Historial de Versiones

Acá se van a detallar los diferentes cambios que se realicen al trabajo luego de cada entrega a fin de tener una mejor trazabilidad.

Versión	Cambios
1.0.0	Primer entrega.

Procesamiento Paralelo y Distribuido

Una respuesta a una creciente demanda de poder de procesamiento

Ramirez Ulises
Universidad Nacional de Misiones
ulisesrcolina@gmail.com

Resumen

En este breve trabajo se busca presentar, describir y experimentar con algunos de los desafíos que acompañan al gran volúmen de datos generados día a día y cómo el procesamiento paralelo y distribuido puede ayudar en el tratamiento de los mismos. Con respecto a la cátedra Paradigmas y Lenguajes de Programación, se puede establecer relación directa con la unidad 1 y 2, estas unidades se enfocan en diferentes problematicas y acercamientos en cuanto a los sistemas de procesamiento paralelo.

Palabras clave: *Multiproceso, programación concurrente, Procesamiento distribuido, Big Data.*

1. Introducción

El manejo de gran cantidad de datos ha ganado atención en los últimos años[1], esto surge a raíz del ritmo acelerado que se generan estos datos y los diferentes escenarios que surgen a la hora de asegurar[2, 3], procesar (lo que se discute en este documento, aunque este a su vez se basa en muchos que van ser citados) y almacenar[4-6] los mismos.

En la actualidad existen mas de 7.8 mil millones de habitantes en el planeta[7], un porcentaje considerable de esos habitantes son usuarios de internet como describe Meeker[8]. Aunque es cierto que la generación de datos viene tomando una tendencia creciente desde hace ya años [9], parte de este crecimiento abrupto de la última decada se puede atribuir, en parte, al gran papel que la tecnología fué adquiriendo con el pasar de los años y su constante evolución: comercio electrónico, publicidad electrónica, pagos electrónicos, etc.[8, 10], las personas conectadas con varios dispositivos (wearables, dispositivos IoT, celulares «especial énfasis», televisores, etc.), sensores produciendo una

ráfaga continua de datos, el avance y la gran disponibilidad de las redes móviles de cara a las redes de siguiente generación (5g)[11], dan como resultado una enorme cantidad de datos.

Esta enorme cantidad de datos acuño un nombre con el pasar de los años, el término es *big data*.

Shönberger y Culkier[12] brindan una síntesis concreta de lo que es big data, sin embargo, como los mismos autores lo mencionan, esto no es nada mas que el inicio.

> "Datos masivos, o cosas que se pueden hacer a gran escala, pero no a una escala inferior, con el fin de extraer nuevas percepciones o crear formas de valor de tal forma que se transformen los mercados."

Transformar estos datos en valor viene ya llevándose a cabo hace décadas, caracterizado con términos que fueron cambiando con el tiempo, conmunmente denominado *inteligencia de negocios* (BI) sobre el cual hay extensa literatura y gran variedad de escenarios en los que se aplicó el mismo con diferentes enfoques[13-19]. El big data, toma estos principios de la inteligencia de negocios y los aplica a una escala mayor.

Aquí, se explora el procesamiento del big data, los desafíos en cuanto a procesamiento que surgieron y siguen surgiendo (aunque también se mencionan brevemente algunas cuestiones relacionadas con los desafíos que intervienten con la seguridad y el almacenamiento de la información), los enfoques a la hora de tratar de encontrar valor a una cantidad de datos masivo teniendo en cuenta diferentes atributos con los que cuentan.

En la sección 2 se describe en mayor profundidad lo que significa el big data y cuales son sus características predominantes, es decir, cuando alguien considera algo como big data, en la sección 3 se introduce lo que es el procesamiento paralelo, se hace especial énfasis en cómo éste permite el tratamiento de los datos haciendo uso eficiente de todos los recursos. A partir de ahí, en la sección 4 se aplica esta idea de utilización eficiente de los recursos y se los traslada a un entorno mayor no limitado a una máquina y con un nivel de acoplamiento inferior.

2. Big Data

Definir *¿Qué es el big data?* es un trabajo no trivial, anteriormente, se brindó un concepto que pretendía dar al lector un primer acercamiento al tema, sin embargo este carece de completitud y especificidad. En la literatura se demuestra que términos difusos como 'grande' hacen aun más difícil el determinar qué es y qué no es big data, esto supone tener que definir variables que se van a tomar en cuenta en diferentes disciplinas con el fin de poder determinar '¿Qué tan grande tiene que ser algo para ser grande?'.

La caracterización principal del big data y por lo tanto la respuesta a la pregunta anterior se tiene en lo que la literatura llama *las 3 Vs*[1][25-27]. Antes

[1]Existen autores que consideran que son 4*V*s[20, 21], 5*V*s [22, 23], e incluso autores que consideran más *V*s [24]. Sin embargo, cabe destacar que estas dos opiniones incluyen las 3*V*s mencionadas anteriormente.

de continuar con las *V*s, es conveniente realizar una aclaración importante: En algunos casos la definición o caracterización aceptada del big data con las *V*s *no es aplicable*, por ejemplo, en el ámbito de la medicina, Baro *et. al* [28] determinan que la cantidad de datos necesarios para ser grande se satisface si se cumple que,

$$\log_{10}(n * p) \geq 7 \tag{1}$$

En donde se tiene que,

- n Cantidad de *individuos estadísticos*
- p Cantidad de variables a ser analizadas dentro del dataset

En este caso específico analiza el significado de ¿Qué es ser big data? para un área en particular, *la medicina*[2], aquí se busca enfocar y delimitar el concepto de grande al área de la salud. El motivo principal detrás de esto es que luego de estudiar los diferentes casos (entendiendo como *casos* las publicaciones en revistas científicas orientadas exclusivamente a la medicina)en un ambiente orientado a la salud, no se satisfacen los criterios de masividad[3], pero aún así existe en el corpus material acerca del big data asociado a la medicina, por lo que se hace útil poseer una definición mas ajustada a dicha disciplina.

2.1. Volúmen

El volúmen hace referencia a la cantidad de datos dentro de un dataset[21, 29], considerado por algunos autores como la característica que brinda los mayores desafíos para el tratamiento con el big data[30].

Se habla de cantidad, y nuevamente se tiene esta idea de *¿Qué tantos datos hacen falta para tener un volúmen lo suficientemente masivo como para ser considerado grande?*

Hacia el 2010, el mundo había creado 1ZB[4] de datos, se menciona esto para tener una idea aproximada de la cantidad de datos manejados hace DIEZ AÑOS ATRAS! (una eternidad en el mundo de las tecnologías de la información), y se estimaba que para el 2020 el mundo habría creado 40ZB[31], sin embargo, para 2018 ya se tenían 33ZB, lo cual llevó al IDC a realizar nuevas consideraciones y se terminó estimando que para 2025 se tendrían 175ZB[32]. Es decir, que en el tramo de 15 años, lo que se consideraba inconmensurable al inicio multiplicaría su tamaño por 175. Hablar de estas magnitudes se va naturalizando a medida que pasa el tiempo, ya en el 2013 se iba haciendo cotidiano hablar de Petabytes (PB) de cara al uso de Exabytes (EB)[30].

2.2. Variedad

Esta es otra de las características principales dentro del big data, y hace referencia a la variedad de datos que conforman un dataset. Este fenómeno tiene lugar por el simple hecho de que existen casi ilimitadas fuentes que pueden

[2]cabe destacar que no es posible asumir que este análisis llevado a cabo por los autores es trasladable a todas las disciplinas

[3]Comparado con la cantidad de consultas que recibe Google, la cantidad de reseñas escritas en Amazon, la cantidad de comentarios que se publican en Twitter, etc.

[4]NOTA: 1ZB equivale a 1×10^9 terabytes, es decir 1000000000TB.

contribuir a un dataset, el cual puede estar compuesto por diferentes tipos y formas de representaciones[30] lo cual hace al big data grande y estos pueden ser resumidos en estructurados, semi-estructurados y no-estructurados[33].

Los datos estructurados y los semi estructurados caben en los Sistemas de Administración de Bases de Datos (DBMS) y/o Data-Warehouses (DW), acá se mencionan los semi estructurados porque estos estan compuestos por archivos tales como XML, en donde se tienen etiquetas para separar diferentes elementos de datos. En cuanto a los datos no estructurados se puede mencionar a la web como una fuente importante de los mismos, un ejemplo primordial aquí es el denominado clickstream[34, 35], aunque menciones honorables van además para los mensajes de textos que se obtienen de las compañias de celulares, datos generados por sensores en dispositivos IoT, etc.

2.3. Velocidad

En la introducción se hizo mención de la utilidad de herramientas para análisis de datos, más específicamente, se habló de Almacenes de Datos o Data Warehouse/Enterprise Data Warehouse (DW/EDW), y su contribución superlativa a la Inteligencia de Negocios (BI) mencionada en la Sección 1.

Este sistema utilizado en BI es una respuesta para múltiples desafíos que se tienen dentro de una organización[36], en este documento se hace énfasis concretamente en uno de los motivos, el *para dar soporte a las desiciones que se toman a nivel estratégico*. Para lograr ese soporte, la organización sigue una metodología que le va a permitir aplicar estas herramientas analiticas con el fín de extraer el valor de los datos. La gente de Datalytics[5], menciona a lo largo de varias charlas que componen el ciclo de "DataSchool", la importancia de estas arquitecturas clásicas, sin embargo, el acercamiento clásico a la creciente cantidad de datos brinda más desafíos, particularmente el que más resuena en el contexto de este documento es el hecho de que esta arquitectura tradicional es *lenta para reaccionar*[37].

El concepto de velocidad es esto, y es tan importante como los que se hablaron anteriormente. La velocidad a la que se pueda realizar un análisis concreto y tomar decisiones basadas en los resultados es clave. El análisis de un fragmento de información acerca de la competencia, datos estadísticos realizados por algún organismo ajeno a la organización que tiene una periodicidad anual (ej: INDEC), etc. posee una gran potencialidad, sin embargo, si no se analiza oportunamente, el valor que pueda existir en esa información se pierde.

3. Procesamiento Paralelo

Diversos autores[38, 39] describen en profundidad diferentes modelos que extienden a la arquitectura tradicional de Von Neumann para lograr paralelización[6]. Esta se logra mediante la separación de una tarea T, en subtareas

[5]https://www.datalytics.com/

[6]No es el objetivo de este documento dar una descripción minuciosa de los diferentes modelos teóricos de paralelización. En el Anexo II, se pueden encontrar fuentes que hablen acerca del tema.

$T_1, T_2, \ldots, T_n$ (Diferentes autores adoptan este concepto con el término de *pipelining* [39, 40]). Luego para lograr el cometido, más de una de las T_n tareas debe realizarse al mismo tiempo, sin malinterpretar la realización de una tarea de manera muy velóz con realizarla al mismo tiempo que otras [40].

Las diferentes soluciones para procesamiento paralelo se desenvolvieron hasta llegar a presentar uno de los siguientes tres tipos de paralelismo[39]:

- Los **Sistemas de Memoria Compartida** que se componen de múltiples unidades de procesamiento unidas a una memoria.
- Los **Sistemas Distribuidos** que se componen de equipos con sus propias unidades de procesamiento y memoria, comunicados a traves de una conexión de red de alta velocidad. Este es el caso que nos compete en este documento y es tratado en la Sección 4.
- Las **Unidades de Procesamiento Gráfico** utilizadas como co-procesadores por su capacidad de paralelizar grandes cantidades de operaciones.

Diferentes enfoques existen a la hora de aplicar el pipelining mencionado anteriormente, y estos no son recientes sino que datan desde la década de 1990. Con el pasar del tiempo se fueron creando más enfoques y especializando los casos generales que se tenían en un principio. Ejemplo de algunas especializaciones son los enfoques que se ven en la cátedra de Paradigmas y Lenguajes de Programación[41], en donde se denomina a este problema *Descomposicion*, y de ahí surgen las diferentes especializaciones: Dominio, Funcion, Recursiva, Mixta.

Para este documento solamente se tienen en cuenta dos de las 3 que se proponen por Fountain [42]: Paralelización de Datos y Paralelización de Funciones.

La paralelización de datos y la paralelización de funciones puede encontrarse en diferentes niveles de la arquitectura del computador, algunos ejemplos van desde, el procesador como lo describe Hyde [42], pasando por el compilador [43], llegando a la aplicación [44], etc. Este documento cubre únicamente a casos que son aplicados a nivel de aplicación.

3.1. Paralelización de Datos

> La idea principal detrás de este enfoque a la hora de aplicar el *pipelining* consiste en que un programa secuencial puede ser transformado a un programa paralelo, realizando ejecuciones de copias idénticas de dicho programa como tareas separadas, a las cuales se les brinda simplemente parte de los datos iniciales [45].

En la figura 1 se presenta una aplicación que está corriendo sobre un núcleo aleatorio en un ordenador. En este ejemplo no se tiene presente la paralelización, D representa los datos sobre los cuales está trabajando la *Aplicación*.

En la figura 2 se presenta la misma aplicación (esta vez se la denomina *app*). En este ejemplo los datos representados por D anteriomente, se dividien en D_n secciones más pequeñas que son distribuidas a distintos núcleos de procesamiento [7]. Una copia idéntica de la aplicación (Figura 1) se encuentra

[7] en el ejemplo se utilizan 4, aunque pueden ser más núcleos dentro de un procesador

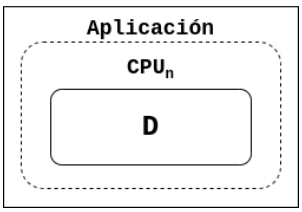

Figura 1: Aplicación sin paralelización

corriendo en cada uno de los cuatro núcleos del ordenador, cada una de estas copias idénticas se encuentra trabajando con datos distintos (ya que a cada una de las copias le toca una sección D_n diferente).

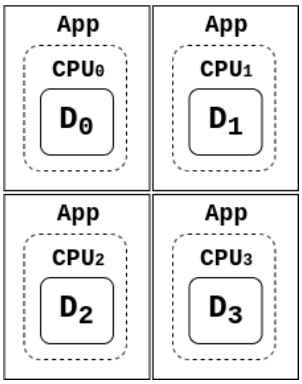

Figura 2: Aplicación con paralelización de datos

Una interpretación alternativa se puede ver en la figura 3, este gráfíco se puede leer como "Una aplicación que se encuentra corriendo sobre distintos nucleos, con distintos fragmentos de los datos originales"

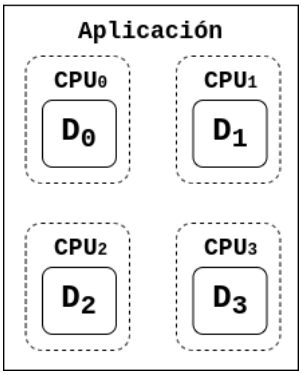

Figura 3: Aplicación con paralelización de datos - Representación alternativa

En la bibliografía existen implementaciones en diferentes sub áreas de las Ciencias de la Computación con este enfoque presente, el procesamiento de imágenes [46], el manejo de bases de datos [47] y la inteligencia artificial [48] por nombrar algunos ejemplos.

Para un ejemplo simple y concreto se propone lo siguiente:

Se presenta la tarea de encontrar el valor mínimo dentro de un arreglo $A = [a_0, a_1, \ldots, a_{n-2}, a_{n-1}]$.

Se puede proceder sin aplicar paralelización de datos, y realizar una aplicación que recorra los n elementos del arreglo, realice las comparaciones necesarias y devuelva el valor mínimo, o se puede optar por la opción con paralelización de datos, una solución con este enfoque será fraccionar el arreglo con A en, por ejemplo, 2 partes A_1 y A_2. Luego se procede a distribuir estas dos partes A_1 y A_2 a diferentes instancias de la aplicación que calcula el mínimo, de esta manera se tendrán dos núcleos de procesamiento que trabajarán en un problema de menor tamaño (ya que cada uno va a trabajar únicamente con el 50 % de los datos), y al final solamente hará falta realizar una comparación entre los mínimos que encuentren las dos instancias de la aplicación.

Cabe destacar que algunos problemas pueden surgir a la hora de particionar datos si esto se hace sin nignun análisis, un ejemplo de esto puede ser considerado en el escenario en donde ocurre la división de una imagen en partes más pequeñas para su análisis en forma paralela [49]. También se hace presente la limitación que impone la ley de Amdhal[8].

3.2. Paralelización de Funciones

Este enfoque a la paralelización toma otro camino a la hora de aplicar el *pipelining*, este acercamiento no fracciona los datos sino que transforma la tarea a un gráfo de dependencias en donde cada nodo corresponde a una operación a ser realizada para cumplir la tarea. Y lo que trata de hacer es ejecutar la mayor cantidad de tareas al mismo tiempo [51-54].

Esta idea de distribuir los datos a diferentes unidades funcionales que se encarguen de realizar una tarea sobre los datos mencionados se presenta en la figura 4.

Aquí se puede observar que lo que se subdivide no son los datos, sino que lo que se está subdividiendo es la funcionalidad que compone a la tarea T, aquí cada o_n es una operación claramente delimitada dentro de la aplicación, y D representa a los datos con los que se va a trabajar.

Un ejemplo concreto para este enfoque se presenta a continuación:

Dada una imagen, se presenta la tarea de determinar si en dicha imagen se encuentra presente algún rostro, vehículo, gato o planta.

Por lo que se puede decir que la tarea T se compone de la siguiente manera

$$T = [o_0, o_1, o_2, o_3] \tag{2}$$

En donde se tiene que,

[8] "El incremento máximo de velocidad (la cantidad máxima de procesadores que pueden ser utilizados de manera efectiva) es la inversa de la fracción de tiempo que la tarea toma para finalizar en un solo hilo" [50]

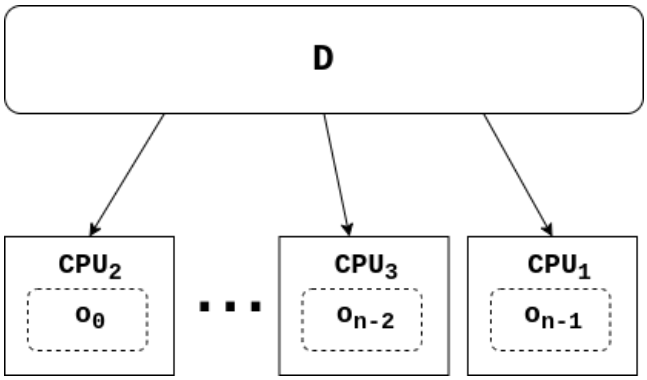

Figura 4: Aplicación con paralelización de funciones

o_0 Buscar rostro
o_1 Buscar vehículo
o_2 Buscar planta
o_3 Buscar gato

Cada o_n en la ecuación 2 representa un nodo del gráfo de mencionado anteriormente. Ahora toca analizar y determinar cuales de las o_n operaciones se pueden realizar al mismo tiempo.

Una cuestion esencial en este ejemplo es identificar el hecho de que ninguna operación o_n depende de alguna operación. Esto es visible a la hora de realizar diversas preguntas con respecto al contexto,

- ¿Es necesario determinar si existe o no un rostro para poder iniciar el análisis y determinar si la imagen contiene un vehiculo, gato o planta? No, entonces, *Buscar rostro* es independiente a las demás operaciones.
- ¿Es necesario determinar si existe un vehiculo en la imagen para poder iniciar el análisis y determinar si la imagen contiene un rostro, gato o planta? No, entonces, *Buscar vehículo* es independiente a las demás operaciones.

El análisis anterior debe continuar hasta que se analicen las o_n operaciones. Habrán veces en donde dos o más operaciones no podrán ser ejecutadas de manera paralela, esto puede darse porque una (también pueden ser varias) de ellas depende de los resultados que se obtengan de otra u otras operaciones.

En el caso del ejemplo para la sección, esta apreciación y el análisis que se realizó permite definir que todas las operaciones que componen a la tarea T se pueden ejecutar de manera paralela por el mismo dato.

4. Procesamiento Distribuido

Este tiene muchas similitudes con los sistemas de Procesamiento Paralelo, es más, es válido decir que los Sistemas Paralelos y los Sistemas Distribuidos son un entrelazamiento de redes, que tienen la caracteristica de ser continuas[9],

[9] o un continuum de redes. `https://dictionary.cambridge.org/es/diccionario/ingles/continuum`

lo que significa que el parámetro que determina si el sistema es distribuido es el promedio de distancias entre los nodos de procesamiento [40]. El desafío que propone Hyde, y también diferencia a los dos sistemas es el de poder determinar el estado exacto de todo el sistema. Este es un desafío no-trivial, ya que dicho estado no puede ser computado instantaneamente por el hecho de que cada operación dentro de la red podría requerir travesías por la misma.

Referencias

[1] *NewVantage Partners Big Data and AI Executive Survey 2019*. 2019. URL: https://www.tcs.com/content/dam/tcs-bts/pdf/insights/Big-Data-Executive-Survey-2019-Findings-Updated-010219-1.pdf (visitado 16-06-2021).

[2] Danda B. Rawat, Ronald Doku y Moses Garuba. «Cybersecurity in Big Data Era: From Securing Big Data to Data-Driven Security». En: *IEEE Transactions on Services Computing* (2019). ISSN: 1939-1374. DOI: 10.1109/TSC.2019.2907247.

[3] Murat Kantarcioglu. «Securing Big Data: New Access Control Challenges and Approaches». En: *Proceedings of the 24th ACM Symposium on Access Control Models and Technologies*. New York, NY, USA. DOI: 10.1145/3322431.3326330.

[4] M H Padgavankar y Dr S R Gupta. «Big Data Storage and Challenges». En: 5 (2014), pág. 6.

[5] Rajeev Agrawal y Christopher Nyamful. «Challenges of big data storage and management». En: *Global Journal of Information Technology: Emerging Technologies* 6.1 (). ISSN: 2301-2617. DOI: 10.18844/gjit.v6i1.383.

[6] Hongming Cai y col. «IoT-Based Big Data Storage Systems in Cloud Computing: Perspectives and Challenges». En: *IEEE Internet of Things Journal* 4.1 (2017). ISSN: 2327-4662. DOI: 10.1109/JIOT.2016.2619369.

[7] Worldometer. *Current World Population*. 11 de jun. de 2021. URL: https://www.worldometers.info/world-population/.

[8] Mary Meeker. *Internet Trends 2019*. 2019. URL: https://bit.ly/3pPfE2L (visitado 11-06-2021).

[9] Peter Lyman y Hal R. Varian. *How Much Information?* 2003. URL: https://www2.sims.berkeley.edu/research/projects/how-much-info-2003/ (visitado 04-07-2021).

[10] Miquel Pellicer. *Las claves del informe 'Internet Trends 2019' de Mary Meeker*. 2019. URL: https://miquelpellicer.com/2019/06/claves-informe-internet-trends-2019-mary-meeker/ (visitado 11-06-2021).

[11] Kashif Sultan, Hazrat Ali y Zhongshan Zhang. «Big Data Perspective and Challenges in Next Generation Networks». En: *Future Internet* 10.7 (2018). DOI: 10.3390/fi10070056.

[12] Viktor Mayer-Shönberger y Kenneth Cukier. *Big data. A Revolution that will transform how we Live, Work and Think*. Boston, Massachusetts: Houghton Mifflin Harcourt, 2016. ISBN: 0544227751.

[13] Paola Verónica Britos. «Procesos de Explotación de Información Basados en Sistemas Inteligentes». Tesis doct. Universidad Nacional de La Plata, 2008. 234 págs. DOI: 10.35537/10915/4142. URL: http://sedici.unlp.edu.ar/handle/10915/4142.

[14] Atika Qazi y col. «Enhancing Business Intelligence by Means of Suggestive Reviews». En: *The Scientific World Journal* 2014 (). DOI: 10.1155/2014/879323. URL: https://www.hindawi.com/journals/tswj/2014/879323/.

[15] Yanhui Su, Per Backlund y Henrik Engström. «Business Intelligence Challenges for Independent Game Publishing». En: *International Journal of Computer Games Technology* 2020 (). DOI: 10.1155/2020/5395187. URL: https://www.hindawi.com/journals/ijcgt/2020/5395187/.

[16] Mohamed O. Khozium y Norah S. Farooqi. «Cooperative Business Intelligence Model Using a Multiagent Platform». En: *Scientific Programming* 2020 (). DOI: 10.1155/2020/8898719. URL: https://www.hindawi.com/journals/sp/2020/8898719/.

[17] Cempírek Václav y col. «Utilization of Business Intelligence Tools in Cargo Control». En: *Transportation Research Procedia* 53 (). ISSN: 2352-1465. DOI: 10.1016/j.trpro.2021.02.028. URL: https://www.sciencedirect.com/science/article/pii/S2352146521001885.

[18] Gloria Phillips-Wren, Mary Daly y Frada Burstein. «Reconciling business intelligence, analytics and decision support systems: More data, deeper insight». En: *Decision Support Systems* 146 (). DOI: 10.1016/j.dss.2021.113560. URL: https://www.sciencedirect.com/science/article/pii/S0167923621000701.

[19] Florian Schwade. «Social Collaboration Analytics Framework: A framework for providing business intelligence on collaboration in the digital workplace». En: *Decision Support Systems* (). ISSN: 0167-9236. DOI: 10.1016/j.dss.2021.113587. URL: https://www.sciencedirect.com/science/article/pii/S016792362100097X.

[20] Satriyo Wibowo y Tesar Sandikapura. «Improving Data Security, Interoperability, and Veracity using Blockchain for One Data Governance, Case Study of Local Tax Big Data». En: *2019 International Conference on ICT for Smart Society (ICISS)*. Vol. 7. ISSN: 2640-0545, págs. 1-6. DOI: 10.1109/ICISS48059.2019.8969805.

[21] Maryam Ghasemaghaei y Goran Calic. «Does big data enhance firm innovation competency? The mediating role of data-driven insights». En: 104 (), págs. 69-84. ISSN: 0148-2963. DOI: 10.1016/j.jbusres.2019.07.006.

[22] Pascal Hitzler y A. Krzysztof Janowicz. *Linked Data, Big Data, and the 4th Paradigm Editorial*. 2013. URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.675.9076%5C&rep=rep1%5C&type=pdf.

[23] Shen Yin y Okyay Kaynak. «Big Data for Modern Industry: Challenges and Trends [Point of View]». En: *Proceedings of the IEEE* 103.2 (), págs. 143-146. ISSN: 1558-2256. DOI: 10.1109/JPROC.2015.2388958.

[24] Stephen Bonner y col. «Chapter 14 - Exploring the Evolution of Big Data Technologies». En: *Software Architecture for Big Data and the Cloud*. Ed. por Ivan Mistrik y col. Boston: Morgan Kaufmann, 1 de ene. de 2017, págs. 253-283. ISBN: 978-0-12-805467-3. DOI: 10.1016/B978-0-12-805467-3.00014-4. URL: https://www.sciencedirect.com/science/article/pii/B9780128054673000144.

[25] *Big Data Analytics*. URL: https://www.ibm.com/analytics/hadoop/big-data-analytics (visitado 12-07-2021).

[26] *¿Qué es big data? – Amazon Web Services (AWS)*. Amazon Web Services, Inc. URL: https://aws.amazon.com/es/big-data/what-is-big-data/ (visitado 12-07-2021).

[27] *What Is Big Data? | Oracle*. URL: https://www.oracle.com/big-data/what-is-big-data/ (visitado 12-07-2021).

[28] Emilie Baro y col. «Toward a Literature-Driven Definition of Big Data in Healthcare». En: *BioMed Research International* 2015 (). DOI: 10.1155/2015/639021.

[29] Maryam Ghasemaghaei. «Understanding the impact of big data on firm performance: The necessity of conceptually differentiating among big data characteristics». En: *International Journal of Information Management* 57 (). ISSN: 0268-4012. DOI: 10.1016/j.ijinfomgt.2019.102055.

[30] Dunren Che, Mejdl Safran y Zhiyong Peng. «From Big Data to Big Data Mining: Challenges, Issues, and Opportunities». En: vol. 7827. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, págs. 1-15. ISBN: 978-3-642-40269-2 978-3-642-40270-8. DOI: 10.1007/978-3-642-40270-8_1.

[31] Son K. Lam y col. «Leveraging Frontline Employees' Small Data and Firm-Level Big Data in Frontline Management: An Absorptive Capacity Perspective». En: *Journal of Service Research* 20.1 (), págs. 12-28. ISSN: 1094-6705, 1552-7379. DOI: 10.1177/1094670516679271.

[32] Gil Press. *6 Predictions About Data In 2020 And The Coming Decade*. Forbes. URL: https://www.forbes.com/sites/gilpress/2020/01/06/6-predictions-about-data-in-2020-and-the-coming-decade/ (visitado 15-07-2021).

[33] Senem Sagiroglu y Duygu Sinanc. «Big data: A review». En: 2013, págs. 42-47. ISBN: 978-1-4673-6403-4. DOI: 10.1109/CTS.2013.6567202.

[34] Philip Russom. *Big Data Analytics*. 2011.

[35] Bill Albert, Tom Tullis y Donna Tedesco. «Beyond the Usability Lab. Chapter 5 - Data Preparation». En: (2010). URL: https://www.sciencedirect.com/book/9780123748928/beyond-the-usability-lab.

[36] Datalytics. *Datalytics DataSchool. Arquitectura de Datos basadas en Data Warehouse*. 10 de sep. de 2020.

[37] Datalytics. *Datalytics DataSchool. Arquitectura de Datos Modernas*. 17 de sep. de 2020. URL: https://www.youtube.com/watch?v=wWNZa-Ez8IU.

[38] Dan I. Moldovan. *Parallel Processing. From Applications to Systems*. San Mateo, California: Morgan Kaufmann Publishers, 1993.

[39] Roman Trobec y col. *Introduction to Parallel Computing. From Algorithms to Programming on State-of-the-Art Platforms*. Switzerland: Springer Nature Switzerland, 2018. DOI: 10.1007.978-3-319-9883307_1.

[40] D. C. Hyde. *Introduction to the Principles of Parallel Computation. Chapter 1: Introduction*. Lewisburg, PA 17837: Department of Computer Science, Bucknell University, 1998.

[41] Germán A. J. Pautsch y Esteban R. Martini. *Paradigmas y Lenguajes de Programación - Unidad II - Metodologías de Programación Paralela*. 2019.

[42] Terry J. Fountain. *Parallel Computing. Principles and Practice*. Cambridge: Cambridge University Press, 1994.

[43] Terry W. Clark, Reinhard v. Hanxleden y Ken Kennedy. «Experiences in Data-Parallel Programming». En: *Scientific Programming* 6.1 (1997), págs. 153-158. ISSN: 1058-9244. DOI: 10.1155/1997/260463.

[44] Lex Wolters, Gerard Cats y Nils Gustafsson. «Data-Parallel Numerical Weather Forecasting». En: *Scientific Programming* 4.3 (1995), págs. 141-153. ISSN: 1058-9244. DOI: 10.1155/1995/692717.

[45] Magne Haveraaen. «Machine and Collection Abstractions for User-Implemented Data-Parallel Programming». En: *Scientific Programming* 8.4 (2000), págs. 231-246. ISSN: 1058-9244. DOI: 10.1155/2000/485607.

[46] Yi Pang, Lifeng Sun y Shiqiang Yang. «Data parallelization of Kd-tree ray tracing on the Cell Broadband Engine». En: *2009 IEEE International Conference on Multimedia and Expo*. 2009 IEEE International Conference on Multimedia and Expo. ISSN: 1945-788X. Jun. de 2009, págs. 1246-1249. DOI: 10.1109/ICME.2009.5202727.

[47] Victor Zakharov y col. «Architecture of Software-Hardware Complex for Searching Images in Database». En: *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*. 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). ISSN: 2376-6565. Ene. de 2019, págs. 1735-1739. DOI: 10.1109/EIConRus.2019.8657241.

[48] Nadine Hajj, Yara Rizk y Mariette Awad. «A MapReduce Cortical Algorithms Implementation for Unsupervised Learning of Big Data». En: *Procedia Computer Science* 53 (2015), págs. 327-334. ISSN: 18770509. DOI: 10.1016/j.procs.2015.07.310. URL: https://linkinghub.elsevier.com/retrieve/pii/S187705091501813X (visitado 26-07-2021).

[49] T. Oshitani y T. Watanabe. «Parallel map recognition with information propagation mechanism». En: *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR '99 (Cat. No.PR00318)*. Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR '99 (Cat. No.PR00318). Sep. de 1999, págs. 717-720. DOI: 10.1109/ICDAR.1999.791888.

[50] David P. Rodgers. «Improvements in multiprocessor system design». En: *ACM SIGARCH Computer Architecture News* 13.3 (1 de jun. de 1985), págs. 225-231. ISSN: 0163-5964. DOI: 10.1145/327070.327215. URL: https://doi.org/10.1145/327070.327215.

[51] Xiaofu Meng y col. «Luminance and Chrominance Parallelization of H.264/AVC Decoding on a Multi-core Processor». En: *2013 IEEE Eighth International Conference on Networking, Architecture and Storage*. 2013 IEEE Eighth International Conference on Networking, Architecture and Storage. Jul. de 2013, págs. 252-256. DOI: 10.1109/NAS.2013.39.

[52] Mengjie Liu y col. «Joint Two-Tier Network Function Parallelization on Multicore Platform». En: *IEEE Transactions on Network and Service Management* 16.3 (sep. de 2019), págs. 990-1004. ISSN: 1932-4537. DOI: 10.1109/TNSM.2019.2920012.

[53] Jianhong Zhou, Gang Feng y Yi Gao. «Network Function Parallelization for High Reliability and Low Latency Services». En: *IEEE Access* 8 (2020), págs. 75894-75905. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2988719.

[54] I-Chieh Lin, Yu-Hsuan Yeh y Kate Ching-Ju Lin. «Toward Optimal Partial Parallelization for Service Function Chaining». En: *IEEE/ACM Transactions on Networking* (2021), págs. 1-12. ISSN: 1558-2566. DOI: 10.1109/TNET.2021.3075709.

Siglas

BI Business Intelligence. 3, 5

DBMS Data Base Management Systems. 5
DW Data Warehouse. 5

EB Exabyte. 4
EDW Enterprise Data Warehouse. 5

INDEC Instituto Nacional De Estadística y Censos. 5
IoT Internet of Things. 2, 5

PB Petabyte. 4

TB Terabyte. 4

ZB Zettabyte. 4

Glosario

clickstream *Flujo de clicks*, es una bitacora detallada de como los usuarios navegan una página web al realizar una tarea. 5

continuum *Continuo*, algo que cambia gradualmente o en pequeños incrementos sin ningún pico evidente. 9

dataset *Conjunto de datos*, sobre los cuales se realizan experimentos. Las conclusiones que los investigadores definan sobre un cierto tema, se da mediante los experimentos realizados sobre uno o más conjuntos de datos. 4, 5

pipelining En Ciencias de la Computación, este hace referencia a una organización en la cual pasos sucesivos de una secuencias de instrucciones son ejecutadas por diferentes módulos, esto para que otra instrucción pueda iniciar antes que una instrucción anterior finalice. 6

wearables *Vestible*, En el contexto de la tecnología, hace referencia a un dispositivo que se pueda *vestir*. Ej: relojes inteligentes. 2

Anexo I: Terabytes ⇒ Petabytes ⇒ Zettabytes

A veces, tratar magnitudes tan extremas no permite transmitir de manera comprensible lo masivo o lo ínfimo de tales magnitudes, esto es frecuentemente el caso al hablar de cantidad de datos tales como los Terabytes, Petabytes, Zettabytes o Exabytes.

Si el lector es una persona que esta familiarizada con los medios de almacenamiento y su tamaño, resulta sencilla la explicación de que un Terabyte es varias veces mayor que un Gigabyte, esto puede deberse a que el Gigabyte es una unidad que se utiliza cotidianamente, por ejemplo, el tamaño de los pendrives, la memoria del celular, el tamaño de discos en ordenadores, etc. Sin embargo, esta explicación puede volverse un desafío si se intenta realizar la explicación a una persona que no posee conocimiento alguno acerca de este tema.

En este anexo, se propone un ejemplo para que el lector (independientemente de su nivel en el uso de tecnología) se familiarice y tenga una mejor concepción la masividad a la hora de hablar del salto Petabytes y Zettabytes[10], en este ejemplo no se van a utilizar conceptos de las ciencias de la computación, se va a tratar con segundos, minutos, horas, días y años, conceptos que se puede asumir son familiares a todo lector.

[10]Este salto de magnitudes es equivalente a los saltos que se dan del Byte, hasta el Gigabyte (B→KB→MB→GB). Cada salto es 1024 veces mayor comparado a la magnitud anterior.