Immense

Real-time scalable fault-tolerant distributed system

Joseph Choi

I am an undergraduate at the California Institute of Technology, in Computer Science. In my dorm room, I have independently developed a prototype of a real-time scalable fault-tolerant distributed system, called Immense. It has the potential for being the main product in a new high tech startup company.

There are numerous cases in the Internet industry as well as the rapidly emerging mobile industry where there is demand for a highly scalable infrastructure. Typically when a new entrepreneur is seeking to create a startup based on a new mobile app for sharing messages with friends in an innovative new way, they will quickly build a prototype that is unscalable. If the mobile application catches on and becomes popular, then they have the possibility of exponential growth. At the advent of the mobile era, more people have devices and the world is more connected than ever before. Hence from now on, exponential growth will only be even more extreme.

There is the important distinction between the front-end and the back-end. By front-end, I mean the mobile applications that you actually interact with on your mobile device by clicking, typing, viewing, etc. By back-end, I mean the complex system of computers and databases that store user data/photos and transfer them between clients. The back-end is technically much more difficult and complicated than the front-end, and is one of the more time-consuming aspects of developing a product. Since a lot of startups in the mobile application space need to test their idea before investing lots of resources and time into developing a more fully-fleshed product, typically the

back-ends for these prototypes are not scalable at all. Typically, unscalable code can only support 1,000-10,000 users simultaneously. Once a service gets popular, the challenge is scaling the system to be able to support more users. This is not as easy as just adding more back-end servers. Almost all of the time for a first version that is unscalable, *almost every aspect of the codebase will need to be rewritten* to support 1,000,000 or 100,000,000 users. What is even more of a problem for quickly growth is that the service has to be continuously running while nearly everything has to be redone to support huge numbers of users. An analogy is trying to replace the engine of a car *while it's being driven*.

Most startups don't account for scalability at the beginning because it is very time-consuming to build (often by a factor of 10 or more), and they want to see whether a product will become popular. When it does, and the system is not scalable, the startup will experience extreme growing pains.

Immense can solve a lot of these issues from the get-go. It is a set of technologies that can be extremely easily be used in the development of nearly any kind of Internet/mobile back-end systems. Part of the reason most startups create unscalable products is because it is difficult and time-consuming to make scalable products, and it might not be worth the time and resources to simply test an idea. With the system I have developed, a set of developers can easily create a massively scalable system even just to test an idea! Thus even the first prototype of a startup's new mobile application can be as scalable as a large company's products. Thus when a startup's new mobile application unexpectedly becomes popular, they don't have to rewrite the nearly the entire system from scratch to make it scalable, because it already is scalable.

Making a system scalable is expensive in terms of developer time. It might take a team of developers months to make a scalable system. If the developers are not the founders, then they will have to be paid, so this is expensive in that dimension as well. The codebase for Immense already provides a real-time scalable fault-tolerant distributed system, so a lot of what goes into making a service scalable has already been done.

One of the key characteristics of this technology is that it is real-time. This means that when messages are transferred between clients (browsers, mobile devices, etc.), it is done so nearly instantaneously. This can be used to accelerate things like posting on social networks (people see what you post immediately, rather than 10 seconds later). Another hypothetical use case is doctors using robotics to treat a patient that is far away. In Twitter, you don't see new posts from the people you are following instantly. Oftentimes, it takes 1-3 minutes to receive a new post made by someone you're following. Posts from celebrities, who can have upwards of 10,000,000 followers, have been known to take up to 5 minutes to propagate to their followers. Immense, on the other hand, can support these numbers as well, but instead of minutes, it is on the order of *seconds*. New startups can take advantage of the power and speed of Immense in developing innovative new applications.

Another aspect of the technology is it is distributed. The system is extremely scalable, the back-end technology is advanced enough that it can support up to 100,000,000 users simultaneously. For example this could be used to have a virtual crowd on the cloud, where millions of people can interact with millions of other people simultaneously, also in real-time which alludes to the feature above. There are lots of distributed systems out there, for instance Amazon Web Services and Google Compute

Engine. These are cloud back-end service that has automatic capacity scaling and scalable deployment. However, this serves a different purpose from Immense. AWS and GCE provide scalable *hardware*. Immense provides scalable *software*. One can't really take advantage of scalable *hardware* without scalable *software*.

The technology is also fault-tolerant. Having a fault-tolerant system is important. What if there is a power outage at one of the data centers? Will users suddenly be unable to communicate with each other? Fault-tolerant systems are designed to keep on going even when there may be power outages, by distributing the system throughout the continent. It is highly improbable that all these servers would go down at once. There is also the property of a self-healing network, where the system can identify where it is missing power, and create additional instances on the fly in order to "heal". This is another area where early startups struggle in making their systems scalable, and Immense automatically takes care of this.

The technology is integrable. This system is easy enough to incorporate into brand new mobile application back-ends, even just to test an idea with a prototype. Startups can develop scalable applications from the get-go!

The model for providing the technology would likely be licensing, due to many reasons. How this would work is that the technology is provided as a codebase that could directly be incorporated into a new codebase for building a web-based mobile application. When a startup starts developing a new mobile application, they would build on top of this codebase. The technology could be licensed in a number of ways, depending on the structure of the organization that is incorporating this technology. For small companies, one could have an annual license per developer or a site-based license, which is typical of most software based systems in the business.

The technology would need to be known by people who are planning on starting to build a mobile application. Most startup developers before having venture capital funds have little money at the outset, so the price of the license cannot be too high. In fact, the majority of the revenue that Immense might generate would come from startups whose products have become really popular and so are making heavy use of the scalability features of Immense. Hence having the scalable technology Immense to be completely free for startups below a certain usage limit would be beneficial to the success and widespread use of Immense. This is good for many reasons. It provides aspiring startup developers and entrepreneurs without extensive knowledge in building real-time distributed fault-tolerant scalable systems, but with great ideas to not have to focus on the technical aspects of scalability as much. Not only is this good for Immense, it is also good for the world in reducing the barrier to creating innovative new Internet and mobile applications.

Current systems that are similar to Immense are Pusher and PubNub. The difference is that these other services require you to go through their own servers along with your servers. This adds another round-trip time to transfer the data in your applications from your servers to theirs. For a system that is real-time, this potentially adds a lot of latency to the service. Furthermore, if a startup is using Amazon Web Services for the physical servers that are needed to serve your mobile application, since Amazon needs to make money, the startup is paying this price. Pusher and PubNub are additional services that rely on Amazon Web Services. Since they are paying Amazon and they themselves have to make money too, there will be even more of a cost to the startup that uses these services, while Amazon is getting money from both parties. Immense would serve as a licensed set of libraries that startup companies would

incorporate into their code. By doing so, they would only be paying Amazon to run the code, while the licensors are not running the code as a service on their own Amazon servers. Hence, not only is the real-time distributed system faster since it eliminates the additional round-trip, it is also more price effective since the overhead of a separate service having to pay Amazon and charging even more to cover the costs is eliminated.

This is an interesting opportunity that eases the problems of scalability for startups as well as a possible business for licensing the codebase. There is no denying how useful this technology would be, as a scalable architecture is the basis for such prominent Internet/mobile services as Google, Facebook, and Twitter. With growing scalability demands from new startups entering the scene, including Snapchat, Secret, and Whisper, as well as the growth of mobile in the world, demand for scalability will only continue to increase as we move further into a world connected by web technologies. It is clear that further work should be done on improving Immense, as it may be converted into a valuable business opportunity in a world with increasing demands on scalability.

About

Joseph Choi is an undergraduate at the California Institute of Technology, in Computer Science. He has experience in big data, user interface design, scalability, real-time applications, and distributed fault-tolerant systems, most of which was learned outside of the classroom. He has interned at Google two times on the System Infrastructure and Gmail Load Balancing Teams, and has received offers from various top tech companies. He is working on developing new technological infrastructure this summer.