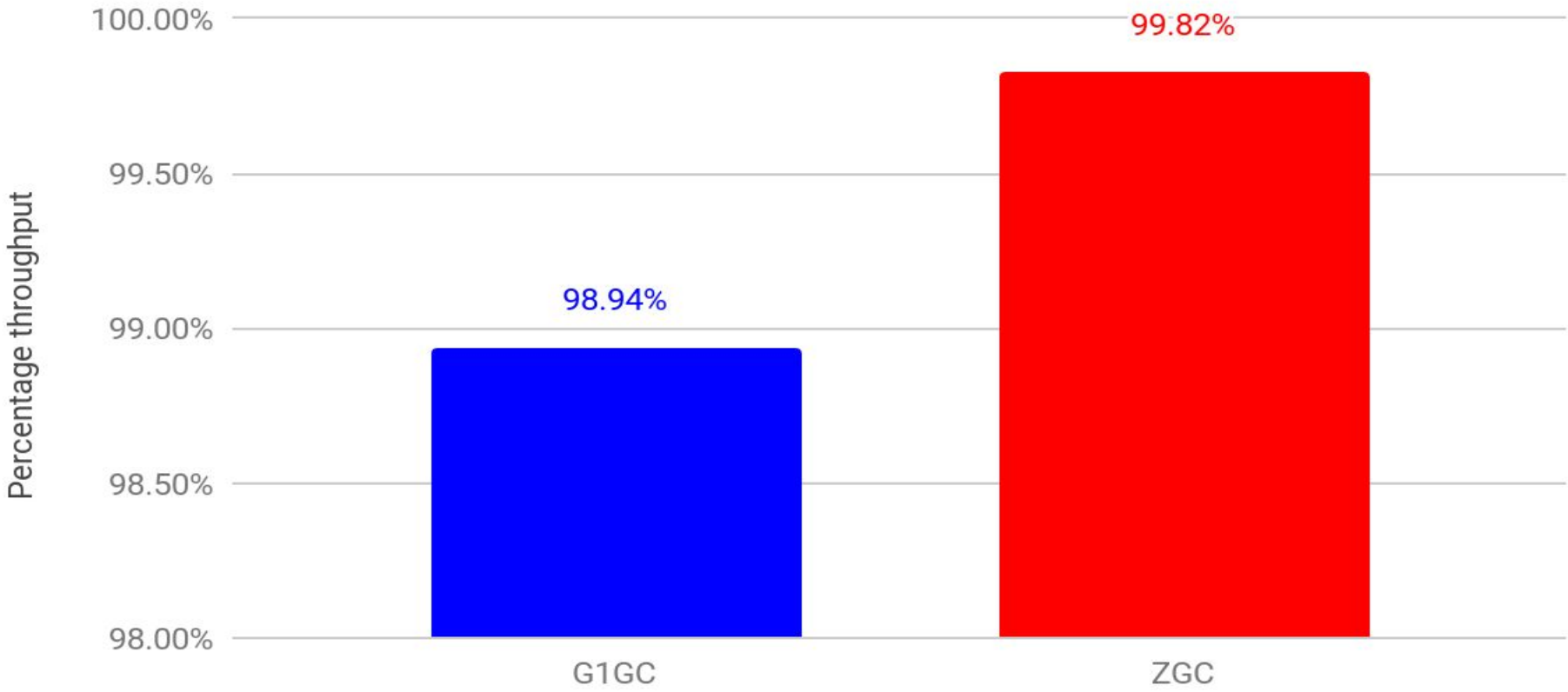
- New in Java 11
- Experimental
- Promises very low pause times

Pause Time Percentiles - G1GC vs ZGC



Application Throughput - G1GC vs ZGC

Total application runtime as a percentage of total test runtime



Summary

- Grocery is a difficult retail sector to run online profitably
- We use Java within our Customer Fulfilment Centres to help us do this
- Within our warehouses, simulation is used extensively
- Many abstractions added to satisfy our need for determinism
- We start simple, test and measure, then optimise where necessary

We're Hiring

https://careers.oca.do

@OcadoTechnology
@bofalot

/ OcadoTechnology