CrailProposal

Crail

Abstract

Crail is a storage platform for sharing performance critical data in distributed data processing jobs at very high speed. Crail is built entirely upon principles of user-level I/O and specifically targets data center deployments with fast network and storage hardware (e.g., 100Gbps RDMA, plenty of DRAM, NVMe flash, etc.) as well as new modes of operation such resource disaggregation or serverless computing. Crail is written in Java and integrates seamlessly with the Apache data processing ecosystem. It can be used as a backbone to accelerate high-level data operations such as shuffle or broadcast, or as a cache to store hot data that is queried repeatedly, or as a storage platform for sharing inter-job data in complex multi-job pipelines, etc.

Proposal

Crail enables Apache data processing frameworks to run efficiently in next generation data centers using fast storage and network hardware in combination with resource (e.g., DRAM, Flash) disaggregation.

Background

Crail started as a research project at the IBM Zurich Research Laboratory around 2014 aiming to integrate high-speed I/O hardware effectively into large scale data processing systems.

Rational

During the last decade, I/O hardware has undergone rapid performance improvements, typically in the order of magnitudes. Modern day networking and storage hardware can deliver 100+ Gbps (10+ GBps) bandwidth with a few microseconds of access latencies. However, despite such progress in raw I/O performance, effectively leveraging modern hardware in data processing frameworks remains challenging. In most of the cases, upgrading to high-end networking or storage hardware has very little effect on the performance of analytics workloads. The problem comes from heavily layered software imposing overheads such as deep call stacks, unnecessary data copies, thread contention, etc. These problems have already been addressed at the operating system level with new I/O APIs such as RDMA verbs, NVMe, etc., allowing applications to bypass software layers during I/O operations. Distributed data processing frameworks on the other hand, are typically implemented on legacy I/O interfaces such as such as sockets or block storage. These interfaces have been shown to be insufficient to deliver the full hardware performance. Yet, to the best of our knowledge, there are no active and systematic efforts to integrate these new user level I/O APIs into Apache software frameworks. This problem affects all end-users and organizations that use Apache software. We expect them to see unsatisfactory small performance gains when upgrading their networking and storage hardware.

Crail solves this problem by providing an efficient storage platform built upon user-level I/O, thus, bypassing layers such as JVM and OS during I/O operations. Moreover, Crail directly leverages the specific hardware features of RDMA and NVMe to provide a better integration with high-level data operations in Apache compute frameworks. As a consequence, Crail enables users to run larger, more complex queries against ever increasing amounts of data at a speed largely determined by the deployed hardware. Crail is generic solution that integrates well with the Apache ecosystem including frameworks like Spark, Hadoop, Hive, etc.

Initial Goals

The initial goals to move Crail to the Apache Incubator is to broaden the community, and foster contributions from developers to leverage Crail in various data processing frameworks and workloads. Ultimately, the goal for Crail is to become the de-facto standard platform for storing temporary performance critical data in distributed data processing systems.

Current Status

The initial code has been developed at the IBM Zurich Research Center and has recently been made available in GitHub under the Apache Software License 2.0. The Project currently has explicit support for Spark and Hadoop. Project documentation is available on the website www.crail.io. There is also a public forum for discussions related to Crail available at https://groups.google.com/forum/#!forum/zrlio-users.

Mericrotacy

The current developers are familiar with the meritocratic open source development process at Apache. Over the last year, the project has gathered interest at GitHub and several companies have already expressed interest in the project. We plan to invest in supporting a meritocracy by inviting additional developers to participate.

Community

The need for a generic solution to integrate high-performance I/O hardware in the open source is tremendous, so there is a potential for a very large community. We believe that Crail’s extensible architecture and its alignment with the Apache Ecosystem will further encourage community participation. We expect that over time Crail will attract a large community.

Alignment

Crail is written in Java and is built for the Apache data processing ecosystem. The basic storage services of Crail can be used seamlessly from Spark, Hadoop, Storm. The enhanced storage services require dedicated data processing specific binding, which currently are available only for Spark. We think that moving Crail to the Apache incubator will help to extend Crail’s support for different data processing frameworks.

Known Risks

To-date, development has been sponsored by IBM and coordinated mostly by the core team of researchers at the IBM Zurich Research Center. For Crail to fully transition to an "Apache Way" governance model, it needs to start embracing the meritocracy-centric way of growing the community of contributors.

Orphaned Products

The Crail developers have a long-term interest in use and maintenance of the code and there is also hope that growing a diverse community around the project will become a guarantee against the project becoming orphaned. We feel that it is also important to put formal governance in place both for the project and the contributors as the project expands. We feel ASF is the best location for this.

Inexperience with Open Source

Several of the initial committers are experienced open source developers (Linux Kernel, DPDK, etc.).

Relationships with Other Apache Products

As of now, Crail has been tested with Spark, Hadoop and Hive, but it is designed to integrate with any of the Apache data processing frameworks.

Homogeneous Developers

The project already has a diverse developer base including contributions from organizations and public developers.

An Excessive Fascination with the Apache Brand

Crail solves a real need for a generic approach to leverage modern network and storage hardware effectively in the Apache Hadoop and Spark ecosystems. Our rationale for developing Crail as an Apache project is detailed in the Rationale section. We believe that the Apache brand and community process will help to us to engage a larger community and facilitate closer ties with various Apache data processing projects.

Documentation

Documentation regarding Crail is available at www.crail.io

Initial Source

Initial source is available on GitHub under the Apache License 2.0:

- https://github.com/zrlio/crail

External Dependencies

Crail is written in Java and currently supports Apache Hadoop MapReduce and Apache Spark runtimes. To the best of our knowledge, all dependencies of Crail are distributed under Apache compatible licenses.

Required Resource

Mailing lists

- private@crail.incubator.apache.org
- dev@crail.incubator.apache.org
- commits@crail.incubator.apache.org

Git repository

- https://git-wip-us.apache.org/repos/asf/incubator-crail.git

Issue Tracking

- JIRA (Crail)

Initial Committers

- Patrick Stuedi <stu AT ibm DOT zurich DOT com>
- Animesh Trivedi <atr AT ibm DOT zurich DOT com>
- Jonas Pfefferle <jpf AT ibm DOT zurich DOT com>
- Bernard Metzler <bmt AT ibm DOT zurich DOT com>
- Michael Kaufmann <kau AT ibm DOT zurich DOT com>
- Adrian Schuepbach <dri AT ibm DOT zurich DOT com>
- Patrick McArthur <patrick AT patrickmcarthur DOT net>
- Ana Klimovic <anakli AT stanford DOT edu>
- Yuval Degani <yuvaldeg AT mellanox DOT com>
- Vu Pham <vuhuong AT mellanox DOT com>

Affiliations

- IBM (Patrick, Stuedi, Animesh Trivedi, Jonas Pfefferle, Bernard Metzler, Michael Kaufmann, Adrian Schuepbach)
- University of New Hampshire (Patrick McArthur)
- Stanford University (Ana Klimovic)
- Mellanox (Yuval Degani, Vu Pham)

Sponsors

Champion

Luciano Resende <lresende AT apache DOT org>

Nominated Mentors

Luciano Resende <lresende AT apache DOT org>

Raphael Bircher <rbircher AT apache DOT org>

Julian Hyde <jhyde AT apache DOT org>

Sponsoring Entity

We would like to propose the Apache Incubator to sponsor this project.