

La mensajería fue posible de implementar tanto a través de RPC como de Mensajes Asincrónicos, pero entre estos hay varias diferencias.

La primera diferencia notable entre estos dos métodos es la forma de implementarlos, es decir, como codificarlos son muy distintos, ya que al implementarlo con RPC hay que seguir formas mucho mas estrictas de programar, hay que saber cuales son las estructuras de los mensajes a enviar, los nombres de las funciones específicas, que atributos tienen estos mensajes, etc. Mientras tanto por el lado de los Mensajes Asincrónicos, era muy flexible no era necesario saber estructuras determinadas o cosas por el estilo, pero esto a su vez tenía sus desventajas, ya que habían aspectos del RPC que facilitaban enormemente el traspaso de información, como por ejemplo mandar una lista, en el caso del RPC solamente eran dos o tres líneas, solamente con yield se podía “streamear” la lista, mientras que en el caso de los mensajes era un poco más tedioso.

Otra diferencia notable, que tiene mas que ver con la naturaleza de los métodos usados en sí, es la durabilidad de los mensajes, es decir, en el caso del RPC si el cliente/servidor envía un mensaje y la contraparte sufre un problema de internet o el computador se paga o cualquier problema que impida que este “vivo” el mensaje se perderá y nunca llegará a su destino. Por otra parte, los Mensajes Asincrónicos tienen la gracia de que si es que el cliente/servidor envía un mensaje este se encolara en una cola y esperará que en algún momento la contra parte agarre el mensaje y lo procese, por lo tanto, esta puede estar apagada o fuera de servicio, pero una vez que vuelva podrá leer el mensaje sin problemas. Aunque existen ciertas desventajas a las colas, como su nombre lo dice, entonces tienen un límite, y eso va a depender de el espacio en disco que tenga la maquina en que se este operando, así limitando la capacidad de las colas, además que si las colas están muy llenas aun así puede existir perdida de mensajes, pero esto es con muy bajo chance y además pueden producir una entrega más tardía en los mensajes por lo llenas que están.

Finalmente, un tema importante en cuanto a sistema distribuidos es la escalabilidad de estas dos implementaciones, ambos sistemas técnicamente son escalables, pero los mensajes asincrónicos son potencialmente mucho mas y esto es debido a la durabilidad de los mensajes, como se menciono anteriormente, a pesar de que una de las maquinas se encuentre apagada el mensaje llegará eventualmente. Además, con el sistema Publisher/suscriptor, es extremadamente fácil poder establecer la conexión entre las dos partes, con tal de tener un nombre con el cual identificarlo se puede lograr una comunicación y se pueden mandar muchos mensajes sin perderlos. Además, permite que varias máquinas con tal de tener acceso a las colas y a los nombres puedan encargarse de diferentes tareas.

Por otra parte, el rpc es más limitado, es más difícil generar mensajes asincrónicos, prácticamente es una llamada a funciones, entonces si se reciben

muchas solicitudes, claramente tendera a colapsar ya que tiene que responder a cada una de las solicitudes pudiendo producir respuestas lentas.

En conclusión, ¿Que técnica se recomendaría para este problema?, la respuesta a esto es “depende”, si se tienen los recursos y se quiere hacer algo que pueda escalar y perdure en el tiempo con mayor facilidad y quiera poder programar a gusto (teniendo en cuenta las desventajas de aun así poder perder mensajes, uso de disco, implementación, etc.), los “Mensajes Asincrónicos” serian la mejor implementación. Por otra parte, si se quiere hacer algo mas acotado, que no requiera mensajes altamente perdurables, pero que sea confiable, y sepa que el mensaje le llegara acertadamente a la persona que se solicita, “RPC” es una gran solución.

Cabe destacar que como se dijo ambas soluciones son bastante buenas dependiendo de que es lo que se tiene y busca, si fuese el caso en que se quisiera implementar RPC con mensajes asincrónicos, esto se puede hacer a través de colas, incluso en la misma documentación de rabbitmq (lo que se uso para esta tarea) explica como implementar rpc con colas.