# Accelerating JUnit Test Performance with Hydra + Jenkins

*Parag Chandra*

*Software Engineer*
*SiriusXM/Pandora*

DEVOPS WORLD
by CloudBees

# My Background

- Android developer for 10+ years
    - Not a very good app developer ;)
    - Prefer working behind the scenes – "developer enablement"

- Started at Pandora in late 2018

- Focused exclusively on tools, CI, test automation, infrastructure for all Android teams

- First task: speed up developer CI builds

# Android PR Builds (Circa Winter 2019)

assemble 10:00

findbugs 10:00

lint 15:00

test 70:00

DEVOPSWORLD
by CloudBees
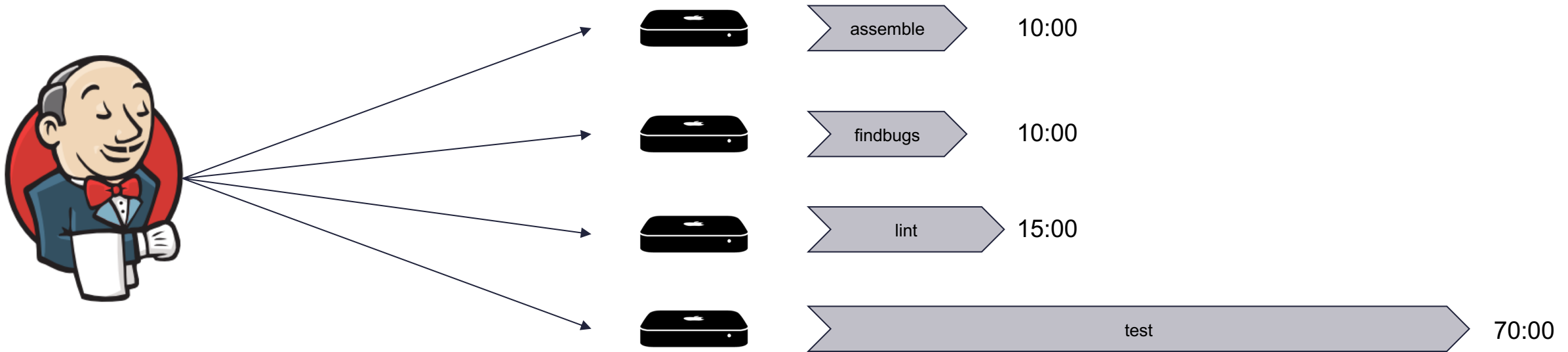
# Test Phase

- 12,000 'headless' - but not necessarily 'unit' - tests ;)

- Too new to understand 10-year-old codebase and disable tests

- Already parallelizing via concurrent Gradle test workers

- So brute force – parallelize across multiple machines

DEVOPSWORLD
by CloudBees

# What is Hydra?

- System to split JUnit test suite across multiple machines
  - 'Standard' Java as well as Android
  - Heterogeneous and homogeneous hardware pools
  - Multiple test partitioning strategies

- Comprised of:
  - Gradle plug-in…
  - + …Spring boot server application…
  - + …CI server (i.e. Jenkins) to coordinate

# Hydra Service

- Containerized Spring Boot application

- Backed by Postgres DB

- Tracks performance & results of each test over time

- Partitions tests into N sets of ~equal runtime

- Results in exclusion list for each machine: i.e. tests to *not run*

**DEVOPS**WORLD
*by CloudBees*

# Test Partitioning – i.e. "Bin-Packing Problem"

- ## Greedy
  - Distribute evenly amongst *N* machines
  - Rebalance after every run
  - Works best w/ homogenous hardware

- ## Greedy w/ failures
  - Greedy, but failing tests always run on same host
  - Helps w/ flaky tests

- ## HostAffinity
  - Try to run same tests on same hosts each time
  - Shift more tests to faster hosts over time – but only if delta is too large
  - Best for heterogeneous hardware

# Gradle Plug-in

- Adds corresponding '_balanced' task for existing test task(s)
  - Specified by developer, can support multiple variants/configurations

- When _balanced task is executed
  - Asks Hydra service for list of tests to *not* run
  - Executes tests as usual
  - Reports test times and results back to Hydra service

# CI Pipeline

- ## Define env vars
  - HYDRA_HOST_LIST, HYDRA_SERVER, JOB_NAME, VM_HOSTNAME

- ## *N* Parallel Stages
  - `checkout scm`
  - `sh './gradlew testDebugUnitTest_balanced'`
  - `stash includes: '**/TEST-*.xml', name: "Executor$i"`

- ## Results Publication
  - `for (i=1; i <= N; i++) { unstash name: "Executor$i" }`
  - `sh 'find . -type f -name 'TEST-*.xml' -exec touch {} \\;'`
  - `junit testResults: '**/TEST-*.xml'`

# Divide & Conquer in a Virtuous Feedback Loop
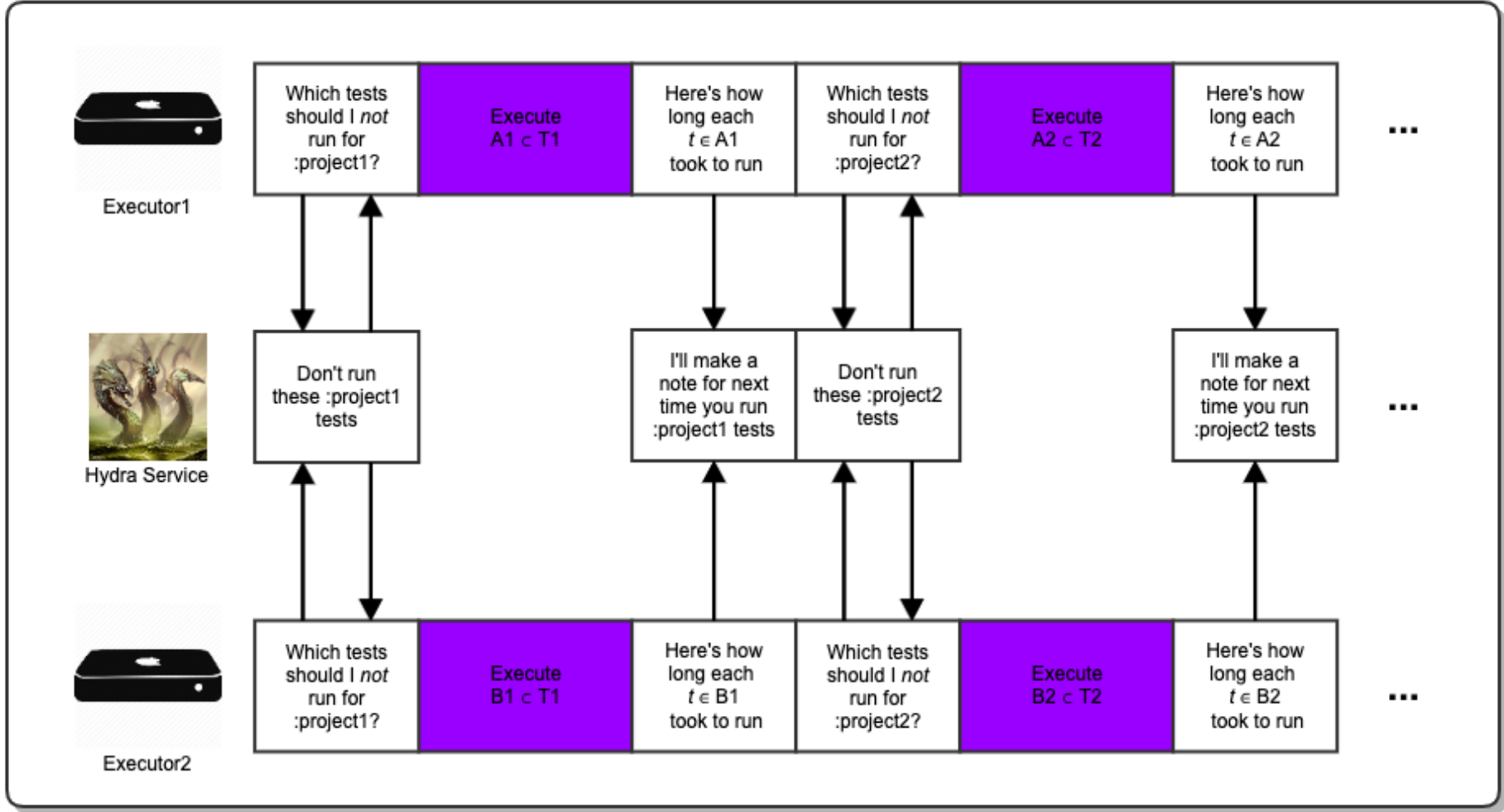
# What to Expect When You're Expecting Faster Tests

- Exclusion lists – tests to *skip*; not tests to *run*
  - Side effect: new / renamed / moved tests run *N* times on first try; once thereafter

- Test splits + constant rebalancing = frequently changing test sequence
  - Will expose implicit test dependencies, aka flaky tests

- Non-linear scaling
  - Each node must still git clone and build
  - Additional overhead of client-server communications
  - Extra time to archive/transmit results back to Jenkins for aggregation

# Results



Avg. Runtime (Minutes)

DEVOPSWORLD
by CloudBees
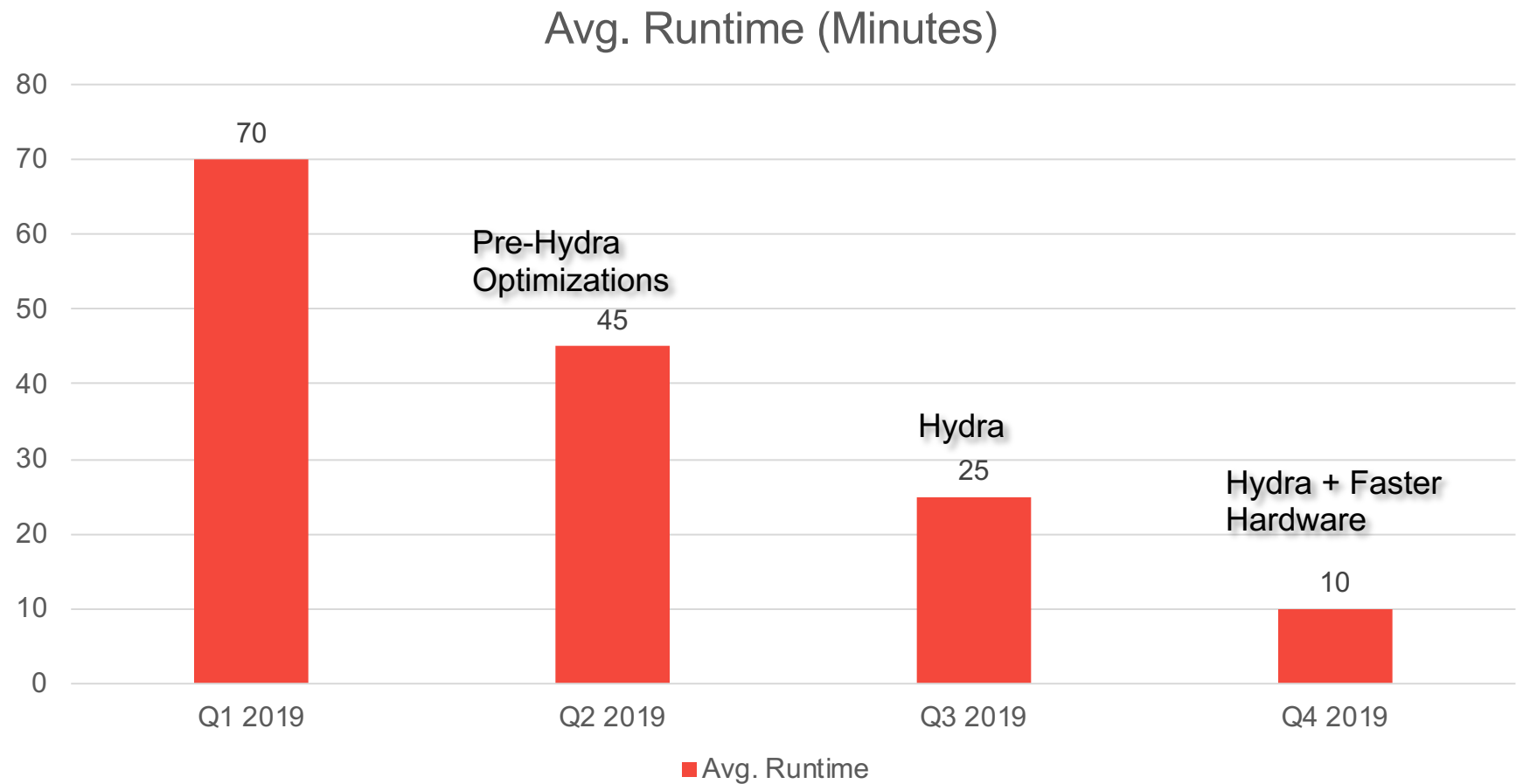
# In closing…

- ## Resources
  - Hydra Open Source Project: https://github.com/PandoraMedia/hydra
  - Bin-packing Problem: https://en.wikipedia.org/wiki/Bin_packing_problem

- ## Acknowledgements
  - Justin Guerra (SiriusXM/Pandora)
  - Aliaksei Dubrouski (LinkedIn)

# Thanks!

*pchandra@pandora.com*

DEVOPS WORLD
by CloudBees