Accenture Academies Online

Big Data y su uso en Cloud Computing 21th april

CM = Carmen Marín

SG = Salvador García

BD = Big Data

CM: Pertenezco al equipo de Talent Acquisiton y mi rol dentro de Accenture es la gestión de actividades de captación para atraer talento junior en las universidades del país.

La sesión tendrá una duración aproximada de 90 minutos. Una primera parte, que os impartirá Salvador y al finalizar la sesión volveré con vosotros para cerrarlas, así que nada, os dejo en muy buenas manos, con Salvador, gracias.

SG: Vamos a dar una pequeña charlilla sobre lo que es Big Data, cómo se trabaja y el objetivo es que al final de la charla tengáis claro en qué se basa esta tecnología, como ha surgido y qué es. Y cómo se trabaja con ella. Que sepáis que no es un servicio, que no es una herramienta… Es una charla muy rápida, no podemos profundizar en grandes temas, pero quiero intentar al menos, que las cosas que se digan quedes claras y tengáis una idea preliminar. Sobre todo, lo que es y qué tendríais que mirar. Por dónde empezar en el mundillo.

Lo primero que me gustaría comentar es que no es una tecnología nueva. Big Data existe desde los principios de la informática, lo que antes se llamaba el procesamiento distribuido. Pero unos factores en los últimos años han conseguido que explosione, que se utilice. ¿Qué factores han conseguido esta popularización de la BD? Los datos, en los que en un primer momento llegan asiduamente en nuestra época. Cuando empezamos a trabajar en el mundo de la informática, todos los datos se solían introducir en BBDD relacionales, porque los datos tenían una estructura manejable, fija y eran finitos. Por muchos datos que tenían, en una máquina Oracle, no estaba preparado para ello.

Total, que empezamos a necesitar algo para manejar esos datos, porque datos que antes no se consideraban útiles o al menos no se obtenían beneficios de ellos, ahora nos damos cuenta de que, de trabajar con datos personales, puede generar beneficios, ya sean económicos, sociales o beneficios éticos. Pero esa popularización de los datos hace que necesitemos una nueva tecnología, o una tecnología usable. Para que os deis cuenta, aquí tenéis el número de peticiones, de Teras que se trabaja por segundo, por minuto en un día. Son unos datos totalmente bárbaros.

¿Para qué sirve la información? ¿Cuándo un dato pasa a ser información?

Vosotros más que nadie sabéis que la información sirve para sacar beneficios, dónde navegáis, dónde vais… Para daros publicidad personalizada, hacer modelos predictivos, qué comprarás, dónde viajaréis.

En una aerolínea de España hacían modelos predictivos de ver dónde iba a viajar alguien para enviarle una serie de comunicaciones. Ese tipo de información genera beneficios siempre y cuando se sepa tratar la información.

Analizar todos esos datos no es trivial. Esos datos crecen exponencialmente, no tienen una estructura fija. En un like de FB hay megas de información que se envían que hay que enviar a quién le disteis el like, quienes son vuestros amigos, las interacciones… Y hacer cálculos con todo ese tipo de datos es bastante complejo. Lógicamente, además de la capacidad computacional para trabajar con los datos, se necesita una capacidad de almacenamiento infinita y, desde luego, hacerlo con una respuesta, con una latencia, con un tiempo, óptimo.

Por eso muchas veces, yo en proyecto o en entrevistas que suelo hacer, pregunto muchas veces qué es BD. Y no hay una definición o yo no conozco una definición canon. BD es un cúmulo de cosas. He seleccionado una definición con la que me siento confortable. BD es poder tratar grandes cantidades de datos que no se pueden utilizar unas tecnologías tradicionales. Yo diría más bien que trabajar con esos datos en un tiempo medianamente óptimo. Desde un día, una hora, segundo…

Lo que sí sabemos es que hace 15 años para manejar informes grandes eran procesos de semanas. Y ese mismo procesamiento ahora se puede hacer en minutos u horas. Vamos a ver por qué.

No me voy a meter mucho en la historia de la informática, pero vosotros sabéis que los ordenadores ocupaban una habitación. Y cuando necesitas trabajar con muchos datos, lo necesitabas cada vez más grande. Tirabas tu ordenador y ponías uno más grande o lo ampliabas verticalmente. Así se procesaban los datos. Un único ordenador, cuanto más potente mejor. Ahí tenéis las fotos de una científica militar increíble. Os recomiendo leer alguna biografía de ella porque era totalmente increíble. Es una de las pioneras de la informática y de los sistemas distribuidos. Ella lo explicó muy rápidamente, ella decía “si tú tienes que cargar un tronco, puedes utilizar un buey, pero si tienes que cargar un tronco más grande, puedes comprar un buey más grande y más fuerte, pero la mejor opción es poner dos bueyes. Y si tienes que acarrear un tronco mayor, puedes poner 5-6 bueyes.” Esto es uno de los pilares de la computación distribuida, que es poner más ordenadores en paralelo.

Big Data es procesamiento paralelo. Cuantos más ordenadores tengas, más rápido te va a ir. Este sistema de distribuido, hace muchísimos años que existe. No es algo nuevo, no es una novedad, existe. Los sistemas distribuidos funcionaban bien, pero tenían ciertas complicaciones, eran muy caros. Cada ordenador que ponías… Pocas empresas se lo podían permitir. Segundo problema: existía un cuello de botella. Tú trabajabas en paralelo, pero solía haber una CPU que era la que iba distribuyendo os trabajos y recopilando los datos. También era complicado trabajar con ellos, codificar. Si alguno ha trabajado con JAVA y se ha metido en los silos, sabrá de lo que estoy hablando, es complicado. No era una cosa común.

Llega un momento en el 2003, que nace algo que se llama Hadoop, y esto es lo que lo cambia todo, lo que hace explosionar BD. Si queréis empezar en el mundo de BD, intentad estudiar un poco o leer algún artículo de Hadoop. Yo os voy a dar unas pinceladas. Si queréis profundizar en él, podéis.

Línea del tiempo: cómo almacena de manera distribuida todos los archivos. El Google File System. En 2004 saca también un web papper que dice cómo procesa todo lo guardado en su

sistema de archivos, el map reviews, un paradigma de programación. Guardo los archivos y los proceso generando un paradigma, map reviews.

Alguien en 2005 se estudia los web pappers, y empieza a trabajar en una herramienta open source y crea Hadoop y sale a producción, en el 2008. A partir de ese año y el nacimiento de Hadoop, la gente se empieza a volver loca con el tema de BD, porque poner un Hadoop es sencillo y barato, entre comillas. Infinitamente más barato de lo que había, no pide un gran hardware para funcionar.

Sale a producción en 2008 y lo cambia todo. Un montón de grandes empresas empiezan a trabajar con Hadoop, y además a lo largo de los años, empiezan a salir otra serie de tecnologías que complementan a Hadoop, el ecosistema Hadoop.

Cosas importantes: Hive, creada por FB, flume, Spark… Seguramente ya habéis oído hablar de Spark, pero Spark es una cosa muy importante a día de hoy. 2011, 2012… Otro de los pilares fundamentales actuales, Kafka. Para que os hagáis una idea, Spark suele ser para trabajar con procesamiento batch y temas realtime, Kafka. Luego ya se empiezan a… con todo este ecosistema. Y empieza a surgir arquitecturas, lambda, kappa… Pero ya os digo, empiezan a salir un montón de tecnologías y frameworks que complementan JAVA.

Total, que vamos a quedarnos un poco con lo que es Hadoop. A día de hoy se sigue trabajando mucho con Hadoop, y, sobre todo, aunque no se trabaje con Hadoop, como spark, pero los pilares fundamentales o el core, funciona igual que como funciona Hadoop. Si entendéis Hadoop, vais a entender BD.

Hadoop es un framework, es un proyecto open source, vosotros lo podéis descargar gratis, instalarlo en todos los ordenadores que queráis. Esto funciona de manera distribuida, ya sea en vuestro portátil, simulando que estéis en distribuido, o en dos o tres portátiles. Entonces es un framework, open source y lo que os comentaba, tiene ciertas ventajas, pero sobre todo que es barato, sencillo programar en él, fiable antes fallo. Si lo tenéis en 15 ordenadores y 4 se estropean, sigue funcionando…etc. Tenéis ciertas ventajas, corre sobre cualquier SO, está escrito en JAVA, es multiplataforma…etc.

Pero ¿qué es Hadoop? Sencillo, son 3 patas: HDFS. Que es el sistema de almacenamiento, o el file System de almacenamiento. Aquí tenéis una gráfica, que lo vais a entender muy fácil. Imaginaos que vosotros queréis subir un archivo a un clúster de Hadoop. Simplemente, ese archivo, lo va a coger una de ellas, el name node, el maestro. Pero es importante una cosa, el maestro no trabaja, no va a generar un cuello de botella. Este maestro manda al nodo que sabe que tiene menos trabajo, le envío el job. Vosotros subís un archivo a un clúster de Hadoop. El name node no lo coge y dice: vale, lo parto en trozos. Si subís un Giga, lo parte en trozos de 300Mb. Y deja un trocito en un nudo, otro en otro y otro en otro. Si juntas estos archivos, tendrás el Giga. Y así tiene los archivos partidos y distribuidos. Además, va a hacer una copia de seguridad en nodos diferentes. Simplemente por si una de estas máquinas se cae, otra máquina tendrá la información. ¿Cuántas copias o nodos de seguridad hay? Parametrizable, de 2 a 20. A más máquinas, más posibilidades hay de que algo falle. A más máquinas, más copias de seguridad es conveniente hacer. HDFS para almacenar archivos, de manera distribuida, partidos y con back up.

YARN, es un gestor de recursos, es lo que le va a indicar que nodo tiene menos carga al name node. Con esto no vais a trabajar, porque ya lo tiene en sí Hadoop y es lo que utiliza.

Y el procesamiento, que aquí está la chicha. Map reviews, os comentaba que Google, lanzó un web papper, Hadoop lo hizo suyo e hizo un map review propio, un sistema de programación para procesar esos datos. Es una cosa muy simple, cuesta un poco de entender al principio, pero es muy simple. Un Word count. Habréis hecho mil millones de veces un lola mundo. En BD no existe, existe un Word count, es un archivo que tiene palabras, ese archivo se separa en líneas, y esas palabras van a nodos diferentes.

Siguiente fase, el reduce. Las palabras se ordenan por clústeres y esas claves llevan a un procesamiento reduce, que puede ser que cada uno de esos procesamientos, estén en una máquina diferente, y cuenta esas palabras. Esta cosa tan sencilla, al final da un resultado donde pone: driver 2, driver 3… esto es un map reviews, la base de toda programación distribuida o toda programación en BD. Y como os comento, se codifica en JAAVA, para que veáis esta tontería de contar palabras, codificar esto en JAVA, es esto y esto. No te vas a morir ni te vas a hacer viejo con esto, pero si para contar palabras tengo que hacer este programa de tantas líneas, para hacer algo más complejo, hay que saber un poco. Siempre se trabajó con map reviews, pero hubo un momento en el que salieron más cosas que complementaban a Hadoop. A día de hoy nadie trabaja con map reviews. Con lo que se trabaja, funciona igual que un map reviews, por eso es importante que lo entendáis. Este programa hecho en JAVA de map reviews, con Spark, se hace en estas 3 líneas. Es lo mismo que el map reviews en JAVA, esto es Spark.

Estas tecnologías utilizan el map reviews, aunque tú no lo veas, como el Hive. Hacer una querys, lanzarla y te genera por debajo un map reviews en JAVA. A día de hoy, Hive y Spark se utilizan mucho.

Simplemente, quedaros con eso, map reviews, HDFS, el file system, Yarn y map reviews.

Salen un montón de programas, aquí hay una pequeña parte, pero en el ecosistema BD hay un montón de programas. Spark con sus galerías de machine learning, en realtime streaming, kafka…. Y BBDD NoSQL. Es Hadoop lo que cambia el paradigma y lo que a día de hoy sigue en boga.

Muchas veces os encontraréis en conferencias que os dicen que BD son las tres Vs. Volumen, velocidad y variedad. Otros dicen que son 5, con velocidad y valor. Dentro de un año serán 27 Vs. A mí me salen sarpullidos, pero sí que es cierto que BD dicen 3 Vs porque volumen, tiene un montón de datos, velocidad porque hay que tratarlos rápidamente y variedad porque no tienen estructura fija.

Vamos a abstraernos un poco de Hadoop y vamos a ver cómo se procesan los datos.

3 problemas: variedad, velocidad y volumen.

¿Qué se está haciendo a día de hoy? Hay 2 cosas, procesamiento batch, que se ejecuta cada cierto tiempo, cada hora, cada 10 minutos, cada día… Grandes volúmenes de datos, que tardan mucho en procesarse, y para algo que necesito muy rápido, streaming o procesamiento realtime. Cuidado con realtime, porque en realidad no existe, porque nada tiene la misma velocidad que la realidad. Yo ponía el ejemplo de FB o TW. Tw se considera realtime, pero tú cuando das un RT o das un me gusta a un tweet, todo el mundo no lo ve a la vez, ni mucho menos. Yo puedo verlo al segundo y Juan el de Málaga lo puede ver a los 3 minutos. Se considera realtime, sí, pero llamémoslo realtime. Un coche autónomo no va a girar a la derecha 3 minutos después, so es realtime o near realtime, tampoco gira al instante. Pero si se

necesita una velocidad mucho más precisa. Ambos son realtime, pero veis la diferencia, no tiene y no puede ser inmediato.

Y para la variedad, que os comentaba antes que los datos no tienen base fija, para eso nace la BBDD NoSQL. Un procesamiento batch o cualquier tipo de procesamiento tiene que ser escalable. Esto lo permite Hadoop y cualquier sistema BD. Cuantas máquinas metas, que sigan funcionando. Esto es todo escalable, distribuido y todo ejecutándose a la vez, en paralelo. Que sea tolerante a fallos es imprescindible. Que se caiga una máquina y siga funcionando, lo necesito.

Eso es un procesamiento batch en streaming, pero la velocidad la necesito mucho más rápida. Lo que se está haciendo ahora es unir estos 2 mundos y generar algo híbrido, que mi arquitectura tenga batch y streaming y ambos se ayuden. Tengo una arquitectura híbrida. Me da igual que sea batch o streaming, las controlo a las dos y las mergeo a las 2. Esto en el mundo actual es difícil ver cualquier tipo de proyecto, cualquier tipo de arquitectura que no sea híbrida. Yo solo trabajo con batch, raro. Yo solo trabajo con realtime, más raro todavía.

No me voy a meter aquí, pero que sepáis que nacen las arquitecturas lambda y kappa, que son las arquitecturas híbridas. Hay una capa batch y una arquitectura realtime. La lambda es la que primero nace, y la Kappa es muy similar, pero dicta que no necesita la capa batch, porque lo puedes hacer también con la arquitectura realtime. También se utiliza mucho la kappa.

Los datos los puedo guardar en el HDFS de Hadoop o en una BBDD, pero las BBDD tradicionales ya no sirven, por la capacidad de almacenamiento, no trabajan en paralelo y porque los datos no tienen una estructura fija y por eso nacen las BBDD NoSQL.

La diferencia más fundamental entre las BBDD NoSQL y las relacionales como Oracle, lo más singular de diferencia entre estas BBDD es el tema de transaccionalidad. Cuando trabajáis con una BBDD Oracle, podéis hacer un update masivo, o todos subdatean o no subdatea ninguno, existe una transaccionalidad. Si vosotros hacéis una transferencia de 50€, hay un primer paso que es que te los quitan de la cuenta y un segundo que es que se le suman a otro. Esa transferencia es transaccional. Si uno de los dos pasos falla, todo se vuelve atrás. En un sistema de BBDD NoSQL, no existe esa transaccionalidad. A día de hoy al menos, y veréis HIVE3 y veréis algunas BBDD que son BD y son transaccional, y no es así, tienen ciertos trucos, pero al no ser transaccionales, significa que voy a dar 50€ a alguien, primero me los quitan, segundo se los suman a la cuenta de la otra persona… Esa segunda operación falla y esa acción primera no va para otras. Las BBDD NoSQL, no son transaccionales.

BBDD NoSQL hay muchos tipos, dependiendo un montón del caso de uso. Hay sobre todo 4: clave valor, documentales, columnares y de grafos. Y os voy a comentar un poco las que más se usan o las que yo veo más útiles, empezar a mirarlas o las que más éxito tienen. MongoDB, Hbase… y de grafos no controlo muy bien, Neo4j se usaba mucho hace años, pero si queréis echar un ojo, recomiendo MongoDB para empezar.

¿Qué se hace en BD? Tenemos que Hadoop ha implosionado en el mundo BD, tenemos dos tipos de procesamiento, batch y realtime. Ingerimos muchos datos y queremos hacer algo con esos datos.

Un flujo muy habitual, son las ETLs: Extracción, Transformación y Carga. Que se basan en eso mismo. Adquisición de datos, almacenamiento y procesamiento de datos, y carga en algún

sitio. Llevo los datos a una BBDD o a un file System… Esto es el abc del procesamiento BD, pero no es lo único que se hace en BD, pero sí es uno de los pilares fundamentales.

Tecnologías. Quiero que os quedéis con alguna tecnología de cómo se hace esta TEL. Adquisición de datos en batch. Para mí, importantes… Nada de lo que estoy poniendo aquí es complicado, ya sea Spark, mongo… cada uno su mundo. Pero adquisición de datos, podéis echarle un ojo a Sqoop. Almacenamiento de datos, una BBDD NoSQL, HFS. Análisis de datos, Hive, una cosa muy sencilla para empezar que se usa mucho desde hace años y se siguen usando y si sabéis lanzar querys y SQL, lo vais a saber usar. Y análisis de datos, Spark o SparkSQL. Al final os diré lo que se pide en el día de hoy y lo que es lo importante. Si te interesa esto, que sepas que esta tecnología es la que os van a pedir. Spark para mi es una de las principales, que es hacer un map reviews con un framework que se llama Spark y donde procesa todo memoria, por lo tanto, es muchísimo más rápido. Y luego los resultados los puedo dejar en una BBDD o en una herramienta de Business Intelligence como Tableau, o QuickSight o crear gráficas o lo que yo quiera. Pero adquisición, almacenamiento y procesamiento de datos, para realtime es lo mismo, con otras tecnologías en algunos casos.

Como os digo, en realtime Kafka es fundamental. En el mundo de Amazon, Kinesis que sustituye a Kafka. Ambas son un poco lo que se está utilizando a día de hoy. Y para análisis, existe el Spark Streaming, pero creo que Spark Streaming, igual que os digo que para batch es el número 1 sin duda, para realtime no empezó con buen pie. Sacó una nueva versión donde cambiaba todo, pero yo si tuviera que recomendar algo para realtime, para mi el número 1 es Flink. Y en resultados pues lo mismo.

¿Cómo trabajamos nosotros? Nosotros siempre empezamos a trabajar con sistemas on premise. Pero desde hace ya unos años, este trabajo on premise, ha quedado un poco relegado porque ha habido un montón de servicios en la nube que ha florecido y se han vuelto muy populares. Yo, personalmente, llevo unos años trabajando con Amazon. Sé alguna cosa de Microsoft Azure o Google Cloud, pero yo soy especialista en Amazon. Lo que os voy a contar aquí está muy enfocado en Amazon. No quiere decir que sea ni mejor ni peor. Veremos una slide luego de cuáles son las más utilizadas y veréis que Amazon es la más utilizada. Pero no hay ni mejor ni peor. Unas tienen unas fortalezas, otras en otras… Y es difícil que no puedas hacer una cosa con Microsoft y sí con Amazon.

¿Qué es al final trabajar en la nube? Ya lo conocéis de sobra. Hay mil millones de definiciones de lo que es la nube. A mí la que más me gusta es: es el ordenador de otro. Trabajar en la nube es trabajar en otro sitio que no es tu ordenador. Entonces, la migración a la nube de sistemas on premise tradicionales, donde tenías un centro de datos y tus ordenadores y un montón de cosas, ya pasa. Cada vez se trabaja más en la nube, cada vez se está migrando más en la nube y ya es el futuro. Es presente y futuro. No todo estará en la nube, habrá todavía sistemas locales, pero la inmensa mayoría se va a trabajar en la nube y se va a trabajar con Amazon, con Azure o con Google Cloud. ¿Por qué? Porque como todo, tiene sus ventajas e inconvenientes. Yo trabajo en OpenBank, que es un banco, y el 100% de todo lo que tiene OpenBank, lo tiene en Amazon, no tiene ni un ordenador propio ni nada, solo trabaja con servicios de Amazon, procesamiento Amazon y clústeres, nodos y todo de Amazon. Esto que lo haga un banco es muy definitorio, no es un videoclub. Que todos los datos y procesamientos estén en la nube es mortal. Las ventajas que tiene son ahorro y cuidado. Como todo, es depende. En la nube, pagas por lo que usar. Si yo estoy procesando algo y necesito 5 máquinas durante 5 minutos, pago 5 minutos. Eso es ahorro.

No siempre voy a ahorrar. Si yo voy a trabajar con 5 máquinas 24/7 es más probable que me salga más barato comprarlas. Pero, como pagas por lo que usas, todos los procesos que se trabajan, es que no te coges una máquina para ti, sino que cuando necesitas una, la pides, levantas, procesas y la tiras. Y ahí puedes decir que vas a pagar por 5 minutos o 20. Si dejo la máquina levantada, voy a tener una respuesta más rápida, y en algunos casos la voy a necesitar, como en realtime. Ahí voy a estar pagando porque siempre está levantada, y ahí voy a pagar más. Es difícil no ahorrar, pero es posible que puedas pagar más en la nube que fuera de la nube. Depende de tu uso.

Yo me despreocupo de comprar más discos duros, máquinas… Si quiero más la pido o escalo.

En hardware lo mismo. Antiguamente si querías 2-3 máquinas nuevas para poner a tu clúster, menos de un mes para tenerlas en producción, era difícil. Tenías que comprarlas, configurarlas, probarlas… Menos de un mes difícil. A día de hoy es inmediato, haces un clic y la tienes.

Actualizaciones automáticas, yo me despreocupo un poco en este tipo de servicios, en el mantenimiento. Si hay una nueva versión de la BBDD, Kinesis… automáticamente me lo ponen y me aseguran que sigue funcionando. También quería comentar la tecnología serverless. Lo tienen todos, Microsoft, Google y Amazon. La tecnología serverless, subjetivamente, es a donde se dirige este mundo. Si quiero una BBDD MongoDB, desde la nube hago un clic. Quiero una BBDD, que tenga 2 nodos maestros, 4 esclavos, esta CPU, esta memoria, almacenamiento… Yo voy seleccionando mi clúster de Hspace por ejemplo. Yo lo tengo ahí y eso es lo normal. Tecnología serverless, significa que te olvides de eso. Que quieres trabajar con una BBDD, con un servicio de realtime… selecciónalo y olvídate de tener que elegir la máquina, la memoria, los nodos… Me va a dar automáticamente lo conveniente, y si necesito más, me va a poner más, pero yo me abstraigo de eso. Eso es la tecnología serverless.

CM: hola Salva, tengo una pregunta de Iván. ¿No hay empresas reacias a subir los datos en la nube? ¿Cómo de seguros son esos datos en la nube para empresas? ¿O se evitan subir los datos más confidenciales?

SG: Es una buena pregunta. Depende de la empresa y de la ley de protección de datos. ¿Cómo de seguros son los datos en la nube? Segurísimos. Digo que lo son porque desde el punto de vista de Amazon, todo el mundo tendrá lo mismo, y los datos en la nube están en habitaciones físicas, en localizaciones físicas, donde prácticamente nadie puede entrar a ellas. El dueño de Amazon no puede entrar ahí. Llevan un control de seguridad que es flipante. Pueden entrar algunos, monitorizados al 100%. Más de una persona siempre para ver qué hacen los demás. Es complicado entrar ahí. Que te lo hackeen… Amazon es muy complicado, pero también te pueden hackear tu sistema on premise en Internet. Y, sobre todo, tienen un sistema de encriptación. En el caso de Amazon, usan las KMS propias. Y tú puedes crear tus propias claves de encriptación. Nadie que no tenga esas claves de encriptación, aunque entre en el centro, robe los ordenadores e intente sacar los archivos, si no tiene las claves no va a poder entrar. Por eso te digo que es muy seguro. Es bastante más seguro que la seguridad que tú consigas poner en tu centro de posicionamiento.

Más que por ese tema de seguridad, los problemas pueden ser por la LPD, porque esos datos sensibles no pueden estar en un centro de China, tienen que estar en Europa o España. Ahí puede haber problemas. Aunque puedas seleccionar en las máquinas que esté, por ahí puede venir algún problema, por la LPD. Puede ser que no puedas hacerlo, porque por política no puedas, pero quería poneros un poco el ejemplo de OpenBank, para que veáis que un banco que tiene datos muy sensibles, lo tiene en la nube, entonces no es un obstáculo insalvable en la mayoría de los casos.

Continuamos si no hay más preguntas por ahora.

Estamos un poco acabando. Voy a hacer una recopilación al final para intentar que nos quede claro.

Os comentaba antes, empresas que trabajan en la nube o que dan sus servicios en la nube. Aquí tenéis el porcentaje de usuarios, de los usuarios que trabajan en la nube, un 31% trabaja con Amazon, un 16 con Azure y un 8 con Google Cloud.

CM: Salva, disculpa de nuevo. Daniel comenta: por no hablar del contra de tener toda tu información concentrada en un único lugar físico. Si esa información se pierde y no la tienes en la nube, se pierde para siempre. Imagínate que un banco pierde toda la información de deuda de todos sus clientes.

SG: Bueno, mejor que cierre el banco si pasa eso. Entiendo la postura, pero no es del todo cierto, porque si tienes algo distribuido cuando tú pones en la nube y puedes elegir un país, puedes elegir Francia, todas las máquinas no están en una misma localización, están en varios centros a lo largo de esa localización, donde tienen réplicas. Y siempre hay réplicas de lo tuyo. Además de eso, si cae una bomba atómica sobre Francia, y todos los ordenadores o centros de datos de Amazon desaparecen, esos también están en otras localizaciones réplicas. Tienen un control sobre desastres naturales como pueden ser terremotos o sunamis, que desaparezcan localizaciones físicas… está controlado para que a lo largo del mundo esté replicado. Sí que es verdad que puedo tenerlo en China o no, pero no están en un mismo lugar físico, están en muchos. Además, tú los puedes seleccionar. Dónde están tus datos, tus replicas. Aparte de esto, hay muchos servicios de backup de esos datos para traértelos si quieres. Hay una cosa que se llama Snowball, donde te viene un camión de Amazon y te da los datos o te los recoge. O maletines de Amazon. Siempre tienes los datos disponibles. Si eres capaz, también tú puedes hacer tus propios back ups.

No quiero pecar de ingenuo, pero creo que no es necesario, porque están distribuidos en varias localizaciones y países.

Simplemente que veáis un poco la cuota de mercado. Alibaba salió hace poco y está pegando bastante. Como veis tiene un 4%. Pero si tuviera que meter 1€ en acciones, lo metería en Alibaba quizás.

Simplemente como os digo, desde el punto de vista de Amazon, vais a trabajar con ello, con Amazon, Microsoft…. En el mundo laboral hay un 80% de posibilidades de lo que lo hagas. Es sencillo y alguno lo habréis hecho ya. Trabajar en la nube, a mí me parece una maravilla, porque no tengo que instalarme las cosas, en mi ordenador no va… Y pagáis céntimos. Para

trabajos de carrera o que no sean industriales, se pagan céntimos. Por levantar una máquina, hacer una querys, procesar… no sabéis el ahorro de quebraderos de cabeza que os va a dar.

¿Por dónde empezar? Mis consejos son de Amazon. No sé cuántos servicios hay, más de mil. Si tuviera que mirar algunos: S3, habéis visto que en Hadoop existía el HDFS. El almacenamiento de Amazon se basa en S3, donde los datos están de manera distribuida. Subo los archivos, bajo los archivos, miro los… etc. IAM, temas de seguridad, donde vais a poder (si os creáis una cuenta en Amazon gratuita por ser estudiante) hacer usuarios, grupos de usuarios, "a este usuario le permito ver esto en S3, a este no…" todo eso bajo IAM. EC2, simplemente para levantar una máquina, "quiero una máquina de 16gb con Linux instalado, Hive y Spark." Pues trabajas con ella.

Y como os comentaba antes, veo súper útil, de presente y futuro, la tecnología serverless. Bajo mi punto de vista los que más se están utilizando son las lambdas, kinesis, DynamoDB y Athena. Esto es una BBDD NoSQL. La dejaré para el final por si no tengo tiempo. Pero con lambda para lanzar cualquier proceso en serverless, lanzo el proceso y no tengo que lanzar con Spark las máquinas. Lo pongo en 4 líneas y me despreocupo.

Kinesis, para temas realtime, y Athena para hacer consultas SQL.

CM: Salva, nueva pregunta de David. ¿Y a la hora de que una empresa tenga todo en la nube, qué opinas sobre los ataques que puedan lanzar y que llegue la información comercial de la empresa? ¿O tenerlo en la nube tiene más protección?

SG: es normal la pregunta y lógica. Si lo tienes en la nube, sí, te pueden hackear. Si no lo tienes en la nube es posible que también si sales. Alguien se puede conectar a tu clúster, a no ser que no tengas conexión a Internet. Es habitual que alguien se conecte. Los dos son susceptibles de ese ataque. ¿Qué va a tener más seguridad, Amazon, Azure, Google, que ha contratado a los más expertos del mundo n seguridad o al menos lo intenta, y ha contratado a 500.000 expertos, o una empresa en la que trabajes tú? Que, seguro que son magníficos, pero por regla general, por porcentaje, es bastante más posible que tenga mejor seguridad Amazon. Si a Amazon lo hackean, desaparecerá, porque nadie querrá tener los datos en Amazon. La inversión en seguridad que tiene que tener ahí. Yo diría que es más seguro Amazon. Lo normal es que todos tus datos, y si no lo haces lo está haciendo mal, que estén en la nube y on premise, estén encriptados, donde hay una clave de encriptación que solo conozca yo y solo utilice yo. Me aseguro de que, si lo roban, tengo los datos encriptados.

Todo es susceptible de hackeo. Pero si tuviera que apostar apostaría por la nube, por la inversión brutal de seguridad que hace esta gente. Es su vida, si lo hackean desaparece, mueren.

Espero que haya quedado mi postura subjetiva.

Jose Luis: quería hacer una pregunta. Amazon y Google hacen unas inversiones tremendas en seguridad, pero en el caso de una PyME que no puede permitirse esa inversión, ¿usted recomendaría tener esos datos en la nube?

SG: sí recomendaría mantener los datos en la nube, porque esas grandes inversiones, repercute en la seguridad de mis datos, mis datos van a estar tan seguros como cualquiera que los tenga en la nube. Y los datos en la nube están seguros de por sí. Me reitero, encriptar mis datos son dos clics. Hacer una clave de encriptación son dos clics, no tengo que invertir en un ingeniero que me encripte los datos ni nada de eso. Están más seguros en la nube.

Jose Luis: Gracias.

SG: Nada. Os comentaba un poco las tecnologías o los servicios con que empezar en AWS. Tenéis un montón de tutoriales en la documentación de AWS o Azure y podéis hacer… Es un ejercicio donde vais a aprender mucho. En mi proyecto muchas veces lo pido, y sé que la gente aprende un montón. Es un ejercicio de ingesta de datos, un stream de datos, que cualquier página da datos de clics de lo que sea, llevarlo a kinesis, el procesamiento de realtime de AWS, de kinesis llevarlos a una lambda y luego dejarlos en algún lado, en S3 o en una BBDD Dynamo.

Este ejemplillo, puedes encontrar un montón de tutoriales y documentación para hacerlo. Os va a venir super bien.

Servicios interesantes de Amazon, los que os dije antes. Y remarcable es MR, que es Hadoop, al final es un clúster de máquinas con Hadoop instalado y ahí se procesa todo. Servicios MR, servicios redsys y blue, que es para hacer procesamiento Spark sin servidor… Hay un montón de servicios guapos. Luego os pasaré las slides. Aquí tenéis un poco las principales. Yo miraría estas primero porque os van a dar una visión más global.

Conclusiones. Procesamiento realtime o near realtime. El procesamiento batch parece que no va a haber novedades. Spark funciona muy bien. Habrá novedades en realtime. Y poco a poco los procesamientos batch, irán migrando a sistemas realtime. Arquitectura híbrida, hay que conocerlo, no hay batch o realtime. En la nube todos nos estamos moviendo. Lo que empieza a día de hoy, se plantea directamente en la nube, y lo que no está en la nube, se empieza a plantear migrar a la nube, se están haciendo esas migraciones. La migración ya está funcionando a día de hoy. En Accenture, internamente, todo lo teníamos en centros locales para datos internos, y se ha hecho un estudio de a qué tipo de nube migrar, ha dado Google Cloud, y todo lo vamos a migrar nosotros mismos a la nube.

Spark, fundamental para machine learning, deep learning… Si tenéis que estudiar algo en esta vida, empezad por Spark. Spark es un framework, tiene un lenguaje de programación.

Realtime habrá novedades próximamente sin duda.

Recomendaciones, lenguaje de programación…

CM: Salva, nos pregunta Pablo ¿has estado hablando mucho de Hadoop, Spark… y qué pasaría con Python?

SG: Pues justamente venía ahora. Date cuenta de una cosa, hablo de Hadoop porque es un framework, hablo de Spark porque es un framework. Y no he hablado de ningún lenguaje de programación, porque Spark, como framework, admite que tú lo programes en lenguaje de programación que tú quiera. Mentira. Te deja JAVA, Scala y Python. Y justamente decía que el lenguaje de programación que recomiendo, pongo Scala. Mirando los gráficos de los lenguajes de programación mejor pagados, Scala estaba de las primeras posiciones. No pongo Scala por eso, pero es importante. Lo pongo porque para trabajar con Spark, creo que se trabaja mejor con Scala. Spark está programado en Scala. Novedades que salgan en Spark, salen primero en Scala, y es el lenguaje nativo de Spark. Por eso he metido Scala. Se puede programar en Python y se programa muchísimo. ¿Y fuera del mundo Spark se usa Python? Muchísimo. ¿Y Scala? No.

Yo soy de Scala, pero si tuviera que aprender un lenguaje de programación, aprendería Python, porque se usa más. En machine learning, Deep learning… y casi cualquier ámbito, se usa Python. Pero Scala a mí me gusta mucho.

Con eso más o menos… Python lo recomiendo. No es nada que afecte a BD, es un lenguaje de programación que se utiliza.

Framework, Spark. Realtime, y esto es super bonito e interesante trabajar en realtime, vais a flipar, y recomiendo Kafka, que es una obra maestra de la ingeniería, sin ningún tipo de dudas. Y como procesamiento… Si quiero meterle algo más pesado, recomiendo Flink porque se utiliza a día de hoy y se va a utilizar mucho mañana.

Últimas recomendaciones. Una certificación de Amazon, Azure… es bastante asequible económicamente, entre 250-300€, lo que es el examen, y hay dos oportunidades. Si suspendéis la primera tenéis una segunda. Y vais a aprender y se valora mucho esa certificación. Es asumible y os va a servir, tanto a nivel personal de aprender como profesional a la hora de seleccionar dónde queréis trabajar.

Y una cosa que se suele olvidar, yo noto en falta cuando viene nueva gente a la empresa o los proyectos, es haber tenido más callo en integración continua. En cualquier proyecto lo vais a utilizar.

Puse Bamboo con una n y está mal. Jenkins y Bamboo en herramientas de integración continua. Para que haga un push, se me genera un hard, suba a preproducción…etc. Toda esa integración continua que no requiera de trabajo manual.

CM: Salva, Celia nos pregunta que si para estudiar para la certificación… ¿Nos recomienda algún curso?

SG: sí, es como todo. Puedes apuntarte a cursos presenciales o demás. Por regla general es autodidacta. Yo viendo a la gente que se ha sacado la certificación o que la ha preparado, en mi proyecto, en mi empresa. Podéis meteros en Coursera, por ejemplo, o Cloud Gurú, que son cursos online y ahí hay muchos cursos de preparación y en mi proyecto han seguido esos cursos. Te preparan, te enseñan la arquitectura. Te preparan para el examen, y a lo mejor te cuesta 15-20€. Pero no deja de ser autodidacta. Recomendaría ese curso y hacer mis pruebas. No es tan complicado. No es tan difícil. Requiere preparación, pero no es imposible.

CM: Jesús nos comenta que Amazon tiene sus cursos previos a la certificación.

SG: Sí, es que vais a ver un montón de información, tutoriales y cursos. Yo recomiendo simplemente por ver el resultado de Cloud Gurú o Coursera. No me he metido en los de Amazon. ¿Alguna cosilla más?

CM: hay una pregunta más. Jesús dice: AWS Academy Cloud Foundation.

Pues gracias Salva, muchas gracias por compartir tu conocimiento. Estamos llegando al final de la sesión y no quería dejar pasar la oportunidad de presentar nuestras credenciales de compañía, así como explicaros nuestro procesos y futuras oportunidades. Como ves en esta transparencia que nos acaba de pasar Salva, Accenture es una compañía comprometida con el negocio responsable y la sostenibilidad. Somos una compañía, somos de las compañías, más diversas e inclusivas del mundo. Nos sentimos muy orgullosos de ello. Tiene más de 500.000 empleados a nivel global, 12.000 en España. Buscamos que los equipos sean diversos culturalmente, generacional, género, discapacidad, formación y origen. Como podéis ver, el 40.2% de las mujeres pertenecemos en España… somos mujeres. Y con un compromiso de paridad en el 2025. Más de 900 profesionales de España son de otras nacionalidades, concretamente 71. Y convivimos 4 generaciones distintas, con más de… 419 titulaciones diferentes.

Consideramos que la diversidad es clave para el éxito de la compañía, y desde el punto de vista social y de negocio. Una palanca muy importante de Accenture es la innovación y apostamos por ella. Tenemos 3.200 profesionales dedicados a la innovación. Y cuando hablamos de que apostamos, apostamos desde la investigación hasta la puesta en práctica, para generar valor a nuestros clientes.

Y hablando de nuestros clientes, una red global de clientes, de más de 6.000, en España 343, el 86% cotizan en el IBEX35 y 57 son de las 100 mayores empresas de España. El año pasado realizamos más de 2.800 proyectos distintos, de los cuales 1.200 fueron proyectos de nueva creación, a través de los 30 centros de solución que tenemos en España.

Y para concluir, somos una de las compañías que más contratan en España. El año poasado contratamos más de 2.700 profesionales en España y tenemos con un contrato indefinido el 97,4% de la plantilla. Compartir con vosotros, que además de las 2.700 contrataciones el año pasado, realizaron también prácticas con nosotros 1.300 estudiantes, tanto universitarios como FP, cuyo objetivo final es que una vez finalizan las prácticas y la evaluación de desempeño es positiva, incorporarles cuando han finalizado sus estudios.

Dudas, preguntas hasta ahora. Bueno.

Pues en cuanto a nuestros procesos de selección, animaos a enviarnos vuestras candidaturas. Comenzaríamos una primera preselección de candidatos según el perfil, os invitaríamos a realizar pruebas online para posteriormente invitaros a una dinámica de grupo y una entrevista personal. También deciros que durante esta etapa de confinamiento que estamos teniendo, continuamos realizando nuestros procesos con nuestros candidatos de manera online. Esperando y deseando que acabe esto cuanto antes.

Para concluir, aquellos que estéis interesados, podéis enviar vuestras candidaturas a través el correo electrónico eventos-recruiting@accenture.com y detallad en el asunto SE-ONLINEBIGD20. Lo podéis ver en la pantalla.

Pues muchas gracias a todos por asistir y deseando que todos podamos disfrutar cuanto antes, de nuestra vida anterior y que podamos tener la posibilidad de encontrarnos en un futuro. Gracias

Muchas gracias.

Gracias.

Gracias, hasta luego.