GAIA: Strengthening the Reliability of Datacenter Computing with Fast and Scalable Consensus

Abstract:

To process the rapidly increasing amount of data, more and more software applications run within a datacenter containing massive computing resources. To harness these resources, applications are mainly ran by two complementary types of infrastructures: schedulers and virtual machines (VM). These infrastructures have brought many benefits, including improving resource utilization and balancing computer load. Unfortunately, as an application runs on more computers, minor computer failures will occur more likely at runtime and can turn down the entire application, causing disasters such as the 2015 NYSE trading halts and recent Facebook outages. Existing infrastructures lack a high availability support for applications.

This GAIA project takes a holistic methodology to greatly improve application availability via three objectives. First, we will create a fast, scalable distributed consensus protocol for general applications. Distributed consensus, a strong fault-tolerance theory, runs multiple replications of the same application and makes these replications behave consistently as long as a majority of them still work normally. An open challenge is that traditional consensus protocols are too slow because their protocol messages go through various software layers (e.g., OS kernels). We will create a fast consensus protocol called FALCON with an ultra-fast OS kernel bypassing technique called Remote Direct Memory Access. Preliminary results published in [SOSP '15] show that FALCON supports unmodified real-world applications, and it is 11.2X faster than traditional protocols even when running 35X more replications than these protocols.

Second, we will construct the first datacenter scheduler for application fault-tolerance by integrating FALCON with popular schedulers. To seamlessly allocate computing resources and replications, we propose a novel replication-aware resource allocation workflow. Preliminary results published in [APSys '16] show that our scheduler supports popular key-value applications efficiently (only 4.2% performance overhead).

Third, we will make FALCON and VMs form a mutual-beneficial eco-system. This eco-system not only leverages the VM hypervisor layer to achieve transparent replication, but it also nurtures a lightweight VM live migration approach for balancing computer load. During a VM migration to a remote computer, prior approaches incur substantial VM downtime or resource consumption on the local computer. Our new approach needs only migrate a FALCON consensus leadership, which consumes little time and resource.

This GAIA project will greatly strengthen the reliability of many datacenter applications and will benefit almost all computer users and software vendors, including many HK financial platforms. GAIA will also advance broad datacenter techniques (e.g., VM migration) and attract researchers to build more reliable infrastructures.

Long term impact:
The emergence of big data with its increasing computational demand is pushing software applications to embrace more and more computing resources. Therefore, applications now often run within a datacenter containing numerous computers. Many applications are mission-critical (e.g., financial platforms, social network platforms, and medical services), so they naturally desire both high reliability and performance. To harness the massive computing resources, applications are mainly ran by two complementary datacenter infrastructures: schedulers and virtual machines (VM).

Unfortunately, as an application runs on more computers, minor computer failures will occur more likely at runtime. If a failure computer happens to run an important application component, the entire application can be turned down. For instance, due to minor computer errors, New York Stock Exchange (NYSE) had a whole-site outage in 2015. Moreover, although social network applications tend to be online 24-7, minor computer failures have turned down the whole Facebook site for several times in recent years.

A key problem causing these outages is that datacenter infrastructures lack a high availability support for applications. The proposed GAIA project tackles this problem with a holistic methodology: it first builds a fast, scalable consensus protocol, it then integrates this protocol into two major infrastructures.

This methodology has two major benefits. First, we can greatly improve the fault-tolerance of mission-critical applications. Distributed consensus is recognized a strong fault-tolerance theory because it maintains multiple consistent replications of the same application to overcome computer failures in minor replications. Although consensus uses extra computing resources for fault-tolerance, it has been widely adopted by industry in practice, because resource capacity is often not a bottleneck for mission-critical applications.

A second benefit of our methodology is that it is easy to make fault-tolerance itself robust. Distributed consensus is notoriously difficult to understand, build, or test. For instance, although some genius companies (e.g., Microsoft) have built consensus protocols for individual applications, ironically, recent research tools have detected numerous bugs in these protocols. Building one consensus protocol for each application could be a nightmare for application developers. Fortunately, with our methodology, state-of-the-art only needs to focus on testing our protocol and infrastructures, then many applications can enjoy robust fault-tolerance.

We envision significant impacts from this GAIA project in three terms.

In the near term, our fast and scalable protocol (Objective 1) can largely improve both the scale and performance of many replication applications running within a datacenter. For instance, Scatter [SOSP '11], a notable key-value store application, deploys up to 12 computers in each consensus group and lets each computer serve requests in parallel for high throughput. With our scalable protocol, now Scatter can deploy one or two orders of magnitudes more computers in each group and thus can achieve much higher throughput.

In the intermediate term, by realizing the new infrastructures in Objective 2 and 3, GAIA will greatly strengthen the reliability of general applications and benefit almost all computer users and software vendors. For instance, HK has many financial applications which naturally demand high availability in operational hours, and GAIA can meet this demand.

In the long term, we anticipate that this GAIA project will advance broad datacenter techniques (e.g., VM live migration) and attract researchers to build more reliable datacenter infrastructures. As datacenter emerges to be a "giant computer", a fast and reliable datacenter OS for such a novel computer will gradually come up. Therefore, reliable consensus protocols, schedulers, and virtual machines will become essential OS components, and the outcomes of this project will be adopted in a future datacenter OS.

Objectives:

1.

[To create a fast, scalable distributed consensus protocol].

We will create FALCON, a fast and scalable consensus protocol by leveraging the advanced Remote Direct Memory Access technique. We aim to make FALCON scale to hundreds or thousands of computers, so that many computers in a datacenter can join a consensus group and can enjoy strong fault-tolerance.

2.

[To construct a new datacenter scheduler for improving application availability].

We will construct TRIPOD, the first scheduler for application fault-tolerance, by integrating FALCON with popular schedulers. We aim to make TRIPOD seamlessly manage resource allocation and replication logic, so that it can efficiently support general applications.

3.

[To form a new VM-based eco-system for improving application availability].

We will make FALCON and popular VMs form a mutual-beneficial eco-system. This eco-system will leverage the VM hypervisor layer to achieve transparent replication for applications. It will also introduce a new lightweight VM live migration approach, so that computers can efficiently achieve balanced load without consuming much time or resource during VM migrations.