Scheduling Serverless Workloads

In serverless computing, short-lived functions are written by application developers, but are instantiated, scheduled, and run by a serverless provider. As such, the efficiency of a serverless framework depends in large part on the ability of the framework to instantiate and schedule a function with minimal overhead. This project seeks to explore the effect of three variants of a small optimization to the scheduling in the open-source serverless framework, OpenWhisk.

Enhancing Privacy in Federated Machine Learning

Data privacy is one of the major concerns of the modern era, on the other hand Machine Learning(ML) can not function without data. With growing tension around data breach and legislation like GDPR future scope for ML may narrow out over time. Secondly, ML requires a large quantity of data. These two challenges can be solved by incorporating Federated Learning with privacy and Edge Computing [6]. Edge computing aims to offload computation from a centralized cloud server to network devices that are closer to the source of the data which is a favourable way of handling data. As such an efficient mechanism is needed to share parameters as well as aggregate the training models with a centralized server. Here we investigate the use of enhancing the performance of federated ML on client data. One concern here is to protect the final model from inferring the true structure of the original data.

Serverless Edge Computing

Serverless architecture is a software design pattern where applications are hosted by a third-party service, eliminating the need for server software and hardware management by the developer. The applications are broken up into individual functions that can be invoked and scaled individually. Although the serverless design was for cloud environments, it is now gaining a foothold in the Edge Computing landscape, intending to bring computational resources closer to the data source. The advent of real-time and data-intensive applications empowered by mobile and Internet of Things (IoT) devices is heavily challenging the idea of centralized data centers. As part of this project, we are comparing and analysing two open source serverless platforms OpenWhisk and Kubeless. Also, we are trying to modify a few components on OpenWhisk to improve the existing implementation.

Security in Vehicular Cloud Computing

With the rise of new technologies like 5G communication, machine vision, and driver assistance and autonomous driving technologies, there is an opportunity to improve traffic management through vehicle-based networks by sharing information about conditions on the road. With this opportunity come security concerns, with potential vulnerabilities throughout the network stack, from the hardware to the network infrastructure to the applications and data hosted by the vehicles. Further, mitigations for these security concerns raise privacy concerns, as identification and attribution of attacks to particular actors or vehicles can come with risks to the privacy of other participants in the network. In this survey, we discuss the main security concerns for vehicle-based networks along with the most important considerations for addressing them. We discuss various systems for ensuring safety of data, security of communication, and protection of participants' privacy as well as the tradeoffs each system entails.

Performing Classification on Edge Nodes using Edge-Centric Distributed Deep Learning and Data Reduction

An immense number of IoT devices has led to an exponential growth of data moving across the network. Traditionally, Artificial Intelligence (especially Deep Learning, DL) tasks on the IoT data are usually run on the cloud or other centralized systems which leads to issues like network congestion and significantly higher latencies for real-time applications such as human activity recognition (HAR) or facial recognition. Thus, unleashing DL services using resources at the network edge near the data sources has emerged as a desirable solution. The latest research in this field is inclined towards performing Distributed Deep Learning on the edge nodes. In this paper, we aim to do inferencing on edge nodes via the compression and splitting of DNN model so as to fit on those resource constrained nodes and serve the real time task of HAR. One of the major obstacles encountered is that edge nodes have considerably smaller computational power compared to their cloud counterparts which act as a hindrance to perform heavy DL tasks.

DeRiS: Decentralized Ride Sharing

Ride sharing is a concept that has gained popularity over the past decade with drivers offering up free spaces in their vehicles to riders in need for a price. While this has helped reduce congestion and automobile pollution, current ride sharing systems use a centralized structure where a single entity orchestrates interactions between riders and drivers. This leads to a single point of failure. As a consequence, this centralized structure raises concerns about data privacy and is vulnerable to attack from malicious users. Furthermore, as the central entity takes a certain cut of the profit from drivers who bear the brunt of the costs of gas and vehicle wear and tear. The propose solution to this problem is DeRiS, (De)centralized (Ri)de (S)haring. DeRiS, as its name implies, is a decentralized ride sharing system based on the Ethereum Blockchain. DeRiS is an open source project and can be found at https://github.com/Nirvan66/DeRiS

Analyzing Differences in Real-life Performance for Decentralized Applications based on Holochain and Blockchain Models

The aim of this project was to compare the real-life performance of decentralized applications developed using the Holochain and Blockchain models on various metrics. We first reviewed the Blockchain model and reviewed research on its performance limitations and suggested architectural improvements, chief among which was the Holochain model as detailed in its whitepaper. We then implemented a decentralized chat application using both models that targeted supporting many simultaneous users and messages to compare their performance. Finally, we conducted benchmarks to test out various scenarios and collated our results.

FLAT-C: An approach for Text-Classification in a P2P Network using Federated Learning

Encrypted communications have become increasingly com- mon with the advent of chat services that support end-to-end encryption. Particularly in peer-to-peer chat systems like Tox [7] messages are encrypted using secure perfect forward se- crecy. In such a scenario using machine learning for tasks like spam classification become increasingly challenging as text messages cannot be sent in plain text and there is no central server that collects data to train models. Recent advances in Federated Learning algorithms have provided ways to train machine learning models on the edge, without having to trans- port training data to a server. In this project, we compare an approach for Federated Learning in a Peer-to-Peer architecture without a central server against a Centralized Federated Learn- ing Approach [3]. We overcome the difficulties of working with encrypted text, by using on-device machine learning and communication of local model updates to peers or the central the server, without breaching privacy or requiring transport of decrypted messages across the network.

Stateful platform with CRDT on the edge

We present a novel model for large scale mobile applications that are distributed in nature. With a number of innovations in Serverless at the CDN Edge (SAE), we aim to set up serverless edge devices. This would result in requests being routed to a mini data center or points of presence. The inherent low resource usage that Serverless offers is a perfect match for Edge Points of Presence(PoPs) small resource footprints. This set up has a host of advantages in terms of latency, bandwidth and robustness. AR/VR, Maps, Advertising are some domains that will truly benefit from such a set up.

AWS Lambda and Cloudflare provide SAE but it is stateless, making it useful for request/response services, effective load balancing but not much more. Coupling state with compute makes it useful for a variety of applications. However in moving from a centralized data-store into multiple decentralized data- stores that are replicated in a mesh, we are bound to have data- conflicts that will increase with the geographical distance between the decentralized nodes With data being replicated at these various sites. We aim to use CRDTs to address the complexities with adding state to the edge. CRDTs[link to article] allow PoPs to autonomously and concurrently modify data and then automatically resolve data-conflicts with zero consensus, which are the base requirements for a low latency SAE platform. We motivate this new model with a distributed TO-DO list for an application.

P2P Resource Pool

Peer-to-peer Computing has been one of the emerging areas of research for the last decade. Ever since the emergence of peer-to-peer file shar- ing systems, researchers have been trying to ex- tend this work to a more generic resource shar- ing abstraction. There have been considerable ef- forts to harness the benefit of peer-to-peer net- works; scalability, reliability and low-cost owner- ship - such as Avaki and Seti@home, which try to use large scale networks for compute resources. In this work, we aim to build a peer-to-peer resource sharing abstraction service that looks at resource layers of the peers in the systems and tries to pro- vide seamless resource sharing channels.

Distributed Computing of Stream Data

Some data naturally comes as a never-ending stream of events. To do batch processing, you need to store it, stop data collection at some time and process the data. Then you have to do the next batch and then worry about aggregating across multiple batches. In contrast, streaming handles never ending data streams gracefully and naturally. Sometimes data is huge and it is not even possible to store it. Stream processing lets you handle large fire horse style data and retain only useful bits. Hence stream processing can work with a lot less hardware than batch processing. Although there are many stream processing tools that are available in present, there are not many frameworks that support in processing video content, as this is starting to become one of the major sources of data. Lack of such frameworks, is not letting us mine useful information from these data sources. Also, today, these jobs often take hours, even for a short movie. Thus, we propose to build a highly scalable, high-performing, highly-reliable video processing framework.

Bounty Based Federated Learning

The goal of the project is to address two of the major problems in the machine learning field. One is the privacy and availability of the data and the other is compute power required for the training of models.Federated Machine Learning (FML) enables multiple parties to collaborate on building models while protecting data privacy for the participants. A measure of the contribution for each party in FML enables fair credits allocation. In this paper we study the advantages of FML over standard machine learning, explore the feasibility of rewarding the users participating in the learning process of the model using block chain by building a proof of concept(PoC) and propose a sustainable and scalable architecture from the findings of the PoC.

Enhanced Blockchain: Scalability and Privacy

A blockchain, is a continuously growing list of records, called blocks, which are linked and secured using cryptography techniques such as the Merkle tree. Each block typically contains a cryptographic hash of the previous block, a timestamp and transaction data. By design, a blockchain is inherently resistant to modification of the data. It is “a distributed ledger that can record transactions between two or more parties efficiently in a verifiable and permanent way". For use as a distributed ledger, a blockchain is typically managed by a peer-to-peer network collectively adhering to a protocol for validating new blocks. We are trying to enhance scalability and privacy to promote efficient blockchain management.

Decentralized DNS using Blockchain

The current DNS system has various problems which makes it a threat to security and failure. All of these concerns can be boiled down to a single problem, Centralization. This paper aims to propose and develop a secure, fail proof and decentralized system which can be used to provide IP resolution in a relatively improved speed than the existing system.

Blockchain Based E-voting

Most common electronic voting systems rely on a centralized server and database. Centralized systems are less robust than decentralized systems, for instance centralized systems are more vulnerable to malicious users, have a lower fault tolerance, and aren't as trusted by the users as decentralized systems. Unfortunately, there are very few functional decentralized voting systems. The ones which are functional are missing one or more key features voting systems rely on, such as ensuring voter eligibility, ensuring third parties can't modify voters votes, and ensuring a voters privacy is maintained. We attempt to implement a distributed voting system which could handle many of these features using blockchain technology.

Byzantine Chain Replication

Byzantine failures are defined as arbitrary, un- predictable failures of a process from its assumed behavior based on the algorithm it is supposed to execute. To address and mitigate the Byzantine failures, designing chain replication protocols is the way to go which detects and addresses the Byzantine failures. Chain Replication is a special primary backup system in which the servers are linearly ordered to form a chain with the primary at the end.

Distributed Systems: Smart Intersections

This paper provides the project proposal, related works, implemen- tation, and evaluation criteria for the CU Boulder CSCI 5673 Spring 2018 Distributed Systems course project. Our team hopes to develop an idea innovating in the space of virtual traffic lights and smart intersections. This will center around simulation and distributed consensus implementations.

Log Processing, Searching using Apache Kafka & Spark

Log Processing and searching a log stream is a challenge. The webservers of companies are bound to generate millions of logs per day and they get thousands of client read and write requests. In order to handle large number of client requests and handle high volumes of data, we propose using Apache Kafka, a distributed low latency PUBSUB log processor. In this project, the producer node will be a webserver written in node.js which connects to the broker cluster (we will be using vagrant to create the instances) and send its logs. Each producer will have his own topic (partition) on the cluster. On the consumer side, the one who subscribes to a particular topic will receive those particular logs.

Content Distribution Networks for Adapting to Latest Digital Media Standards

This project is about designing a Content Distribution Network topology, implement the same with Amazon's Web Services infrastructure and finally evaluate the system with regard to high-definition video streaming and content delivery. Uniqueness to this work is at the design wherein the inclination is to support and understand play of high definition content in varying bandwidth environment.

Blind Proxy Caches

The client/server model that the web is built upon has yet to take full advantage of the power of distributed computing. Scoped iterations in this field have proven to be most successful so we intend to provide a means to harness the performance gains of peer-to-peer content sharing. In this paper, we propose the blind proxy cache, a low-latency distributed cache that provides a “best effort” level of security and privacy initially targeted to organizations with displaced resources.

A Distributed Systems Approach to Phantom Traffic Jam Avoidance

In this paper, we take a distributed systems approach to implementing phantom traffic jam mitigation techniques. We evaluate the effects of introducing networked cars to a traffic model and using information sharing between the cars to automatically adjust speeds and maintain gaps between vehicles, thereby reducing the number of jams that occur. We build our model by using individual processes to run each networked vehicle and simulate vehicle to vehicle communication. We determine the effectiveness of vehicle decision making when the information received from other vehicles may not always be complete. Our traffic model uses a modified Nagel-Schreckenberg cellular automaton model which creates a baseline traffic model against which we can evaluate the effectiveness of introducing networked vehicles.

NRDSwap: Network RAM Disk Swapping

Current scientific applications are putting pressure on high-performance system designers to create compute clusters with enough resources to be able to finish the jobs. One of the most critical resources is memory, being difficult to create a node with enough memory to finish the work using the current huge datasets. NRDSwap provides more memory to a given node by creating a RAM disk in a remote node used as a swap space. By connecting the nodes using a fast link, such InfiniBand, the amount of memory in a given node can be arbitrarily large by paying a small impact on the performance.

Application Power Modeling for Android Smartphones

Understanding the energy consumption of mobile devices is a key area of interest for mobile distributed computing developers. Previous work has only been able to provide limited information concerning the energy consumption of individual applications because of limited access to under- lying hardware and system software. This project contains three parts to estimate the power consumption of each application: power model, status collection of hardware and kernel scheduler monitor. The estimation system should be able to determine the power usage of various hardware components in the device, which includes CPU, display, WiFi, GPS and other sensors. The status collection part will do the analysis of usage statistics and status changes of the hardware components. Then the accurate power model for hardware components should be available to determine the level of energy consumption for each component. The kernel scheduler monitor would determine which application is using the hardware components at a specific time.

Storm versus Spark: Streaming Cloud Technologies Compared

Cloud technologies have blossomed recently as a way for users to focus more on computation challenges, and less on the underlying hardware. With this comes a transition in focus to analytics of large datasets managed by Cloud technologies. As the number of available alternatives grow, it becomes necessary to compare the strengths and weaknesses of each, especially across varying applications. We perform a comparison of two recent additions to the Cloud analytics arena, namely Berkeley's Spark and Twitter's Storm. This project resolves the issues of establishing a side-by- side comparison between the Spark interactive query tool and the Storm stream-based tool. Additionally we provide analysis of strengths and weaknesses inherent in both tools. We use Hadoop’s MapReduce as a baseline for the comparison.

ByteFlow

Storing information in the cloud is a very widely used technique and the growth in its popularity shows no sign of stopping. Growing as well is the demand for systems that provide a secure data storage. Existing secure storage solutions rely on trusting the security and integrity of the service provider and we believe this to be a vulnerability. We present ByteFlow, a design and proof- of-concept implementation of a scalable remote encrypted filesystem. By encrypting data on the client side, we ensure privacy and secure even in the environment of an untrusted service provider. Our system Byteflow works with a broad variety of remote storage providers and supports multiple users through directory oriented key management and shows much promise for growth in the future.

Mobile Botnet

As far as we have been able to discern, there exist two basic types of mobile botnets: those in which the bots query some location directly for instructions and peer-to-peer systems where instructions are sent to some bots directly and propagated from there. While each has its advantages, we believe that a hybrid system has the potential to be yet more effective; our task is to design and develop such a hybrid system. This requires implementing a URL Flux scheme, implementing a peer to peer overlay network, and developing a system for determining which bots should participate in which actions. The efficacy of the botnet should be measured on its ability to remain loyal to the original botmasters, how likely it is to be detected, and its resistance to bot churn.

Smartphone Sensor-Based Context Detection

The use of smartphones that are equipped with different types of sensors became very popular in recent years. Examples for those sensors include the accelerometer, gyroscope, proximity sensor, ambient lights sensor, cameras, microphones, GPS and Compass [1]. The widespread use of such smartphones paved the way for applications that make smart on-the-fly decisions based on user context derived from collected sensor data. A major challenge for such applications is the accurate discovery of the user's context. Smartphones are subject diverse usage patterns within different environments. Such diversity will surely impact the accuracy of the decision taken by the applications due to noisy data collected from sensors. Our objective in this study is to identify the different common usage patterns and study their impact on the accuracy of the smartphone sensor based applications. Smartphones users will be deliberately put under different scenarios and sensor data will be collected and analyzed. We aim from this experiment to identify the impact of common usage scenarios\phone positions on the accuracy of the results.

Swarm Computing Using Smartphones

The attempts towards sharing computing resources amongst a group of networked computers have been numerous. The focus of these attempts have encountered limited success in general computing and only a few projects have been proved to be really useful, one such project is BOINC. The Berkeley Open Infrastructure for Network Computing (BOINC) is an open source middleware system for volunteer and grid computing. It was originally developed to support the SETI@home project. It became useful as a platform for other distributed applications in areas as diverse as mathematics, medicine, molecular biology, climatology, and astrophysics. The intent of BOINC is to make it possible for researchers to tap into the enormous processing power of personal computers around the world. Extending the idea to mobile we design a collaborative computing system which would enable all devices in wifi range to form a swarm and lend their idle computing cycles to any user in the swarm. Our project proposes modifications and extensions to existing distributed computing frameworks working in tandem with efficient data distribution protocols running over wireless networks. The peers are mobile devices in physical proximity or connected over a common wireless network.

Trusted Cloud Computing

Cloud Computing is an emerging technology which uses the internet and central remote servers to maintain data and applications. Cloud computing saves organizations money as it can be paid incrementally and it also increases mobility. But the clients of cloud computing services don't have means of verifying the confidentiality and integrity of their data. The paper addresses this problem with the design of a Trusted computing platform. This platform enables the clients to attest the presence of necessary security measures at the cloud provider machines. The cloud provider implements a Trusted computing platform which provides a closed box execution of the client Virtual machine on the service provider hosts.

Federated Storage API Across Cloud Based Services

Cloud storage providers allow individuals and organizations to access a lot of infrastructure that may otherwise be prohibitively expensive for users. For a variety of reasons, this is gaining popularity as a highly-scalable and cheap solution to storage. However, in organizations used to having their data stored locally and having full control of it, it is a scary thought to release it to an entity on the Internet and hope that it will be available when it is needed. Users also do not want to get locked into one provider in case a change is required at a later point. To address these issues, we propose a middleware service that will allow users to easily use multiple cloud-based services for data replication without having to learn each interface and worrying about data consistency.

GPS Assisted Adhoc Routing on Android Platforms for Poorly Connected Areas

Have you ever got frustrated with your phone signal strengths being quite low and not able to receive or make phone calls and especially when it is quite important? Sometimes we face this situation in our campus specifically in basement areas where some labs are located. We will exploit the Wi-Fi feature of smartphones to overcome the embarrassing situation of not being able to connect to cellular network. We plan to provide a solution using custom Wi-Fi routing protocol proposed in this project to help people in these situations to have continuous cellular connection.

Cloud Computing for Random Forests and Ensemble Methods

The success of machine learning has had profound effects on the technology industry, however, the full-fledged adoption of machine learning techniques has been hindered by seemingly slow model creation and evaluation for extremely large data sets. With multi-core machines becoming commonplace and the rise of cloud computing, machine learning can become much more feasible for large datasets if the problem can be stated in a manner which allows the data set to be split and distributed without a loss of precision. A team from Stanford University laid out a method for distributing machine learning tasks across a multi-core architecture using the Statistical Query Model. This method, however, only works for machine learning techniques when the problem requires significant statistical computation or gradients, as they can be expressed in a "summation form". There are several classes of machine learning techniques that cannot be written in that summation form, one that will be analyzed here is the method of Random Forests3. Random Forests are "a combination of tree predictors such that each tree depends on the values of a random vector sampled independently." Due to the random nature of the splits, the problem does not require statistical summation of the entire data set. However, since it is an ensemble technique, the problem lends itself very well to distributed computing in a cloud-like environment.

Mobile Peer-to-peer advertising

The ubiquity of mobile handsets presents a great opportunity to many service providers as they can reach a broader range of customers. This awareness has triggered the rise of relatively new concepts such as mobile advertising, a form of advertising via mobile phones. However, most mobile advertising solutions nowadays assume the presence of a cellular network, e.g. SMS advertising. Our project aims to break ground by proposing a system for a new type of service, namely, mobile peer to peer advertising. This will extend the possibilities of advertisers by allowing them to reach more people at low costs. Also, by carrying his/her phone with him/her, a private seller will be able reach a prospective customer in the vicinity, while the latter could find out that the service he/she is interested in is in a range of some meters.

Concentus: A Trust System for Fully Distributed Peer-to-Peer File Sharing Systems

Peer-to-peer systems, ranging from file sharing applications like BitTorrent to anonymous publication systems, have be- come increasingly popular in recent years. Proposed reasons for this have included scalability, decentralization, and anonymity. However, the ability of anyone to join many of these systems has caused security, privacy, and performance concerns. Peers may behave in ways that are either greedy or outright harmful to these systems. Concentus is an invitation-based peer-to-peer file sharing application that utilizes a reputation system to detect and neutralize malicious peers, as well as to reward people for contributing in a positive fashion.

CU Museum of Natural History Mobile Application

The main goal of this project is to develop a mobile application for the University Of Colorado Museum Of Natural History that provides information about areas of interest around the University of Colorado’s Boulder campus. The goal of this project is develop a distributed service that allows users to get information about these objects via their smartphones while they are walking in and around campus. Currently there is no mobile and easy to use system for learning about the history and the various landmarks around CU-Boulder's campus. Our project would give anybody the opportunity to tour and learn about the campus on their own simply by using their phone. The app would provide the user a means to easily find even the most secluded points of interest around CU. The application will include ways for administrators to add, modify, and remove content that is displayed on the app.

A Diverse Measurement Study of Mesh Networks for Challenging Environments

Wireless mesh networks and the protocols they leverage are a topic of great interest in networking today. Their diverse utility, relatively low cost, and speed of implementation make them ideal as a form of disaster relief and recovery communications infrastructure. We conduct a measurement study of real-world applications across a diverse hardware test bed simulating the features of challenging network environments. Specifically, we extend upon previous work in two key approaches. First, we wish to expand the node diversity and network diameter to include smart phone devices, tablets, and introduce mobility as a measurement metric. Second, we simulate austere, terrain restrictive, and degraded link reliability and/or throughput to create a real-world study for the purpose of disaster relief/recovery operations.

Performance Optimization in a Migrating Cloud System

With the advent of cloud computing, clients of the cloud have focused on how to gain the greatest amount of computing power for the least cost. This has led to the creation of systems for the execution of efficient computations as well as autoscaling programs that determine when an application needs to increase its resources based on its current load. This paper approaches this same problem from a different perspective, it focuses on determining the optimal placement of a computationally intensive application across data centers in order to experience the highest performance of that application. Therefore, a migrating cloud system is proposed, where an application is moved between different data centers at different times of the day based on the known performance of that data center at a given time. This system helps to increase the computing performance of an application by moving it to a data center that is able to provide it with the best computing resources.