SIBR Grant Application

Project Summary

Overview

Keywords: Data Denormalization, Stream Processing, Distributed Databases, Search Indexes, Bulk Processing

Modern data intensive software stacks consist of much more than a relational database. EZstack represents a new, innovative way to think about data denormalization. The goal of the project is to provide a highly scalable, highly efficient, eventually consistent data stack. This product is designed to benefit big-data companies in achieving an efficient, highly scalable data processing system with a simple interface, thus eliminating engineering difficulties within the data model. While most other big data stacks attempt to tackle these problems by investing large amounts of resources in developing their own custom solutions that are highly specific to their case, EZstack will attempt to provide a general solution that is applicable to many different use-cases, favoring ease-of-use and customization.

Intellectual Merit

This Small Business Innovation Research Phase I project strives to solve major challenges related to big-data systems, while simplifying their usage. High performance in this generic system will be achieved by leveraging a streaming data architecture. EZstack will have two major distributed datastores, a write optimized system of record and a search index. In between these two datastores will be a distributed denormalizer that transforms the normalized data in the system of record into more searchable formats. The transformations performed by the denormalizer will be determined by advanced query analysis. Additionally, EZstack will be extremely straightforward to deploy. Administrators will simply install an agent onto every node and EZstack will manage itself.

Broader Commercial Impact

Educators, government agencies, large corporations, and fast-growing startups have the same problem: They have a need to create connections between datasets in order to enhance our environment, provide useful statistics, or increase user connections. Big data is a large topic that has many issues that need to be tackled. Currently, there are no systems that can provide the ease-of-use, scalability, and performance boost that EZstack aims to achieve. EZstack should provide a plug-and-play solution for applications with many different data models, allowing engineering teams to easily and quickly deploy to production. If this project is successful, it will lower the barrier to entry of educators, government agencies and many other entities to develop data intensive applications.

Elevator Pitch

The Market Space

EZstack will have a great amount of use for many different groups of people. Smaller software development teams that would normally not have the financial capabilities required to have access to big data management are very handcuffed in their capabilities. Access to big data management systems can be extremely beneficial for companies in almost every field, and currently it is often far too expensive for companies that are not large corporations to develop systems that can efficiently manage data. Additionally, large companies that are capable of creating big data systems must spend large amounts of resources on those systems. Additionally, if a system needs to be altered to match new data, significant new costs will often be incurred. The goal of EZstack is to address both of these problems with one system that is both easily deployable and economical for all customers, regardless of financial means.

The Value Proposition

Currently, there are no systems that can provide the ease-of-use, horizontal scalability, and performance that EZstack will achieve. EZstack should provide a plug-and-play solution for applications with many different data models, allowing engineering teams to easily and quickly deploy to production. If this project is successful, it will lower the barrier to entry of startup companies, research teams. and many other entities to develop data intensive applications. There is a significant need to create connections between large datasets in order to enhance our environment, provide useful statistics, and enrich user experiences. Big data is an expansive field with many issues that need to be tackled, and EZstack will do exactly that. The overall achievements of the EZstack system are guaranteed efficiency for both reads and writes, horizontal scalability, and simple deployability without the significant costs that such a system would normally require.

The Innovation

EZstack strives to solve major challenges related to big-data systems, while simplifying their usage. High performance in this generic system will be achieved by leveraging a streaming data architecture. EZstack will have two major distributed datastores, a write-optimized system of record and a read-optimized search index. In between these two datastores will be a distributed denormalizer that transforms the normalized data in the system of record into more searchable formats. The transformations performed by the denormalizer will be determined by advanced query analysis. Additionally, EZstack will be extremely straightforward to deploy. Administrators will simply install an agent onto every node and EZstack will manage itself.

Commercial Opportunity

EZstack has an extremely vast potential user base. Potential users include large corporations, fast-growing startups, government agencies, researchers, and hobbyists. The product is engineered to work for customers at any scale, whether that be on a single server, or on thousands of servers. The purpose of EZstack is to provide a scalable backend for any data intensive application. Creating a data intensive backend is extremely expensive and time consuming to engineer, maintain, and deploy. This difficulty is precisely the problem that EZstack will solve, and it is a strong market driver for the software.

The business model for EZstack is very similar to those of Red Hat, MongoDB, and Elasticsearch. The software itself will be free and open source. This means that the public at large will be able to use the software free of charge. Additionally, the underlying source code of EZstack will be publicly available, which allows anybody to contribute modifications and improvements to the project. The developers of EZstack will monetize the software via two main revenue streams. The first main revenue stream is the sale of service and support to the more affluent users of the platform. Service and support is essential for enterprise customers of large software, as they require guarantees that they will be able to meet their service level agreements (SLA's). The second source of revenue for EZstack is the sale of the software as a fully hosted service. This means that the developers behind EZstack will completely manage the deployment of the software for customers that wish to use it, but don't wish to be responsible for maintaining it.

EZstack is an entirely managed backend as a service (BaaS) software. With entirely managed services there are many increased risks that are transferred from the operations team deploying the system to the developers developing the service. In the field of managed services, customers are to assume that the boxes deploying the service are black boxes that are well secured and are accurate in their execution. Due to this, during development significant ops work must be integrated into the product to enable secure deployment and access to data. Encryption, managed access, and role validation have to be build into EZstack from the outset, otherwise it will be very impractical for EZstack to guarantee secure usage of its services.

EZstack also has to account for industry standard security tools used by many enterprise customers that would be pursued. EZstack must have plugin support for systems such as Splunk and Q-Radar to ensure internal visibility and alerting can be integrated in an enterprise security management model. EZstack also has other risks relating to reliability, uptime, and data integrity. EZstack's managed service must have clear and high Service Level Agreements (SLA) to ensure the users will be focusing on developing their best product without having to worry about the underlying infrastructure.

EZstack is a unique product that is integrating many open source projects under the hood, which significantly increases the difficulties of ensuring data flow throughout the system. EZstack's system is using open source projects such as Kafka, Samza, Elasticsearch, Cassandra, various app servers, and many other services. Those services have to have a reliable and transparent communication system and data integrity needs to be verified at every stage of the system. This project takes on risks similar to those taken by cloud providers and Software as a Service (SaaS) companies; however reaping the many benefits cloud providers such as Azure and AWS have would be worth the risks accompanied by this project.

Societal and Global Impact

There is a very pervasive problem with data intensive applications: their backends are very complex, to the point where engineering them can become nearly impossible. This causes many potentially amazing applications to be very expensive to deploy and maintain. Those extreme expenses make it prohibitive for groups of people that do not have sufficient funds to break into the global software market. Even though large corporations have the capital required to engineer data intensive applications, there still exists many major problems that they cannot avoid. The most significant of these problems is that many of the existing systems are inflexible. It is nearly impossible to make alterations to an existing stack because most major alterations require the system to be almost entirely re-engineered, which is extremely expensive.

EZstack will provide a way of avoiding all of the aforementioned issues. It will enable smaller teams that had no previous method of developing data intensive applications an opportunity to do so, as well as allow large companies to be able to deploy and maintain systems that they otherwise would have had to spend large amounts of resources on. The system is designed to be plug-and-play, so it is usable by *anyone*. Additionally, EZstack is built to run at truly massive scale, so there will be no need to re-engineer the system to scale up.

The immediate beneficiaries from EZstack are engineers. With EZstack, the systems that engineers will be able to create will provide significantly more flexibility and scalability at a fraction of the cost of existing backends. In addition, EZstack will be far less complex to maintain, so the IT personnel tasked with assembling and maintaining systems built on top of it will benefit greatly. The agony of engineering case specific systems from the ground up will be over with the introduction of EZstack. EZstack should be a transformational innovation in the field of software engineering.

Although the most direct impact of EZstack is on those who will be implementing data intensive application, the greatest impact of the project is on the users of the applications created using EZstack. Because applications built on EZstack will be extremely simple to engineer, developers should have significantly more time to develop new products and new features, which ultimately benefit users. This impact will be global, as EZstack has no technical limitations.

The ethical and environmental implications of EZstack are relatively minimal. The existence of EZstack does not directly affect the core functionalities of the applications it powers, therefore it is difficult to assign any ethical or environmental accountability to it. Like most open source technologies, EZstack can indeed be used for unethical purposes. However, the culpability for such unethical uses lies on the engineers and the applications they create, not on EZstack itself. For this reason, EZstack should not require any regulation.

EZstack will be providing an avenue for all groups of people to have access to a scalable software stack for data intensive applications; no longer will teams be limited by a small budget. This will lower the barrier to entry for many start-ups and small businesses, enabling more people to contribute great things to society. This will have a disparate impact on many different sectors of the economy. The virtue of a flexible system is that it can apply to anyone and everyone; there are no limitations on what can be built with EZstack. The technology used to avoid re-engineering costs is the same technology that enables the system to be used in a near-infinite amount of use cases. The usefulness of EZstack is not limited to domestic users. Many international governments, corporations, and research groups have the same needs as American entities, and EZstack will help solve their problems as well.

If widely adopted, it should be very easy for people to create connections between massive datasets in order to enhance the environment, provide useful statistics, or increase user connections. There are no existing systems that can comparably accomplish this. The utilization of EZstack could facilitate scores of new disruptive technologies.

Technical Discussion and R&D Plan

Innovations

There are a few technical innovations being introduced in EZstack. The largest of these is streaming denormalizations. Streaming denormalizations is the ability to pass the normalized datastore through a denormalization engine, and convert the data on the fly into a more queryable format, which will greatly increase query speeds on the other end. The way this is being achieved is that the denormalization engine will be run on top of Apache Samza, which is a distributed stream processing framework. The engine will be passed all of the writes made to the system, which will be streamed to the denormalization engine using Apache Kafka, which is a stream processing platform. Once the denormalizations have been made to the data, it will then be streamed into the denormalized search engine through another Kafka stream. The denormalized search engine will be Elasticsearch.

This system will be operating as an "eventually consistent" system. This means that if a user makes a write to the datastore, the write will be picked up by the Kafka stream and they can confirm that the write was processed, which will be very fast. From this point, the data will need to be processed before it becomes available in elasticsearch, which will take a few seconds, but at that point in time it will then be accessible to reads in an efficient manner. This is a very clear tradeoff being made here, but it certainly holds many advantages that makes the system very desirable for many different types of users.

Another innovation being introduced in EZstack is automatic rule generation, which informs the denormalizations made to increase query speed in EZstack. The automatic rule generation system will exist within a query aggregation system, which will be running on another Samza instance, separate to that of the denormalization engine. The query aggregation system will be given a stream of the queries made by the user applications to the Elasticsearch search engine. This stream will also run on Kafka.

Once the queries are picked up in the query aggregation system, they will be picked apart to determine a few things: What types of denormalizations could be made - if any - to speed up the query, and how long the queries took versus how often they occurred. These factors will determine the prioritization of each denormalization to be made on the data, and the information will be updated over time to reflect how the data changes over time. This way, the system will always maintain a sufficient level of optimization in being designed to run the most common queries at a faster speed than those that happen much less frequently, as well as speed up queries that took much longer to run before their data was changed into a denormalized format.

The rules determined by the query aggregation system will be sent back to the RESTful app servers that the queries were sent from, and from there will be sent to Zookeeper, which is a distributed hierarchical key-value store, which is being used to manage the different components of the project. Once there, the denormalization engine will be able to pull down the newly updated rules, which it can convert into new denormalizations to make on the normalized datastore. These updates will happen sporadically as the frequencies of certain common queries changes, and after the denormalization engine runs through all of the most recent writes for each document in the datastore, the system will be completely denormalized over the new rules, and will be operating at the desired efficiency.

Phase 1 Project Research

Phase 1 research of the project has some specific goals and interests that will be very important to the completion of the project. Technical feasibility isn't in question, however it will be important to be able to determine what tools are necessary to complete the project. The list of necessary investigations includes learning about the usability of potential datastores. The primary interest is in Elasticsearch, which will be the search engine used for the denormalized datastore. The reason Elasticsearch has been designated as such is because of its capabilities as a distributed, multitenant-capable full-text search engine. It is also open source, and usable under the terms of the Apache License, which makes it easily the most desirable candidate for this job.

Another technology that will be necessary to research is Apache Kafka. Kafka will play an essential role in the project, as it will be basis for the data pipeline that moves all sort of data and information around the different components of the project. Kafka benefits from the same open-source and usability as Elasticsearch, as it is also usable under the Apache License. Kafka is the clear choice for our stream processing platform, as it provides a unified, high-throughput, low-latency platform for handling real-time data feeds. In addition to this, it also is massively scalable as a pub/sub message queue architected as a distributed transaction log, and this means that it is extremely valuable for processing streaming data, which is among the biggest concerns of the technical feasibilities of our project.

As previously mentioned in the Innovations section of Technical Discussion, Apache Samza is another technology that will be utilized by EZstack. The benefits given by Samza are highly similar to Kafka, because the framework was developed in conjunction with Kafka, so the two working well together will be of great benefit to the project. Samza fulfills the necessary hole of an asynchronous computational framework to work alongside the processing platform that Kafka provides.

On top of the technological investigations, there is research to be done on some algorithmic questions in the project. There are multiple ways of determining denormalization efficiency, and it will be necessary to have answered these questions so that designing a system that can correctly prioritize certain denormalizations is possible to implement. Among these questions are what types of response time feedback should be considered. Simply taking the average response time for individual queries may not be sufficient, as it will also be of some importance to look at specific percentiles to see how fast a majority of certain queries run, as opposed to the ones in the middle. There are many ways to go about accomplishing this, but it is imperative to have one in mind as the project moves forward into the developmental phase.

There is no legitimate concern in the commercial feasibility of the project. There is a large commercial opportunity for EZstack to take advantage of, as no current system is able to provide the efficiency and flexibility that EZstack is able to provide. The level of novelty in the design is such that there will be no issues in finding both large customers interested in using EZstack as a backend as a service (BaaS) software, and small users who are simply interested in taking advantage of the system for their individual use. The smaller users are of no concern, and it is the larger interested entities that provide the commercial value, as they will be interested in paying for a service that would provide constant support to the system in addition to simply using the system itself.

Critical Milestones

The project's milestones are quite well defined for the beginning of the project's development. There are certain objectives that simply must be met for everything to get off the ground and running in a sufficient manner. Early on, there will be a lot of testing that will go into using the different technologies involved in the project. Simply getting different technologies to work in the necessary manners will take some amount of work, and although not the most difficult area of the project, it still is one of the most important milestones of the project to have individual technologies working on their own.

The first large goal of the project is to have a very simplistic working version of EZstack, where the parts of the project are put together, with the only pieces not included being the ones that add the significant complexity to the project. This means that the project at this point in time would essentially consist of a pipeline framework capable of moving written data through Kafka streams and Samza jobs into an Elasticsearch engine, where it is possible to manually query the objects. This will be a very significant challenge, because despite the pieces being able to work flawlessly on their own, getting them to work well together will require a large amount of work. Among the problems involved will be deciding on the versioning of each platform, which will dictate what work is necessary to put the pieces together in the first place. However, this will also involve determining what external technologies will be necessary to get everything working together as well. This means a large investigation into Apache Zookeeper, as this will be the technology used to get the different components of the project coordinated together, and give them the ability to communicate with each other efficiently.

The working version of the project in this state will not have the efficiency that EZstack strives to achieve. Instead, it will be more of a proof of concept that the system is capable of functioning at a basic level, and from that point on, the goal will be to improve on the designs to the point where we can make progress towards having a product that is actually desirable.

Once the project is at a state where there is a working pipeline, the goal will be to iterate over the project, adding the required complexities to give the project its intelligence. This means the next major milestone will be having a basic working query aggregation system that can successfully determine and communicate specific rules that need to be made by the denormalization engine, based on user queries sent by the RESTful app server to the query aggregation system. The intention for this milestone would be to have a system that provides a satisfactory level of efficiency to the point where it can be determined that the project is successful, but isn't to the point where the project has been finalized, as there will still be improvements that are possible to be made in a short-term time period. The working version of this will have at least a level of efficiency that EZstack guarantees for the final project, with the capability of further improvement in the future.

The final milestone is less concrete than previous iterations of the project. At this point, the algorithms and systems will simply be improved endlessly, with the goal of improving the efficiencies and response times of all aspects of the project. There is no endgame here, as there is no such thing as a “perfect” algorithm, and as such the system will be able to be improved infinitely. However, each improvement will likely have diminishing returns over time, as it will become more and more difficult to find things to alter in the build, and it will be up to the developers of EZstack to decide at what point the project can be initially considered complete. The working version of this final milestone will just be a more efficient and effective version of the previous milestone, and it isn’t clear what exactly this means, but what *is* clear is that even after the project is considered complete there will still be potential improvements that would make EZstack even better than it is on launch.

R&D Plan

The initial plans for research and development as outlined earlier in the Critical Milestones section will be primarily composed of researching the necessary technologies to the point where they are understood well enough to be run in the exact ways that are necessary for the project to be successful. This means that in the early months of research, it will be imperative to be able to set up Kafka streams that can process streaming data, as well as also set up Samza jobs that can work initially as computational frameworks on their own, independently of one another. It will also be important to set up Elasticsearch so that its capabilities as a search engine are better understood, so that it can be better utilized when the time comes.

The following month of development will be altering these systems to work as we need them, and this may be the most significant part of the project. There is a lot of specification that is required for everything to fit together, and it will take a lot of effort to be able to use these technologies in conjunction with one another successfully. The outline for this month will need to take into consideration the amount of time necessary to do research on the technologies required to enable this coordination, which will primarily be backed by Apache Zookeeper. Setting this up to be able to coordinate everything in the necessary manners will be time consuming, and will more than likely take up the better part of the month.

The important developments made in the time following the completion of an initial working pipeline will be heavily theoretical. A lot of mathematical and algorithmic problems will need to be solved, and it will be necessary to iterate through multiple versions of any query aggregation algorithms that are produced so that a working version that is fast and efficient can be introduced to the query aggregation system for producing new denormalizations to make on the datasets.

At this point, everything done in the remaining time until the project deadline will be iterative, and the goal will only be to improve on existing designs. There is no specific goal to hit in a technical sense, but there will never be a lack of work to be done on the project until the deadline hits. This means that the physical goal to aim for in the remaining time will be to make EZstack work as efficiently as possible before the project deadline in May.