WEKA.IO

World's Fastest File System

Using D as the programming language of choice for large scale primary storage system

Liran Zvibel
WekaIO , CEO & Co-Founder
@liranzvibel

Agenda

- History and background
- WekaIO intro
- Where we stand now
- Mecca unveiled
- Q&A

History and background

Kent Beck

@KentBeck

Following

flint, a C++ linter written in D
code.facebook.com/posts/72970934 …
interesting tricks possible with compile-time interpretation

Under the Hood: Building and open-sourcing flint

Flint, Facebook's lint program, issues the lint errors and warnings appear automatically in our code review system (phabricator) alongside each proposed code change, notifying the program...

code.facebook.com

4:50 PM - 24 Feb 2014

15 Retweets **26** Likes

1 15 26

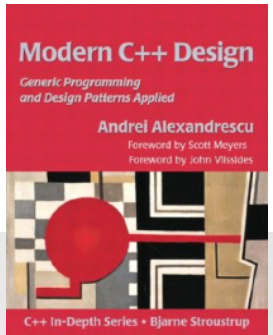
Tweet your reply

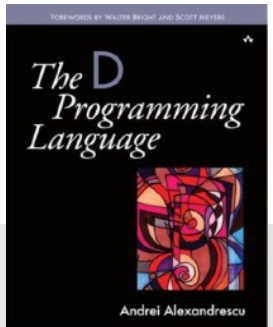
February 24, 2014 INFRA · OPEN SOURCE

Under the Hood: Building and open-sourcing flint

Andrei Alexandrescu

Lint programs are an odd class of program verifiers, and for a while I wasn't convinced they were something I should focus on building out for Facebook. I don't like the style police on my back, and false error warnings can trip up an entire task. There's a lot of good, however, about a verifier that mechanically looks for issues that are not traditionally monitored by the compilers and that would almost always improve code quality once fixed.





WEKA.IO

Using D for Development of Large Scale Primary Storage

Liran Zvibel
Weka.IO, CTO
liran@weka.io
@liranzvibel

WEKA.IO

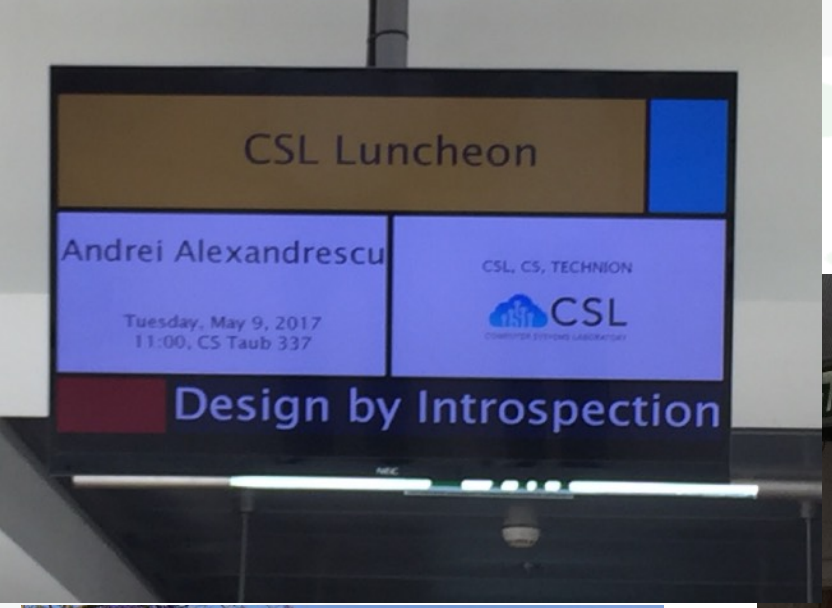
Using D for Development of Large Scale Primary Storage

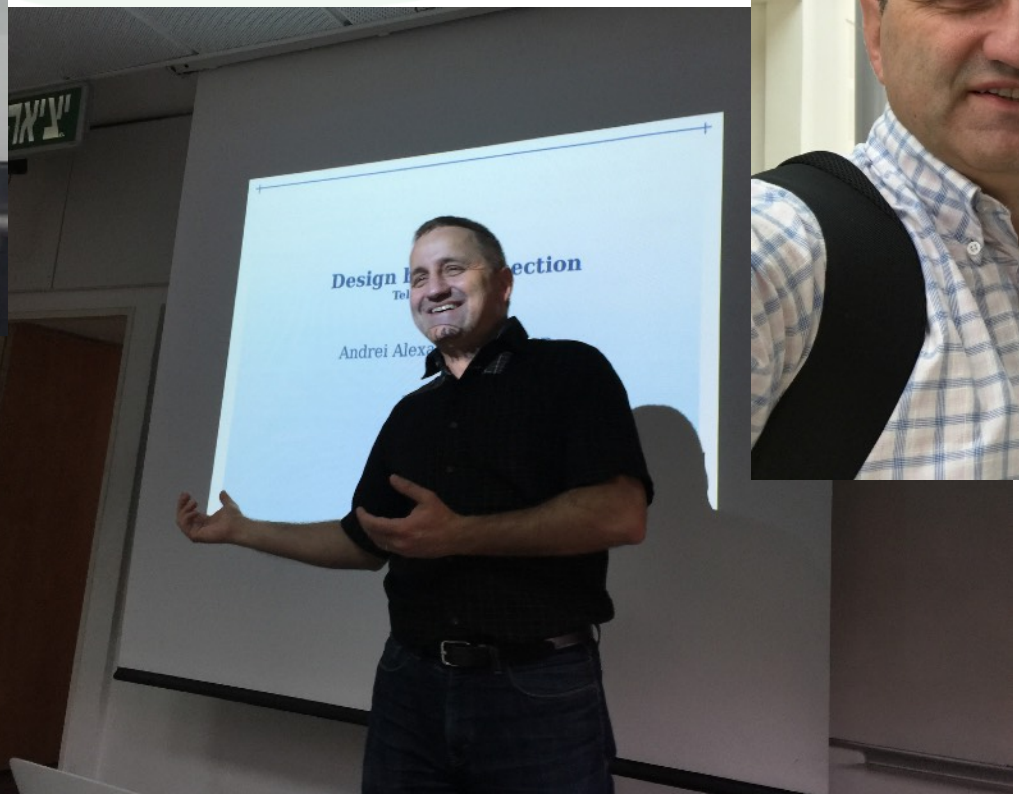
Liran Zvibel
Weka.IO, CTO
liran@weka.io
@liranzvibel

#DConf2016

After DConf 2015 …

- David Nadlinger came to the rescue and fixed LDC for us
- Were able to combat optimizations and runtime issues
- Started working towards no-GC runtime
- Code size and complexity started hitting us (symbol length, compilation time, exe size, etc)
- Johan Engelen stepped in to maintain LDC for us and bridge our work with DMD





The D Blog

The official blog for the D Programming Language.

Introspection, Introspection Everywhere

May 22, 2017 Andrei Alexandrescu

Prelude: Orem, UT, May 29 2015

Short summary

- The D language is proving to be critical to our success
- WekaIO Matrix is a large and complex project
- D Language allows us to have a single language and codebase for data path and also control plane
- Introspection, CTFE and meta programming allow us to manage complexity of the project
- Could improve support for large projects, and also use cases that require real time (not just java or python that compiles) around safety and GC
  - No programming language is perfect, though!

WekaIO introduction

WekaIO Introduction

WHO WE ARE

WekaIO Matrix is the fastest, most scalable parallel file system for AI and technical compute workloads that ensures your applications never wait for data.

THE PEOPLE

THE PARTNERS

THE ACCOLADES

Premium Customers

DREAMWORKS

*At Dreamworks Animation, we constantly strive to provide technology solutions that **remove barriers to creativity**... With WekaIO as part of our HPE High Performance Compute cluster, file service scalability and reliability issues are a thing of the past. We're using the WekaIO Matrix file system as **burst-buffer style transient storage for the most demanding render and simulation workloads** in our pipeline.*

Scott Miller, Technology Fellow Engineering and Infrastructure

*WekaIO demonstrated that it was the only file system that could **fully saturate the GPU cluster**. With WekaIO, the data scientists were able to significantly improve productivity by removing time consuming data copy tasks into local disks. In addition WekaIO provided **seamless integration to their massive training system data lake***

tgen® TRANSLATIONAL GENOMICS RESEARCH INSTITUTE

*Future-thinking companies like WekaIO, complement our core principle of **accelerating research and discovery**. The ability to run more concurrent high performance genomic workloads will significantly advance our time to discovery.*

Nelson Kick, Manager of HPC Operations

SDSC SAN DIEGO SUPERCOMPUTER CENTER

*We are using WekaIO technologies over **InfiniBand** to address the challenges of **data analytics at extreme scale** in life sciences, particle physics, geosciences, and other fields. That process is still ongoing but to-date we've already achieved some promising results.*

Michael Norman, Director of San Diego Supercomputer Center at UCSD

Nasdaq

Hot Topics: ETFs | Smart

Our Businesses ▼ Quotes ▼ Markets ▼

Enter symbol, nam

HPE Launches Vertical AI Solutions, Dramatically Accelerates Deep Learning Training

By GlobeNewswire, March 21, 2018, 07:45:00 AM EDT

Data Centre ▸ Storage

WekaIO pulls some Matrix kung fu on SPEC file system benchmark

Like a bat out of parallel...

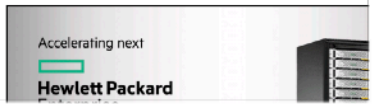
By Chris Mellor 22 Mar 2018 at 11:12 16 SHARE ▼

GET SUPERCHARGED AI-READY INFRASTRUCTURE WITH NEXT-GENERATION STORAGE SOLUTIONS

Darrin P. Johnson, Director of Technical Marketing, NVIDIA

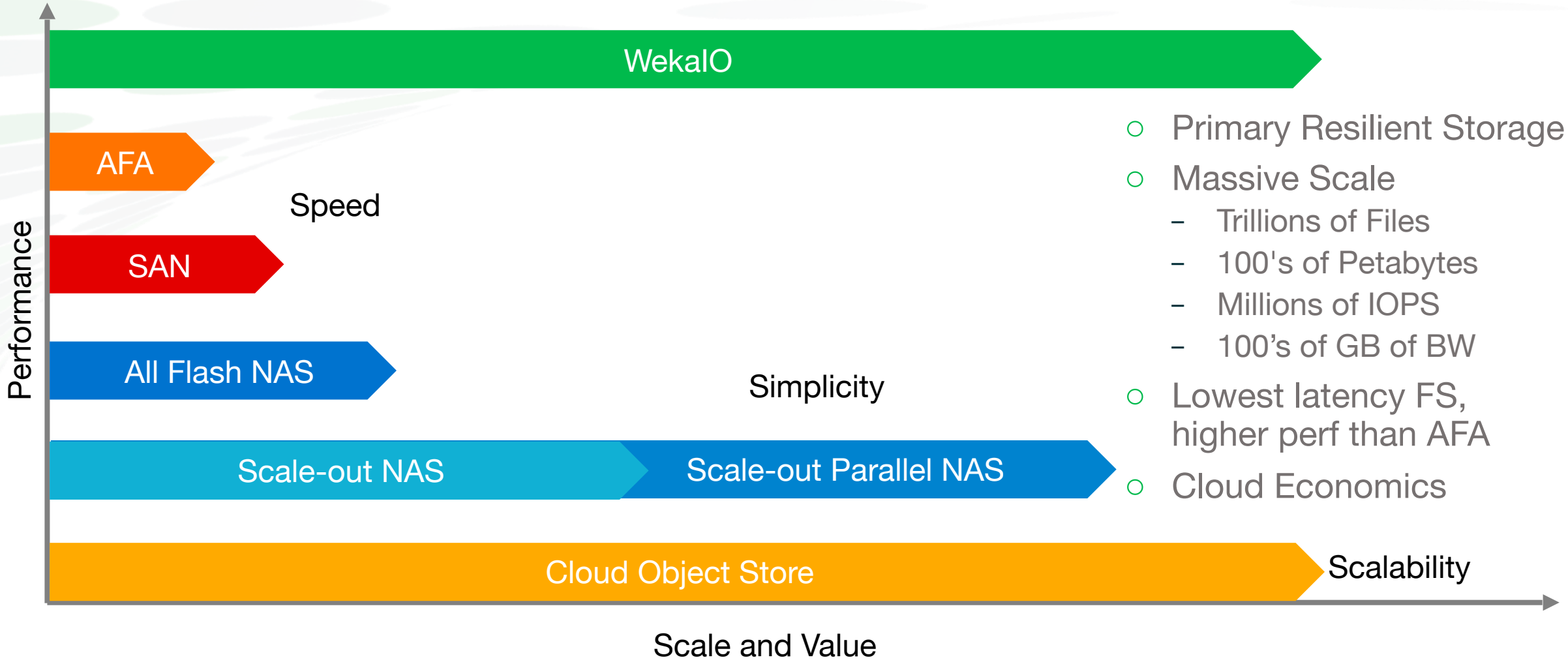
Is it worth taking out a file on WekaIO? It seems to be disrupting the data industry

The artificial intelligence data player is on the hunt for channel partners and Nick Booth thinks the firm is worth a closer look



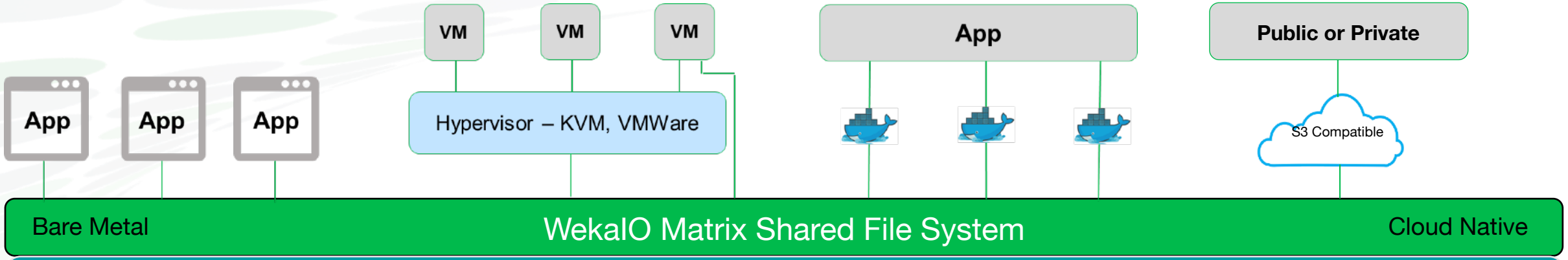
WEKA.IO

Highest Performance Primary Resilient Storage at Scale



- Primary Resilient Storage
- Massive Scale
  - Trillions of Files
  - 100's of Petabytes
  - Millions of IOPS
  - 100's of GB of BW
- Lowest latency FS, higher perf than AFA
- Cloud Economics

WekaIO Matrix: Full-featured and Flexible



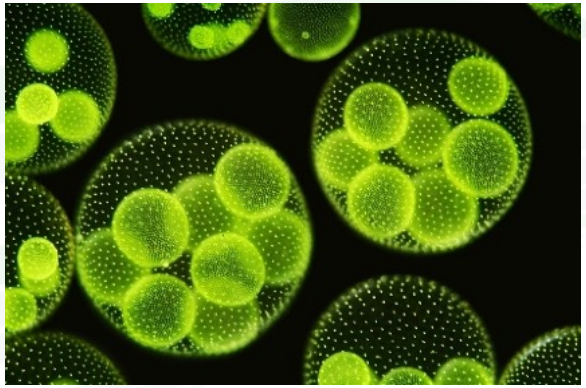
Fully Coherent POSIX File System That Delivers Local File System Performance

Distributed Coding, More Resilient at Scale, Fast Rebuilds, End-to-End DP

Instantaneous Snapshots, Clones, Tiering to S3, Partial File Rehydration

InfiniBand or Ethernet, Hyperconverged or Dedicated Storage Server

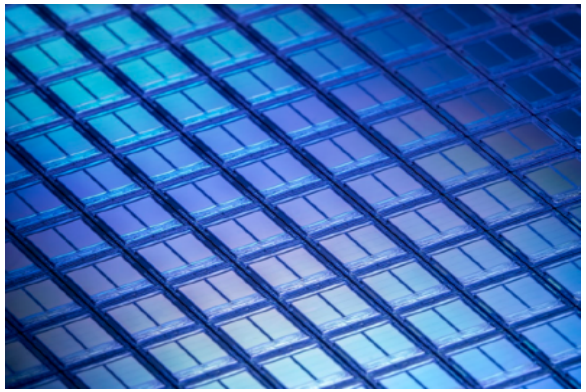
Focused On the Most Demanding Workloads



- Genomics sequencing and analytics
- Drug discovery
- Microscopy

- Autonomous cars
- Machine Learning & AI
- IoT

- Business analytics (SAS Grid, SAP HANA)
- Algorithmic trading
- Risk analysis (Monte Carlo simulation)



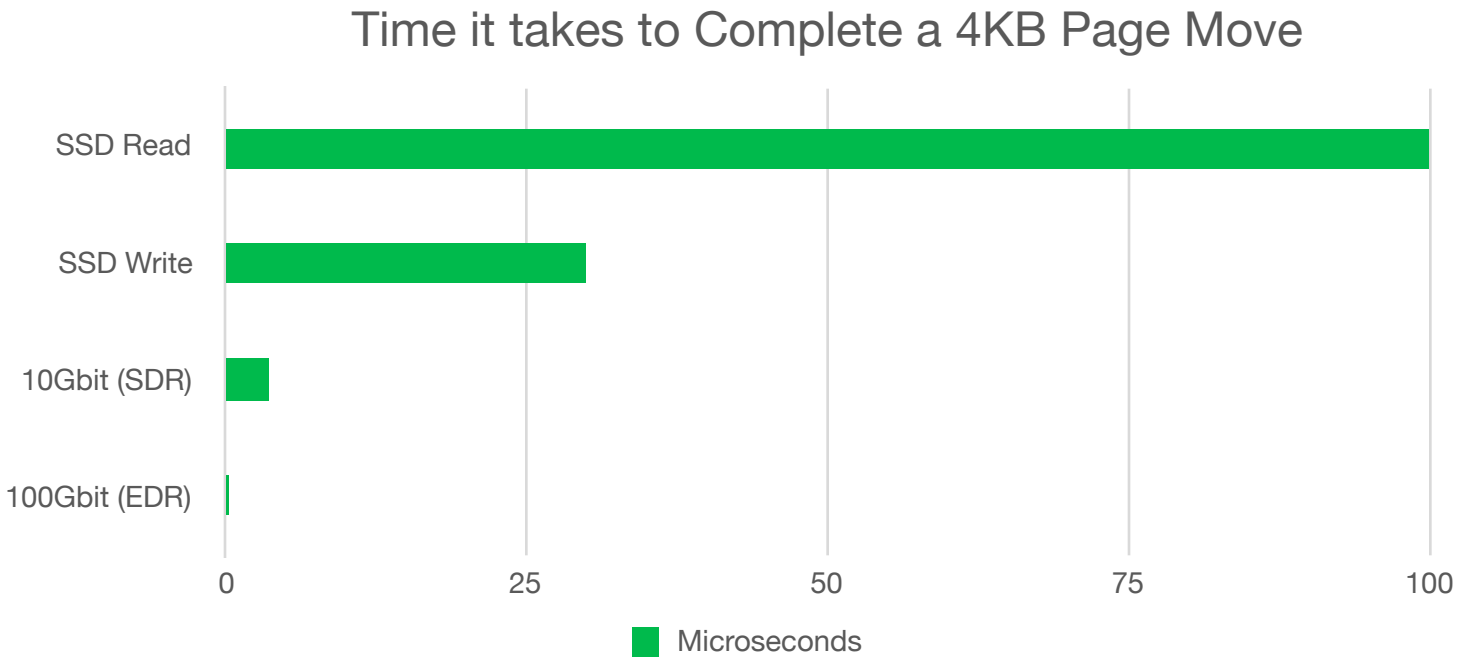
- Semiconductor verification
- Manufacturing (CFD)
- Software compilation

- Media rendering
- Transcoding
- Visual Effects (VFX)

- DevOps
- Real-time analytics
- Batch analytics

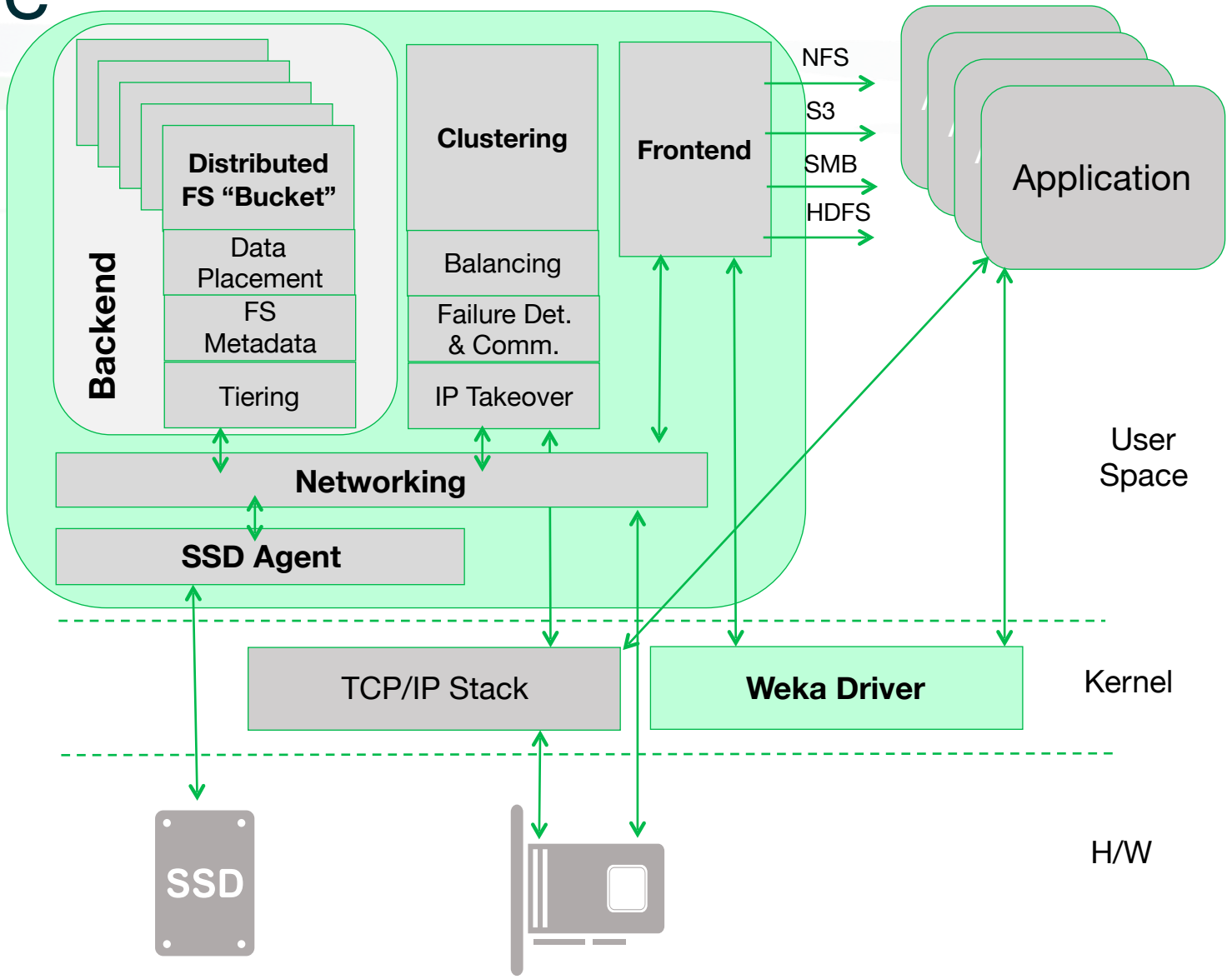
Why Data Locality is Irrelevant

- Local copy architectures (e.g. Hadoop, or caching solutions) were developed when 1GbitE and HDDs were standard
- Modern networks on 10Gbit Ethernet are 10x faster than SSD
- It is much easier to create distributed algorithms when locality is not important
- With right networking stack, shared storage is faster than local storage



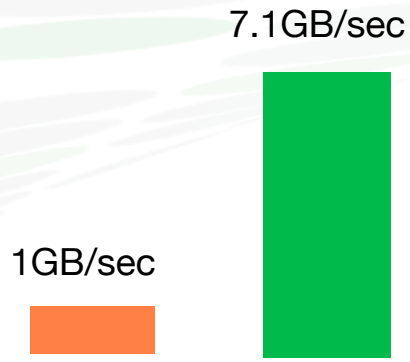
Software Architecture

- Runs inside LXC container for isolation
- SR-IOV to run network stack and NVMe in user space
- Provides POSIX VFS through lockless queues to WekaIO driver
- I/O stack bypasses kernel
- Scheduling and memory management also bypass kernel
- Metadata split into many Buckets – Buckets quickly migrate ➔ no hot spots
- Support, bare metal, container & hypervisor

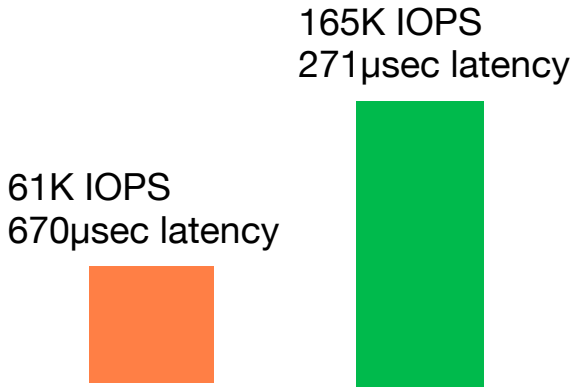


Actual Results from Deep Learning Bake-off

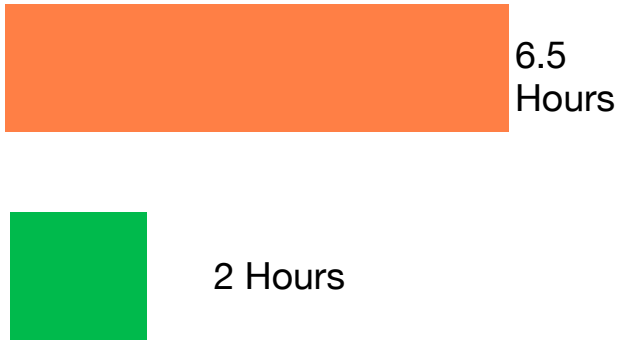
7x Faster 1MB Throughput



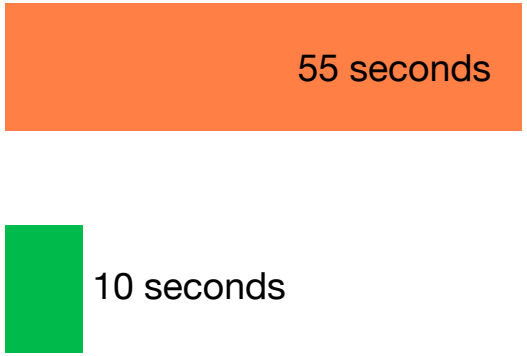
2.6x Better 4KB IOPS/Latency



3.3x Better FS Traverse (Find)

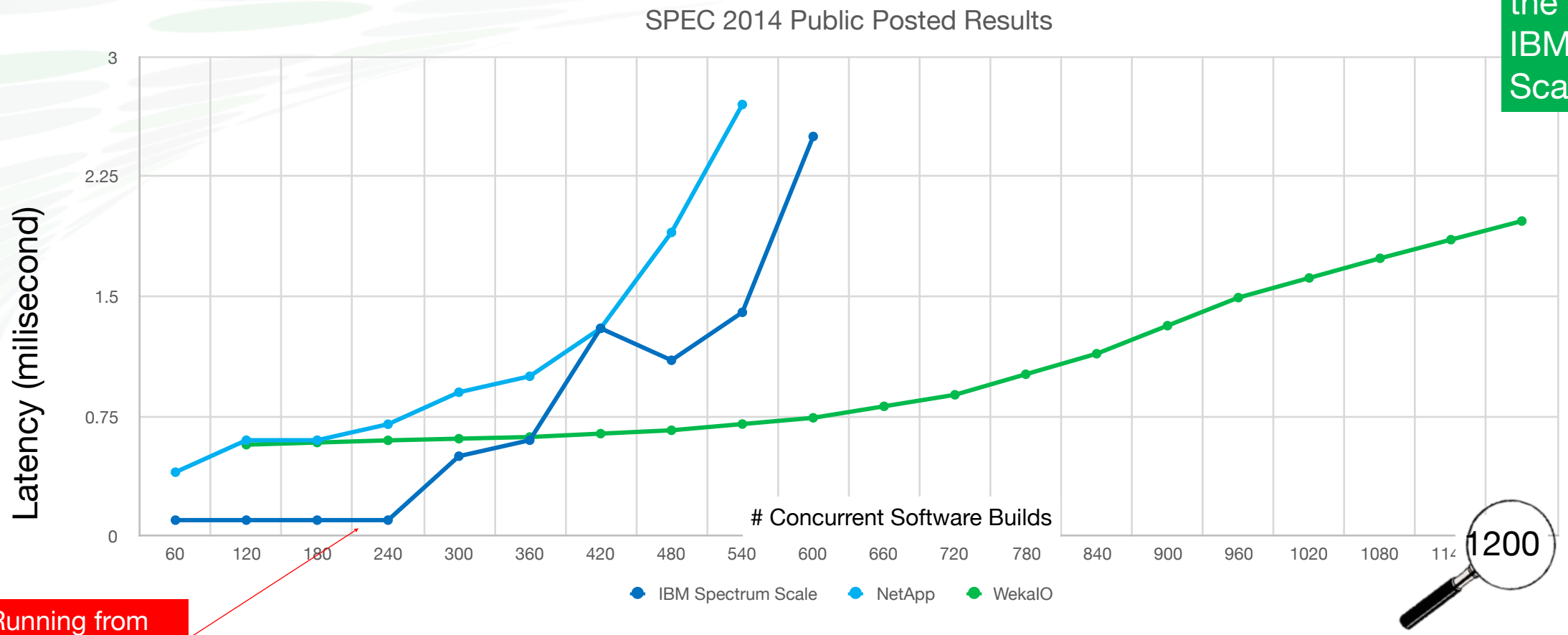


5.5x Better ‘ls” Directory



Fastest File System

WekaIO does 2x the workload of IBM Spectrum Scale



Running from RAM cache

Current state of the project

Some statistics

- 1232 .d files
- About 280 KLOC
- About 2k ‘`static if`’ statements
- 20 ‘`static foreach`’ statements
  - Probably many more foreach indeed static
- 115 ‘`mixin template`’
- About 27,500 explicit template instantiations (with ‘`!`’)
- 30 mentions of ‘`__ctfe`’ in code, countless usage of actual

Anecdotal cool example — verifying ABI for RPC

- Enterprise systems must support seamless upgrades
- Upgrades are performed as a “rolling”process
- Two versions must know whether RPC is ABI compatible or not.
- Standard mangling is not enough, as types may have changed between versions
- Introspection allows our no-IDL RPCs to automatically verify ABI compatibility by recursively opening structs and hashing the whole result

Anecdotal pain point — delegates, scope and GC

- GC cannot be used in a real time, low latency based system
- Delegates generate GC by default, as their scope may escape the current one (we cannot know that the stack remains in the scope)
- Even simple std.algorithm examples, where all executing is recursive and would stay on the stack force GC allocations
- No effective way of marking such delegates as scoped so this won't happen

What do we care about?

- Safety
- Performance
- Brevity
- Ability to manage complexity

- What we don’t need and others do : “First 5 minutes!”
  - Community must get D easier to start with

Mecca Unvailed

Again, some history

- Work started in August 2016 by Tomer Filiba

```
commit 51182a64360518aa4cbabfe1ce99561d2584378a
Author: Tomer Filiba <tomer@weka.io>
Date:   Mon Aug 29 23:50:53 2016 +0300

    Mecca: make weka's infrastructure great again
```

- Moved to external repository May 2017
- Shachar Shemesh started working full time June 2017

- Mecca is our OS implementation, sans IO and networking modules

Some statistics

- 3 major components: Reactor, lib, containers
- 20575 LOC: 8361 in reactor; 7782 in lib, 4432 in containers

- Reactor — scheduling fibers coordinating (synchronizing)
- non-GC containers — Arrays, pools, queues, linked lists
- Lib — introspection, division, no-gc exception handling, CTFE enabled hashing, non-gc interators and algs, string and time manipulation.

WEKA.IO