

# 최고의 퍼포먼스를 위한 CI/CD 파이프라인 현대화 및 EDA 전략

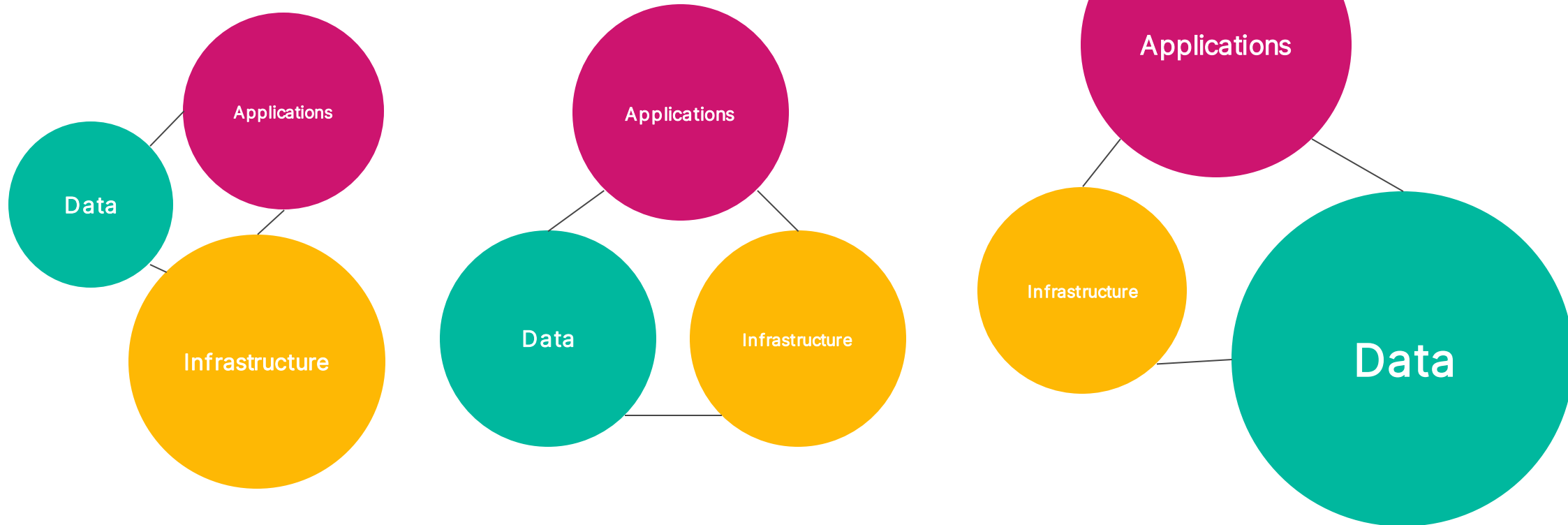
**Bikash Roy Choudhury**

Director of Solutions, HPC/DevOps, Pure Storage

# What is DevOps?

# What does the trend look like?

Application, Infrastructure, Data - Today and beyond



2000



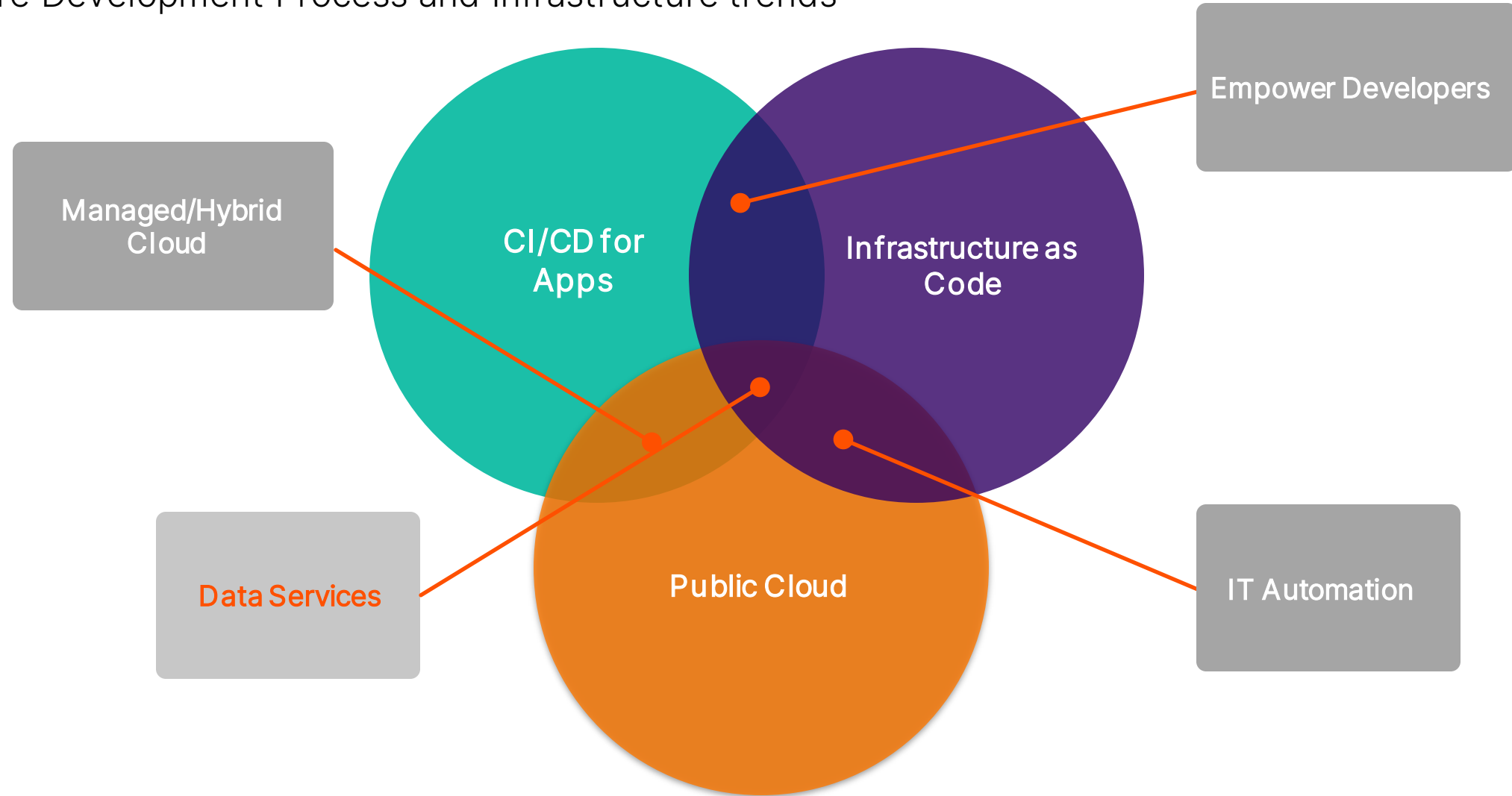
2010

2020



# Evolving Trend

Software Development Process and Infrastructure trends



# What is DevOps?



People

Culture  
Autonomy



Communication  
(Dev & Ops)



Collaboration

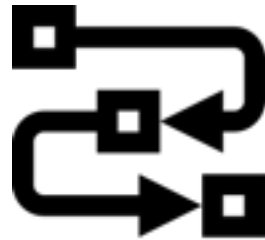


Accountability  
(Blameless culture)

DevOps



Product



Process

Tools  
Workflows



CI/CD



AI/ML



Big Data Analytics

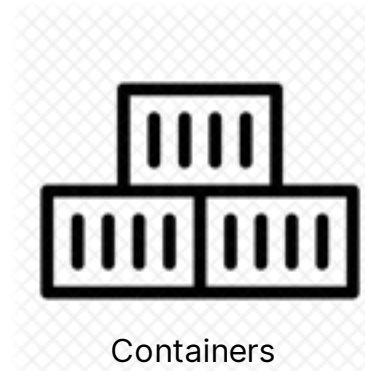


Technology

Workloads  
Consolidation



Automation



Containers



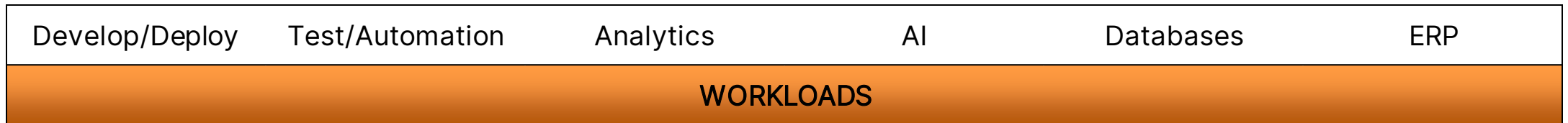
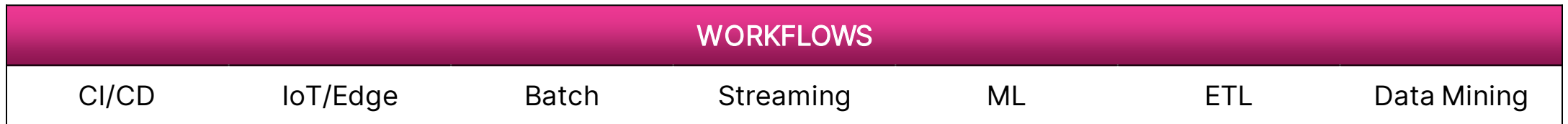
Cloud



# Application Workflows & Workloads

Commercial Applications

Enterprise Applications



# Cloud native Reference Architecture

**Application Development Workflows/tools**  
(Software Development Lifecycle(SDLC), Databases, Analytics)



**Orchestration/ Serverless Computing**  
(Policy & Event driven, Monitoring, logging, Scaling and Routing)



OPENSIFT



VMware Tanzu



docker



kubernetes



MESOS

**Runtime**  
(Abstracted and managed hardware resources and software needed for program execution and operation)

**Provisioning**  
(Configuration Management)

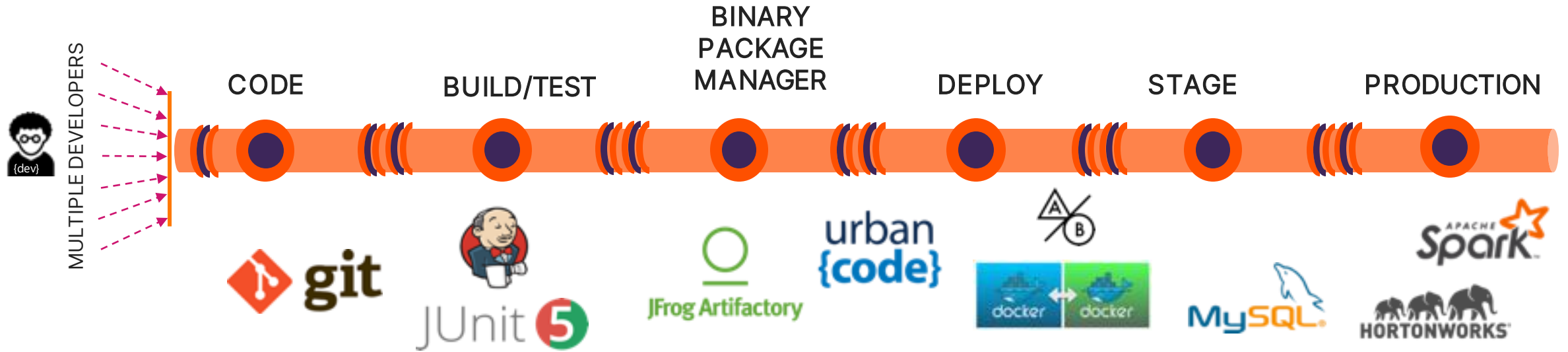


**Infrastructure & Services**  
(Compute, Network, Storage, ELB, EMR, ECS)



# Traditional way of application development

Software Development Process and Infrastructure trends



Scalability – Performance and Capacity

Service Level Objectives – Latency, IOPs, Bandwidth

Manageability – Server sprawl

Data mobility – Network Utilization





# Differentiation

# Pure Storage differentiators for CI/CD workflow

**01** Auto – healing & scaling for various workloads

Improve code quality & reliability

**02** Adaptive SLO for heterogeneous workloads

Speed of software delivery

**03** Array-level compression for entire data pipeline

Reduced cost & IT productivity

**04** Data management capabilities for multi cloud

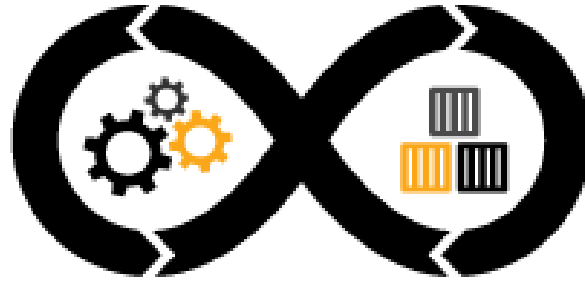
Data homogeneity & continuity

**05** Simple & Easy REST APIs for Zero Storage Touch

Improve customer experience



# What is Continuous?



## Continuous Everything



Continuous Integration (CI)

Identifying bugs at very early part of the development cycle

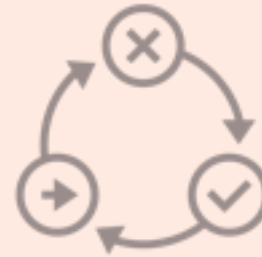
Reduced build time driving innovation



Continuous Deployment (CD)

Provides flexibility to test a feed of small code changes

Promoting and reverting releases between prod., test before release



Continuous Testing (CT)

Appropriate tests run for every code change; Part of the pipeline

Better code quality; fixing bugs quickly



Continuous Delivery (CD)

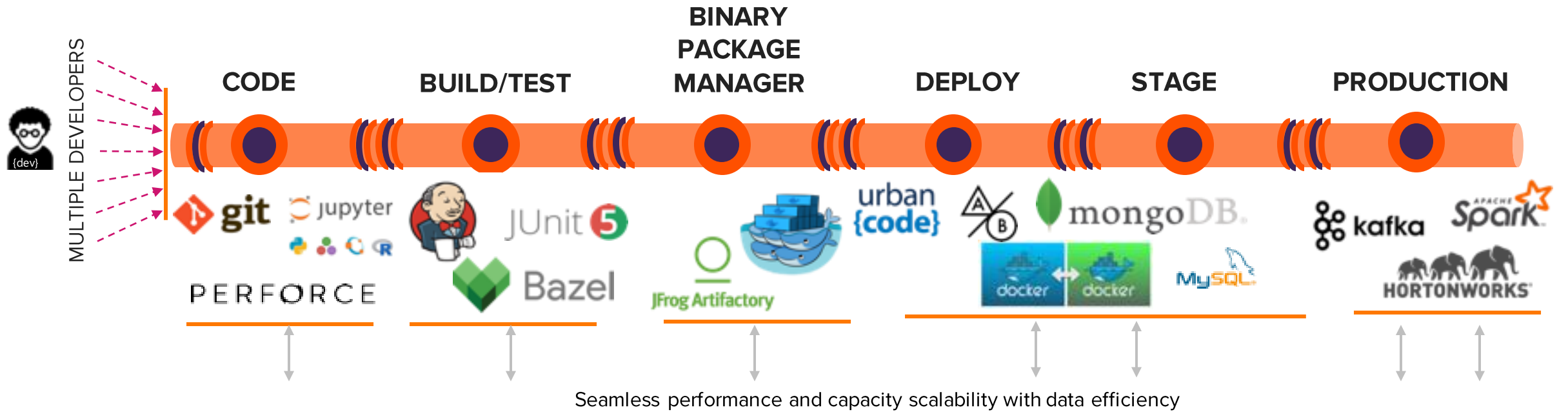
Releasing applications painless and low risk releases

Seamless infrastructure provisioning & ability to operate and scale.



# MODERN CI/CD PIPELINE

FAST AND EFFICIENT with Pure Storage



Fast Commit

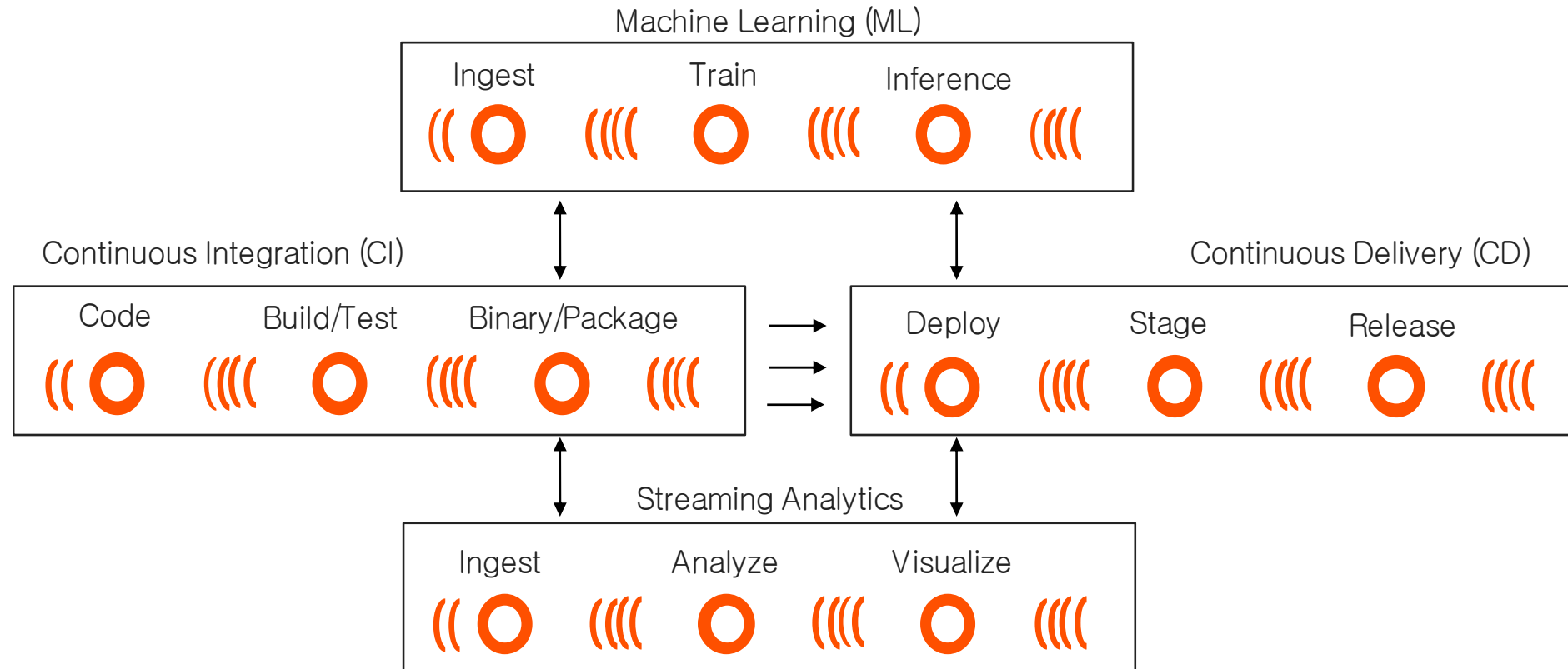
Fast Build/Test

Fast software delivery

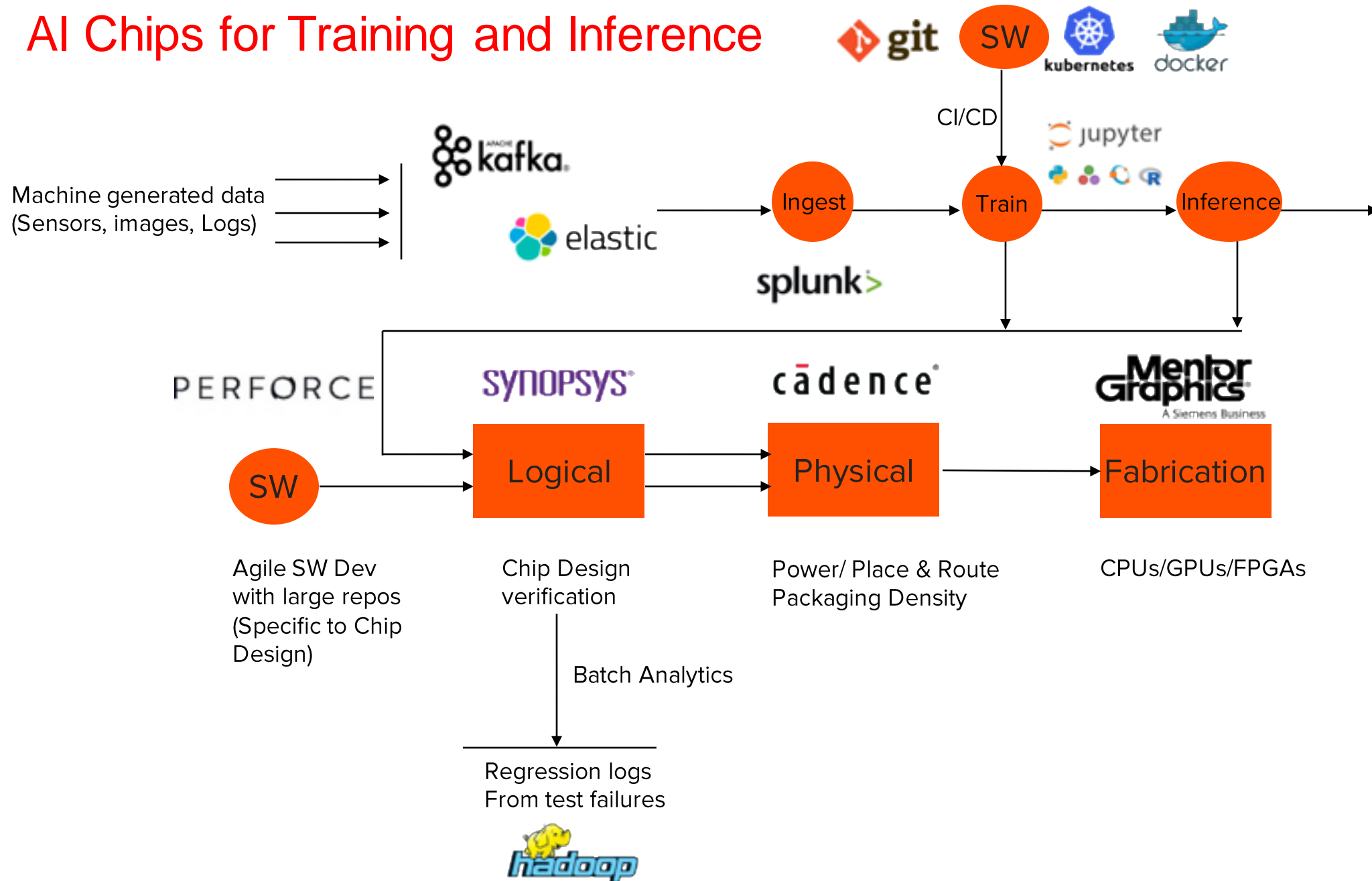
Fast Log Ingest + Batch/Stream Processing Analytics



# Assembly line for different data pipelines



# AI Chips for Training and Inference



# DevOps Partner Solutions & Solutions

Software Delivery  
Tools

Hybrid Cloud/  
Container  
Orchestration

Infrastructure  
Automation

Monitoring &  
Reporting

PERFORCE



Prometheus



Grafana



JFrog Artifactory



Private Docker Registry



openstack.



OPEN METRICS<sub>16</sub>



# Pure Storage use cases and value



## Engineering Productivity

Accelerate CI process with faster build cycles by more than 33%  
Rapid developer workspace creation @ scale

Compute, network and storage efficiency

Rapid recovery from test failures



## Infrastructure Efficiency



## Business & Operation benefits

Improved Code quality; faster TTM  
Standard Data platform; Datahub for all development workloads

Zero Storage touch for developers – APIs, Ansible playbook

Less re-work and more new testing – better performance



## Behavioral benefits







# Customer Stories

## Use Cases- Workloads

- Cell & memory characterization
- Front end verification and regression
- Analog/mixed signal simulation
- Timing

## Software Build and regression

- Physical design
- Tapeout



**One engineer who had 12  
NetApps dedicated to  
him is running on 1  
FlashBlade**

*“Our SW developers are seeing workloads finish greater than 5X faster than status quo with FlashBlade and we are benefiting from 2.5:1 data reduction through compression”*

- Andy Nallappan– CIO Broadcom





# Mentor Questa Build and Regression

2800 core EDA software build and test (200+GB dataset)

- Most of the workload transacted out of NVRAM
- Only 20GB actually hit flash

*Job completed in record time, faster than any other array tested (compared to 20 other platforms)*

*Helped removed need for home-grown caching system*

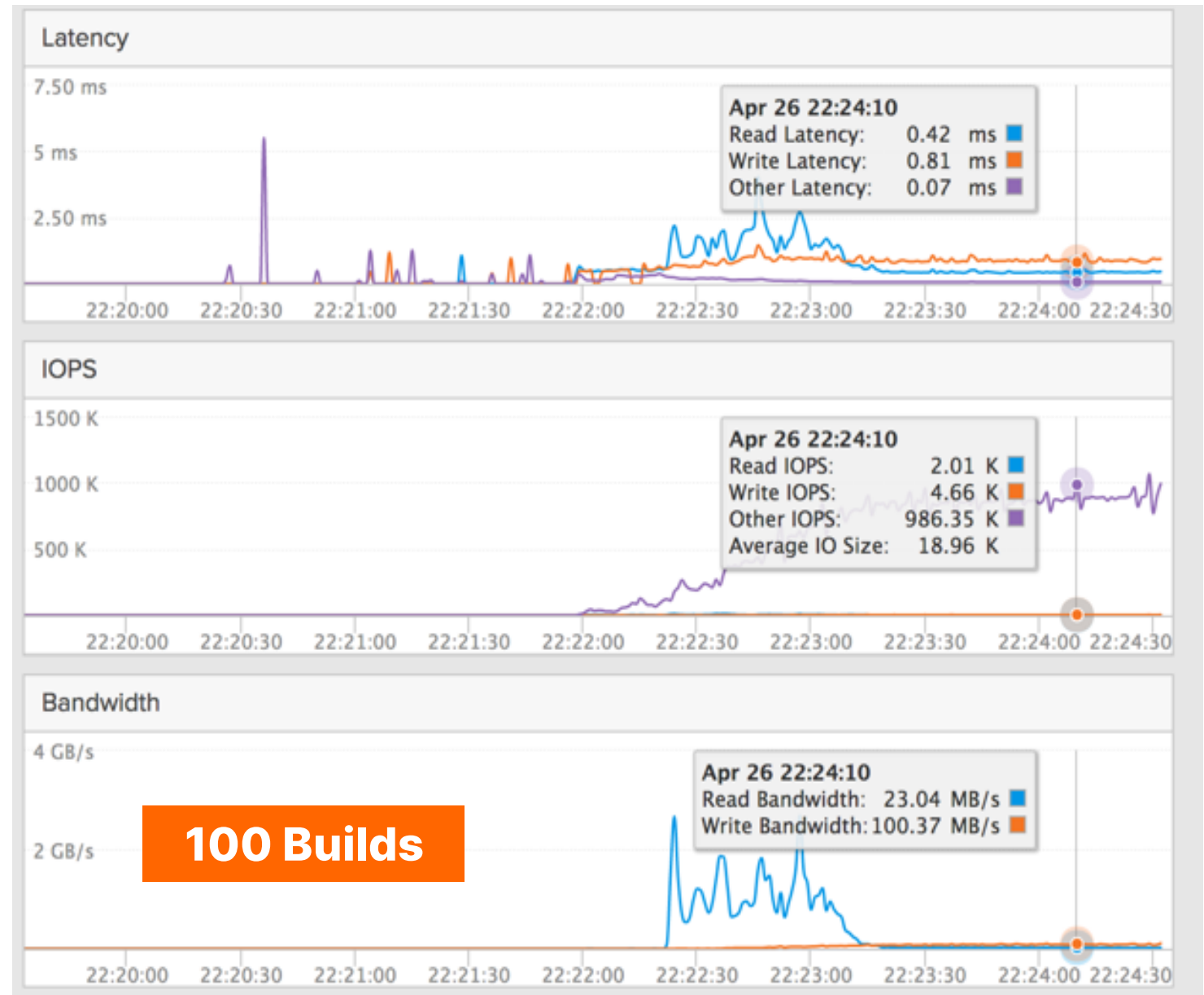
| System                                   | Run Time  |
|--|---|
| Previous Best out of 20 platforms tested | 17:48 (average over 40min)                        |
| <b>FlashBlade (15 blade)</b>             | <b>11:12 (within 1 minute of local disk time)</b> |

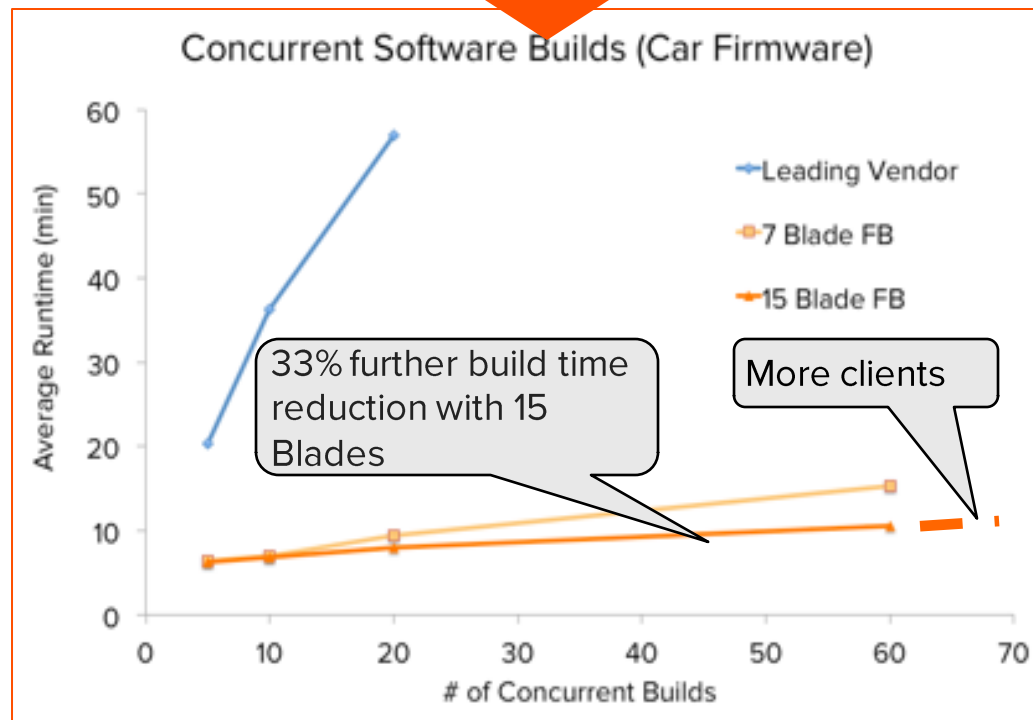
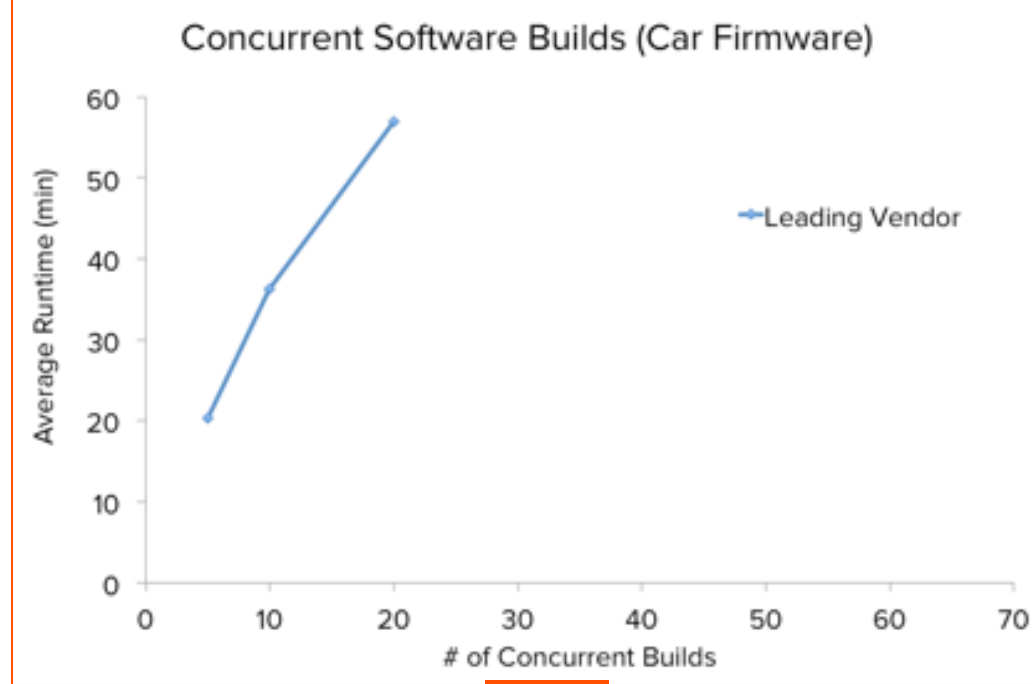
**Run Time Improvement ~ 40%**  
**(11:12 vs 17:48, 2800 parallel cores)**

| #Builds | Avg Time (s)             |
|---------|--------------------------|
| 1       | 237                      |
| 10      | 235                      |
| 25      | 235                      |
| 50      | 225                      |
| 75      | 238                      |
| 100     | 267 (CPU oversubscribed) |

# Linux Kernel builds (-j6)

64K files, 2.3 GB per build





# 3X FASTER BUILDS, 3X HIGHER SCALE with FlashBlade



# Thank you

