GOTO 2016: Uber

A lo largo de esta charla el cofundador de Uber Matt Ranney nos habla sobre la escalación de Uber y de lo rápido que cambian las tecnologías y Administraciones de los servicios. Hoy en día Uber se encuentra presente en más de 400 ciudades, 70 países y a llegado a tener más de 6000 empleados, debido a que su crecimiento se ha multiplicado por 10 en el último año y medio.

las lecciones de vida que ha aprendido a lo largo de este crecimiento exponencial son la clave para esta charla, haciendo énfasis en la dificultad que genera el desarrollo de una arquitectura de micro servicios debido a la gran cantidad de grupos de trabajo y lenguajes que se emplean en todo el proceso.

Para introducir a los micro servicios se menciona que los sistemas funcionan mejor cuando no están en constante cambio, pero estos cambios son importantes y deben realizarse para mejorar el funcionamiento del sistema en general. Se plantea entonces la pregunta de ¿porqué usar microservicios?, la capacidad de mover independientemente las partes del sistema, poder organizar adecuadamente los tiempos y poder implementar diferentes herramientas según sea conveniente dentro del mismo sistema es la respuesta adecuada a esta pregunta.

En oposición encontramos los costos que se generan al usar microservicios, encontramos que los sistemas distribuidos pueden generar inconsistencias que serán difíciles de corregir más adelante, todo se encuentra en un RPC, lo que puede generar que si el sistema se cayera, toda la red dejaría de funcionar adecuadamente y habría una pérdida mayor del servicio por ese plazo de tiempo. El expositor también menciona costos menos obvios como la existencia de una compensación para todo, si bien puedes arreglar un problema, siempre habrá una reacción, además se puede construir sobre errores buscando taparlos, generando un error permanente en el sistema; la complejidad de las políticas del negocio también se pueden ver influenciadas y se generan prejuicios muy difíciles de romper, debido a las preferencias de los trabajadores y los grupos que se forman naturalmente dentro de la cultura de la organización.

Se sabe entonces que si bien la posibilidad de usar diversos lenguajes de programación según sea necesario puede ser un beneficio para el desarrollo, también es una desventaja para la administración de todos estos equipos de desarrollo. Los costos que se generarán incluyen la dificultad para compartir código y la fragmentación de la cultura,como se mencionó anteriormente, pues se forman opiniones y grupos relativos al lenguaje y frameworks que se emplean por cada servicio.

Entonces, como habíamos mencionado previamente, todo se convierte en un RPC generando otras complicaciones relativas a http/rest, la necesidad de generar un esquema general para json, la lentitud frente a pcs y la confusión de un servidor con un buscador.

Pasando a otro tema,se habla de la cantidad de repositorios necesarios y se menciona que uno sería una buena opción, pero en la práctica Uber posee más de 4000 repositorios y sigue en aumento. También, por el lado operacional la existencia de grupos y diversos métodos de trabajo dificulta la ayuda de otros grupos, pues difícil ver el sistema como un todo, cuando se encuentra dividido en tantas partes más pequeñas.

Por el lado de la performance se depende de las herramientas del lenguaje que son difíciles de medir cuando estas son tan diferentes, por lo cual se buscó una solución más estándar, proponiendo un dashboard general que todas las áreas debían utilizar para obtener una comparación adecuada del desempeño a nivel de toda la empresa. Del mismo modo se requiere de simple perf requirements para un mayor conocimiento de las actividades y del negocio en general.

Encontramos también casos de lentitud del sistema causados por razones muy obvias o de difícil descubrimiento, en cualquier caso el empleo de tracings permiten encontrar la mejor forma de entender el fanout, pues puede llegara a ocurrir la existencia de fanout sin intención en que se cree que el código se encuentra bien desarrollado. En casos donde el problema es de mayores dimensiones se requerirá de un muestreo para encontrar causas probables y evitan que estos fallos puedan propagarse a otras secciones del programa.

En relación a logging hay que tener en cuenta la necesidad de consistencia y estructuración , pues al existir tanto lenguajes de programación este proceso crece en complejidad y los loggings floods pueden aumentar este problema. Del mismo modo las pruebas (load testing) requieren una prueba antes de salir a producción, lo cual en el contexto de Uber es difícil debido a la permanencia del sistema en lima, y se ha optado por probar el producto en los momentos con un menor flujo de data. De manera opuesta se menciona que los failure testing si bien no son del agrado de la gente, son necesarios para el correcto funcionamiento del proceso y del equipo.

La migración a la arquitectura de microservicios es sin duda un problema, pues mientras que se encuentra en desarrollo los anteriores procesos aún tienen que trabajar y las partes inmutables del sistema serán las que requerirán un mayor número de cambios generando un proceso tedioso de cambio que si bien es difícil no es imposible, pues uber es una prueba de ello. Al momento de cambiar de sistema también se debe tener en cuenta que programas emplear, los open source son una opción viable, pero también lo es construir tu propio programa, creando una disyuntiva con pros y contras por ambos lados. Tampoco hay que dejar de lado que la utilización de los sistemas libres para la generación de dinero puede generar un descontento de un sector del público y una mala imagen.

Finalmente se menciona que las servicios permiten a las personas jugar con las políticas de la empresa y generar una organización de compañía - equipo - persona; además se recalca que todo tiene un precio, utilizar una solución en vez de otra siempre tendrá una consecuencia, pero es preferible ser consciente de estas consecuencias y hacerlas de manera intencionada, de forma que se conozca el alcance de las decisiones que se toman y todos las posibles repercusiones que se podrían generar.