Hadoop, MapReduce, Lustre P2P, BitTorrent

Aguilar Maya Daniel [*] **López Ortega Diego** [**]
Mancera Hernández Eduardo Abdiel [***]
Rojas Terrazas Laylet [****]

[*] *Facultad de Ingenieria, UNAM, Ing. en Computación, daniel.aguilar@ingenieria.unam.edu*
[**] *Facultad de Ingenieria, UNAM, Ing. en Computación, diego.lopez@ingenieria.unam.edu*
[***] *Facultad de Ingenieria, UNAM, Ing. en Computación, eduardo.mancera@ingenieria.unam.edu*
[****] *Facultad de Ingenieria, UNAM, Ing. en Computación, laylet.rojas@fi.unam.edu.com*

Resumen: The present work provides insights into various facets of distributed systems, exploring topics such as Hadoop, MapReduce, Lustre, P2P, and BitTorrent. Through an examination of the historical context and practical applications of each concept, the work aims to elucidate key theoretical notions associated with these distributed systems. Special attention will be given to the pivotal role of Hadoop in handling Big Data, accentuating how MapReduce contributes to this intricate task. Additionally, Lustre's significance in the realm of distributed file systems will be underscored. The discussion delves into the decentralized architecture of Peer-to-Peer (P2P) computing, where nodes collaborate seamlessly through direct communication and resource sharing, eliminating the need for a central server. P2P systems showcase attributes such as scalability, resilience, and autonomy, with each node serving both as a consumer and provider of resources. While widely applied in domains like file sharing, content distribution, and communication platforms, P2P computing poses challenges related to security, trust, and resource management efficiency. Despite these challenges, the ongoing evolution of P2P technology has left a lasting impact on distributed systems, shaping the development of applications and services that harness the collective strength of interconnected peers.

Palabras clave: P2P, Big Data, MapReduce, Map, Reduce, Clúster, Lustre, P2P, Napster, Middleware, BitTorrent, BOINC, Hadoop, YARN, HDFS

1. INTRODUCCIÓN

El presente documento proporciona una visión exhaustiva sobre diversos sistemas distribuidos y su papel en el panorama del cómputo distribuido. Desde la infraestructura de Hadoop y MapReduce para el procesamiento masivo de datos hasta la robusta solución de almacenamiento de archivos de Lustre, se exploran tecnologías clave que abordan desafíos en escalabilidad y rendimiento. Además, se analiza el paradigma P2P y sus aplicaciones, incluido el protocolo BitTorrent, que revolucionó la distribución descentralizada de archivos.

El documento también destaca la importancia del cómputo voluntario, ilustrado por sistemas como BOINC y Folding@Home, que aprovechan el poder colectivo de los usuarios para la investigación científica. Estos sistemas ejemplifican la diversidad de enfoques dentro del cómputo distribuido y su impacto en una amplia gama de aplicaciones, desde el procesamiento de datos hasta la investigación médica y científica. Finalmente se destaca el papel de la Facultad de Ingeniería de la Universidad Autónoma de México con su participación en proyectos BOINC.

2. HADOOP

2.1 Historia

Con el crecimiento exponencial de la World Wide Web a finales del siglo XX y principios del XXI, la necesidad de herramientas para la recuperación eficiente de información se hizo patente. Los primeros buscadores e índices, inicialmente gestionados manualmente, se vieron rápidamente superados por la magnitud de la red. La explosión de la Web, de docenas a millones de páginas, impulsó la automatización de los procesos de búsqueda. Los rastreadores web, desarrollados principalmente en proyectos universitarios, sentaron las bases para las primeras compañías de búsqueda, como Yahoo y AltaVista.

En este contexto, surgió Nutch, un proyecto de código abierto liderado por Doug Cutting y Mike Cafarella. Su objetivo era revolucionar la velocidad de búsqueda mediante la distribución de datos y cálculos en múltiples computadoras, permitiendo el procesamiento simultáneo de tareas. Cabe destacar que, de forma paralela, se desarrollaba otro proyecto de buscador: Google. Este compartía la filosofía de Nutch, basándose en el almacenamiento y

procesamiento distribuido de datos para ofrecer resultados de búsqueda más rápidos.

La evolución de los buscadores web ha sido un proceso continuo de innovación y adaptación a las necesidades de los usuarios. La automatización, la distribución y el procesamiento paralelo de datos han sido claves para mejorar la velocidad y la precisión de la búsqueda, sin embargo, aún hay desafíos por delante, como la gestión de la información no estructurada, la búsqueda semántica y la personalización de los resultados. El futuro de la búsqueda web dependerá de la capacidad de los investigadores y desarrolladores para abordar estos retos y ofrecer experiencias de búsqueda cada vez más eficientes y relevantes.

En 2006, Doug Cutting se incorporó a Yahoo, llevando consigo el proyecto Nutch, así como ideas derivadas de los trabajos pioneros de Google en la automatización del almacenamiento y procesamiento distribuido de datos. El proyecto Nutch se bifurcó, manteniendo la sección de rastreo web como Nutch y transformando la sección de computación y procesamiento distribuido en Hadoop, nombre inspirado en el elefante de juguete del hijo de Cutting. En 2008, Yahoo liberó Hadoop como proyecto de código abierto.

Actualmente, la estructura y el ecosistema de tecnologías de Hadoop son gestionados y mantenidos por la Apache Software Foundation (ASF), una comunidad global sin fines de lucro integrada por programadores de software y otros colaboradores.

2.2 Funcionamiento

Funciona en un modelo de programación de "divide y conquista", donde los datos se dividen en bloques más pequeños y se distribuyen en diferentes nodos del clúster para su procesamiento paralelo. Supongamos que tenemos un conjunto de datos muy grande que deseamos analizar para encontrar información específica. Este conjunto de datos se almacena en un clúster de servidores que ejecutan el software de Hadoop. [16]

- **División de datos:** Primero, Hadoop divide el conjunto de datos en bloques más pequeños. Por ejemplo, si tenemos un archivo de 1 terabyte, Hadoop podría dividirlo en bloques de 128 megabytes cada uno.
- **Distribución de datos:** Luego, estos bloques de datos se distribuyen en diferentes nodos del clúster. Cada nodo es responsable de almacenar y procesar uno o más bloques de datos.
- **Procesamiento paralelo:** Una vez que los datos están distribuidos en el clúster, Hadoop ejecuta el proceso de procesamiento paralelo. Esto significa que múltiples nodos del clúster trabajan simultáneamente en el procesamiento de los datos.
- **Recopilación de resultados:** Después de que todos los nodos hayan completado su procesamiento, Hadoop recopila los resultados parciales de cada nodo y los combina para generar el resultado final. Esto se realiza de manera eficiente mediante el uso de un componente llamado reducción.[en] el modelo MapReduce de Hadoop.
- **Almacenamiento de resultados:** Finalmente, los resultados finales se almacenan en el sistema de archivos distribuido de Hadoop, como Hadoop Distributed File System (HDFS), para su posterior análisis o uso.

Un ejemplo práctico de cómo funciona Hadoop sería el procesamiento de grandes conjuntos de datos para realizar análisis de datos, como el análisis de registros de servidor web para identificar patrones de tráfico, o el análisis de datos de redes sociales para detectar tendencias y comportamientos de los usuarios. Mediante la distribución y el procesamiento paralelo de datos en un clúster de computadoras, Hadoop permite realizar estas tareas de manera eficiente y escalable.

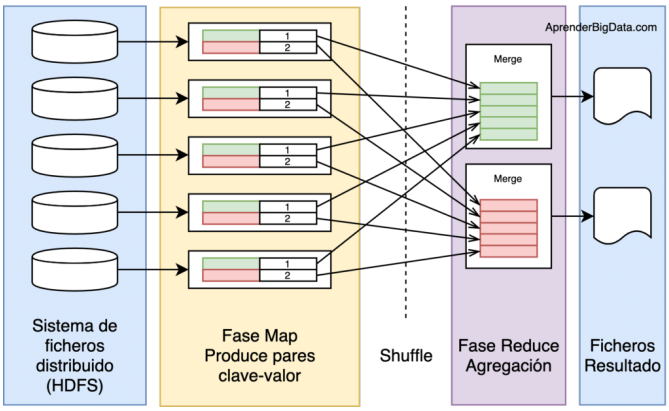

Figura 1. Arquitectura Hadoop

2.3 Utilización en la actualidad

Hadoop se utiliza actualmente en una amplia gama de aplicaciones en diversos sectores. A continuación, se detallan algunos ejemplos:

Análisis de datos a gran escala

- **Empresas de Telecomunicaciones:** Hadoop se utiliza para analizar registros de llamadas, datos de red y patrones de tráfico para optimizar la infraestructura y mejorar la experiencia del cliente.
- **Empresas Financieras:** Hadoop se utiliza para analizar datos de transacciones, detectar fraudes y evaluar riesgos crediticios.
- **Empresas de comercio electrónico:** Hadoop se utiliza para analizar el comportamiento del cliente, personalizar las recomendaciones de productos y optimizar las campañas de marketing.

Procesamiento de datos en tiempo real

- **Redes Sociales:** Hadoop se utiliza para procesar y analizar flujos de datos en tiempo real de publicaciones, comentarios y Me gusta.
- **Sistemas y Dispositivos IoT:** Hadoop se utiliza para ingerir, almacenar y analizar datos en tiempo real de sensores y dispositivos IoT.
- **Sistemas de detección de fraudes:** Hadoop se utiliza para analizar datos de transacciones en tiempo real para detectar fraudes de forma instantánea.

Almacenamiento de datos a gran escala

- **Archivos de Datos Históricos:** Hadoop se utiliza para almacenar grandes conjuntos de datos históricos, como registros médicos, imágenes y videos.

- **Data Lakes:** Hadoop se utiliza para crear Data Lakes, repositorios centralizados de datos sin procesar para análisis posteriores.
- **Repositorios de Datos Científicos:** Hadoop se utiliza para almacenar y analizar grandes conjuntos de datos científicos, como datos climáticos y genómicos.

Aprendizaje autómatico de Inteligencia Artificial

- **Entrenamiento de modelos:** Hadoop se utiliza para distribuir el entrenamiento de modelos de aprendizaje automático a gran escala.
- **Análisis predictivo:** Hadoop se utiliza para realizar análisis predictivos utilizando modelos de aprendizaje automático.
- **Procesamiento del lenguaje natural:** Hadoop se utiliza para procesar grandes cantidades de texto para aplicaciones de análisis de sentimiento y extracción de entidades.

2.4 Principal Industria

Hadoop está dirigido a una amplia gama de industrias que necesitan gestionar grandes volúmenes de datos de manera eficiente y escalable. Algunos ejemplos de industrias que suelen beneficiarse del uso de Hadoop incluyen:

Tecnología y medios de comunicación
Empresas de tecnología, plataformas de redes sociales, empresas de publicidad en línea, plataformas de streaming, entre otros, que manejan grandes cantidades de datos de usuarios y necesitan almacenar, procesar y analizar estos datos de manera eficiente.

Finanzas
Instituciones financieras como bancos, firmas de inversión, aseguradoras, que necesitan analizar grandes conjuntos de datos financieros para identificar tendencias, gestionar riesgos y optimizar sus operaciones.

Salud
Organizaciones de atención médica, hospitales, compañías farmacéuticas, que recopilan y analizan grandes volúmenes de datos de pacientes, registros médicos electrónicos, resultados de pruebas, etc., para mejorar la atención al paciente, la investigación médica y la gestión hospitalaria.

Retail y Comercio Electrónico
Tiendas minoristas, empresas de comercio electrónico y cadenas de suministro que recopilan datos de ventas, inventario, comportamiento del cliente, entre otros, para mejorar la experiencia del cliente, la gestión de inventario y la toma de decisiones comerciales.

Telecomunicaciones
Proveedores de servicios de telecomunicaciones que necesitan analizar grandes volúmenes de datos de red, como registros de llamadas, datos de tráfico, para optimizar la infraestructura de red, mejorar la calidad del servicio y ofrecer servicios personalizados.

Energía y recursos naturales
Compañías en sectores como la exploración de petróleo y gas, minería, energía renovable, que recopilan y analizan grandes volúmenes de datos de sensores, equipos, etc., para mejorar la eficiencia operativa, la seguridad y la toma de decisiones. [28]

3. MAPREDUCE

3.1 Historia

El contexto histórico de MapReduce se remonta a principios de la década de 2000, cuando Google estaba lidiando con el desafío de procesar grandes volúmenes de datos generados por su motor de búsqueda y otros servicios en línea. En ese momento, la cantidad de datos que Google estaba recopilando y procesando crecía exponencialmente, lo que planteaba desafíos significativos en términos de almacenamiento y procesamiento eficiente de esos datos.

Para abordar este desafío, Google desarrolló un modelo de programación llamado MapReduce, que se basaba en los principios de programación funcional y procesamiento distribuido. La inspiración detrás de este algoritmo se encuentra en la programación funcional y las funciones "mapƨ reduce", que son inherentes a lenguajes como Lisp y Scheme. Jeff Dean y Sanjay Ghemawat tomaron prestadas estas ideas y las adaptaron al procesamiento de datos distribuidos en clústeres de servidores. En lugar de procesar datos en una única máquina, MapReduce permitió la distribución del trabajo en múltiples servidores, lo que aceleró significativamente el tiempo de procesamiento

La clave del éxito de MapReduce radicaba en su simplicidad y capacidad para escalar horizontalmente en clústeres de computadoras estándar, lo que permitía procesar petabytes de datos en miles de servidores. Además, MapReduce se diseñó para tolerar fallos de hardware y proporcionar una abstracción de programación de alto nivel que permitía a los desarrolladores escribir aplicaciones de procesamiento de datos distribuidos sin preocuparse por los detalles de implementación de bajo nivel.

El trabajo inicial en MapReduce fue publicado por Google en un artículo técnico titulado "MapReduce: Simplified Data Processing on Large Clusters", escrito por Jeffrey Dean y Sanjay Ghemawat, y presentado en la Conferencia sobre Sistemas de Almacenamiento y Sistemas de Procesamiento de Datos en 2004. Este artículo sirvió como una descripción detallada del modelo MapReduce y sus aplicaciones prácticas en Google.

Con el tiempo, MapReduce se convirtió en un componente fundamental en el ecosistema de Big Data y fue adoptado por muchas empresas y organizaciones para procesar y analizar grandes volúmenes de datos de manera eficiente. Aunque hoy en día existen otras tecnologías de procesamiento de datos distribuidos más avanzadas, MapReduce sigue siendo relevante y continúa siendo utilizado en una variedad de aplicaciones y entornos. [12]

3.2 Funcionamiento

Su funcionamiento se basa en dos operaciones principales: la operación de "mapƨ la operación de reduce", que se aplican secuencialmente en los datos de entrada para producir resultados finales.

División de datos
El proceso comienza con la división del conjunto de datos

de entrada en fragmentos más pequeños llamados "splits". Cada split consiste en un conjunto de datos independiente que puede ser procesado de forma paralela en los nodos del clúster.

Fase de Map (mapeo)

- Cada split se envía a uno o más nodos del clúster para la fase de map.
- En esta fase, se aplica una función de map a cada registro del split para producir un conjunto intermedio de pares clave-valor.
- La función de map toma como entrada un par clave-valor y genera cero o más pares clave-valor como salida.
- Los pares clave-valor intermedios se almacenan en un área temporal de almacenamiento local en cada nodo.

Shuffle y clasificación

- Después de la fase de map, los pares clave-valor intermedios se envían a los nodos de reducción correspondientes.
- Durante este proceso de shuffle, los pares clave-valor se clasifican y agrupan por clave para que todos los valores asociados con una misma clave se envíen al mismo nodo de reducción.

Combinación y ordenamiento opcional

- Opcionalmente, se puede realizar una operación de combinación y ordenamiento antes de la fase de reduce para reducir el tráfico de red y mejorar la eficiencia.
- Durante esta operación, los pares clave-valor intermedios se combinan y se ordenan localmente en cada nodo de map antes de ser enviados a los nodos de reducción correspondientes.

Fase de Reduce (reducción)

- En esta fase, cada nodo de reducción procesa los pares clave-valor intermedios asociados con una misma clave.
- Se aplica una función de reducción a cada conjunto de valores asociados con una misma clave para producir los resultados finales.
- La función de reducción toma como entrada una clave y una lista de valores asociados con esa clave, y produce cero o más pares clave-valor como salida.
- Los resultados finales de la fase de reducción se escriben en el sistema de archivos de salida final.

Almacenamiento de resultados finales

- Los resultados finales del procesamiento se almacenan en el sistema de archivos de salida final, que puede ser utilizado por otras aplicaciones o procesos según sea necesario.

[29]

3.3 Utilidad en la actualidad

La utilidad de MapReduce en la actualidad sigue siendo significativa, aunque ha evolucionado y se ha diversificado junto con el desarrollo de otras tecnologías de procesamiento de datos distribuidos.

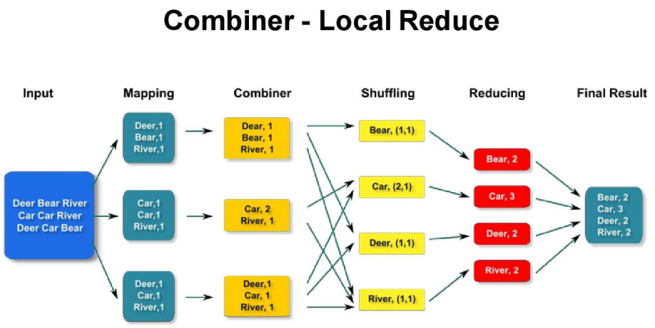

Figura 2. Arquitectura MapReduce con Combiner

MapReduce sigue siendo una herramienta eficaz para procesar grandes cantidades de datos en entornos distribuidos. Esto es especialmente útil en escenarios donde se manejan petabytes o incluso exabytes de datos, como en motores de búsqueda web, análisis de registros, redes sociales, análisis de datos científicos, entre otros.

También es capaz de manejar tanto datos estructurados como no estructurados, lo que lo hace adecuado para una amplia variedad de aplicaciones de análisis de datos. Puede utilizarse para realizar consultas complejas, agregaciones, filtrados y análisis estadísticos en datos provenientes de diferentes fuentes y en diferentes formatos, como texto, archivos de registro, datos de sensores, archivos de registro de eventos, etc.

Aunque MapReduce es inherentemente batch (lote) en su naturaleza, todavía se utiliza en algunos sistemas de procesamiento de datos en tiempo real, especialmente cuando se combina con otras tecnologías como Apache Kafka o Apache Flink. Por ejemplo, se pueden utilizar para realizar preprocesamiento o agregaciones de datos antes de enviarlos a sistemas de procesamiento en tiempo real.

MapReduce es especialmente útil para realizar análisis retrospectivos en grandes conjuntos de datos históricos. Esto puede incluir análisis de tendencias a largo plazo, identificación de patrones de comportamiento, análisis de riesgos históricos, entre otros.

Es una tecnología escalable que se puede implementar en entornos de nube para procesar grandes volúmenes de datos de manera rentable y eficiente. Los proveedores de servicios en la nube, como Amazon Web Services (AWS), Google Cloud Platform (GCP) y Microsoft Azure, ofrecen servicios gestionados de MapReduce que permiten a las organizaciones procesar grandes cantidades de datos sin preocuparse por la infraestructura subyacente.

Otro punto fuerte de MapReduce es su capacidad para lidiar con problemas. Los sistemas de procesamiento de datos a gran escala a menudo se ejecutan en infraestructuras complejas, y los fallos son una posibilidad que está presente siempre. Es por eso que MapReduce está diseñado para ser sólido y puede manejar automáticamente tareas en caso de fallos en un servidor, asegurando que el proceso continúe sin problemas y que los resultados estén disponibles cuando los necesitemos.

Utilidad en Sectores
MapReduce no está limitado a un sector en específico. Su utilidad se extiende a una amplia variedad de campos, desde finanzas hasta atención médica e investigación académica. En finanzas, se usa para analizar riesgos y detectar fraudes. En atención médica, contribuye a la investigación genómica y al análisis de datos clínicos. En la investigación académica, facilita el procesamiento de conjuntos de datos masivos en campos como la astronomía y la física de partículas. Es por eso que muchas empresas hacen uso de este algoritmo, algunas de las más importantes son:

- **Google:** Al ser su creadora es uno de los usuarios más notables cuando se trata del uso de esta tecnología. Principalmente la usa en indexaciones de páginas web y el motor de búsqueda.
- **Facebook:** Usa el algoritmo para poder analizar los grandes volúmenes de datos que se generan por parte de los usuarios.
- **Amazon:** Se usa en la plataforma de servicios web (AWS) para así poder proporcionar solución al procesamiento de datos escalables a los clientes, mejorando la eficiencia operativa y logística de las operaciones de comercio electrónico.
- **Netflix:** Usa MapReduce para procesar y analizar datos relacionados con el comportamiento de los usuarios y así poder realizar la recomendación de contenido por lo que lo hace personalizado.
- **Twitter:** Es usado para el análisis de datos en tiempo real y así poder generar el análisis de tendencias, detectar spam y comprender la interacción de los usuarios.
- **Uber:** Se usa para el análisis de datos de viajes y comportamiento del conductor con lo que se optimizan las operaciones y así mejorar la experiencia de usuario.
- **Walmart:** Esta empresa usa el algoritmo para analizar las operaciones minoristas, la gestión de inventarios, análisis de precios y optimización de la cadena de suministro.
- **NASA:** La NASA utiliza MapReduce en aplicaciones de análisis de datos relacionadas con la exploración espacial y la investigación científica. Esto incluye el análisis de datos de misiones espaciales y la simulación de sistemas complejos.

3.4 Principal Industria

Tecnología y medios de comunicación

- Análisis de datos de usuarios en redes sociales para identificar tendencias, patrones de comportamiento y preferencias.
- Procesamiento de datos de clics y visualizaciones en plataformas de publicidad en línea para mejorar la orientación de anuncios y la eficacia de las campañas publicitarias.
- Análisis de datos de audiencia en plataformas de streaming para personalizar recomendaciones de contenido y mejorar la retención de usuarios.

Finanzas

- Análisis de datos de transacciones financieras para detectar fraudes, identificar patrones de gastos y prevenir el lavado de dinero.
- Modelado y análisis de riesgos financieros utilizando datos históricos y en tiempo real para tomar decisiones de inversión y gestión de carteras.
- Análisis de datos del mercado de valores para identificar oportunidades de inversión y realizar pronósticos de mercado.

Salud

- Análisis de datos médicos y registros electrónicos de pacientes para identificar tendencias de salud, patrones de enfermedades y efectividad de tratamientos.
- Investigación biomédica utilizando grandes conjuntos de datos genómicos y proteómicos para descubrir biomarcadores y desarrollar terapias personalizadas.
- Análisis de imágenes médicas para diagnóstico asistido por computadora y detección temprana de enfermedades.

Comercio Electrónico

- Análisis de datos de ventas y comportamiento del cliente para mejorar la segmentación de mercado, la personalización de productos y la optimización de precios.
- Gestión de inventario y planificación de la cadena de suministro utilizando análisis de datos de demanda, pronósticos de ventas y datos de logística.
- Análisis de datos de comentarios y reseñas de clientes para mejorar la satisfacción del cliente y la reputación de la marca.

Telecomunicaciones

- Análisis de datos de tráfico de red para optimizar la infraestructura de red, mejorar la calidad del servicio y detectar anomalías y problemas de rendimiento.
- Análisis de datos de uso de servicios para personalizar ofertas y planes de tarifas, mejorar la retención de clientes y reducir la rotación de usuarios.
- Análisis de datos de redes sociales y comentarios de clientes para mejorar la atención al cliente y la experiencia del usuario.

4. LUSTRE

Pese a la cantidad gigantesca de software especializada en Big Data existente, hoy en día, sigue habiendo iniciativas y proyectos antiguos que siguen vigentes gracias a la comunidad que da soporte y ayuda a diversas cuestiones. Es el caso de Lustre, una combinación de dos términos: *Linux* y *Clúster*. El primero, entendido como proyecto GNU/Linux en la que distintas distribuciones comparten un kernel particular, y por el otro lado Clúster, entendido en el ámbito de los servidores: sistemas distribuidos de granjas de computadoras unidos entre sí normalmente por una red de alta velocidad y que se comportan como si fuesen un único servidor.

4.1 Historia de Lustre:

Lustre es un sistema de archivos distribuido de alta performance diseñado originalmente en 1999 por Peter J. Braam en la Universidad de Stanford. Su desarrollo inicial tuvo lugar en el Laboratorio Nacional Lawrence Livermore (LLNL) para satisfacer las necesidades de computación de alto rendimiento (HPC, cómputo de alto rendimiento). Desde entonces, Lustre ha sido adoptado ampliamente en entornos HPC en todo el mundo, convirtiéndose en una pieza fundamental para la gestión de datos en clústeres de computadoras de gran escala.

Braam fundó Cluster File Systems en 2001. En 2007, Sun Microsystems adquirió los archivos de Cluster File Systems, integrando Lustre en su hardware de alto rendimiento. Posteriormente, en 2010, Oracle Corporation, tras adquirir Sun, asumió la administración de Lustre. Sin embargo, en diciembre de 2010, Oracle anunció que detendría el desarrollo de Lustre 2.x, lo que generó incertidumbre sobre el futuro del sistema de archivos. Ante esto, surgieron nuevas organizaciones como Whamcloud, OpenSFS y EUROPEAN Open File Systems (EOFS) para continuar el desarrollo de Lustre en un modelo de desarrollo comunitario abierto.

A lo largo de los años, Lustre ha experimentado diferentes etapas de gestión y propiedad, pero en la actualidad, es ampliamente considerado como un proyecto de código abierto. El desarrollo y la evolución de Lustre se han beneficiado significativamente de la comunidad de código abierto. Organizaciones como Whamcloud, OpenSFS, EOFS y otros han contribuido al desarrollo y mantenimiento continuo del sistema de archivos Lustre. El hecho de que la marca comercial Lustre haya sido transferida a OpenSFS y EOFS en noviembre de 2019 refleja el compromiso de mantener Lustre como un proyecto de código abierto y fomentar la colaboración dentro de la comunidad.

Las últimas novedades de Lustre en sus versiones 2.X implican:

- Optimización de los componentes Lustre
- Mayor velocidad de procesamiento de datos
- Aspectos de seguridad y encriptado
- Uniformidad en los procesos de trabajo

Todos estos aspectos quedarán más claros una vez que se tome en cuenta la arquitectura cliente-servidor sobre la cual se almacena la información en Lustre

4.2 ¿Sistema de archivo distribuido?

Dado que Lustre es un sistema de archivos distribuidos, conviene antes definir dicho sistema: Un sistema de archivos distribuido (DFS, por sus siglas en inglés) es un tipo de sistema de almacenamiento de archivos que permite el acceso y la gestión de archivos en una red de computadoras distribuidas. En lugar de almacenar todos los archivos en un único sistema de almacenamiento centralizado, un DFS distribuye los archivos en varios servidores de almacenamiento interconectados.

4.3 Definición

El sistema de archivos Lustre es un sistema de archivos paralelo de código abierto que admite muchos requisitos de entornos de simulación HPC de clase líder. Nacido a partir de un proyecto de investigación en la Universidad Carnegie Mellon, el sistema de archivos Lustre ha crecido hasta convertirse en un sistema de archivos que soporta algunas de las supercomputadoras más poderosas de la Tierra. Puede escalar a miles de clientes, petabytes de almacenamiento y cientos de gigabytes por segundo de ancho de banda de E/S. Los componentes clave del sistema de archivos Lustre son los servidores de metadatos (MDS), los destinos de metadatos (MDT), los servidores de almacenamiento de objetos (OSS), los destinos de servidores de objetos (OST) y los clientes Lustre. [10]

4.4 ¿Cómo funciona?

Lustre separa las informaciones de los archivos en dos partes distintas: los datos del archivo y los metadatos. Los datos representan el contenido real de los archivos, mientras que los metadatos contienen información sobre atributos como el tamaño del archivo, permisos de acceso, y fechas relacionadas.

El sistema de archivos Lustre se compone de un conjunto de servidores de E/S conocidos como servidores de almacenamiento de objetos (OSS), y discos denominados destinos de almacenamiento de objetos (OST), donde se almacenan los datos reales. Por otro lado, los metadatos de un archivo son manejados por servidores de metadatos (MDS) y se almacenan en destinos de metadatos (MDT). En esencia, estos servidores gestionan las solicitudes de acceso tanto al contenido como a los metadatos de los archivos; las aplicaciones no acceden directamente a los discos. Los sistemas Lustre tienden a utilizar varios OSS/MDS en combinación con varios OST/MDT para garantizar capacidades de E/S de manera paralela. [6]

- Servidores de almacenamiento de objetos (OSS): manejan las solicitudes de los clientes para acceder al almacenamiento. Además, gestionan un conjunto de OST; Cada OSS puede tener más de un OST para mejorar el paralelismo de E/S .
- Destinos de almacenamiento de objetos (OST): normalmente, un OST consta de un bloque de dispositivos de almacenamiento en configuración RAID. Los datos se almacenan en uno o más objetos y cada objeto se almacena en un OST independiente .
- Servidor de metadatos (MDS): un servidor que rastrea las ubicaciones de todos los datos para poder decidir qué OSS y OST se utilizarán. Por ejemplo, una vez que se abre un archivo, el MDS ya no interviene.
- Destino de metadatos (MDT): el almacenamiento contiene información sobre los archivos y directorios, como tamaños de archivos, permisos y fechas de acceso. Para cada archivo, MDT incluye información sobre el diseño de los datos en los OST, como los números de OST y los identificadores de objetos.

Lustre ha sido diseñado para facilitar eficientes operaciones de E/S en paralelo, especialmente cuando se trata de manejar archivos de gran tamaño. No obstante, se enfrenta a desafíos cuando se trata de operaciones que involucran

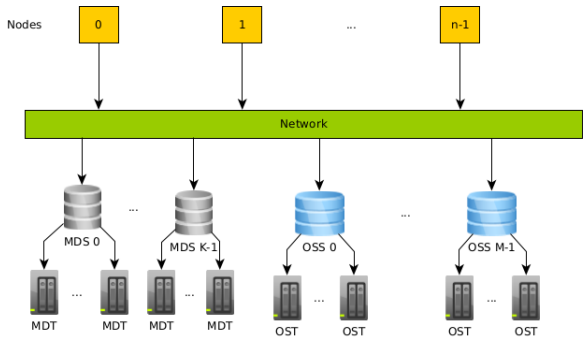

Figura 3. Arquitectura Lustre

archivos pequeños y una alta carga de operaciones de metadatos, lo que puede generar un cuello de botella en los servidores de metadatos (MDS) y en los destinos de metadatos (MDT).

En el contexto de una aplicación paralela, si varios procesos están llevando a cabo numerosas operaciones en los mismos archivos pequeños, esto puede resultar en una ralentización de las operaciones de metadatos. Incluso comandos aparentemente inocentes en Linux, como "ls -l", que muestran metadatos de archivos, pueden generar una carga de trabajo significativa en los servidores de metadatos, especialmente cuando se ejecutan en directorios con una gran cantidad de archivos, lo que resulta en múltiples solicitudes al MDS.

Para aprovechar al máximo las operaciones de entrada/salida (E/S) paralelas con Lustre, es fundamental distribuir los datos entre múltiples Object Storage Targets (OST). Este proceso de distribución de datos entre varios OST se conoce como segmentación de archivos. Conceptualmente, un archivo se considera como una secuencia lineal de bytes. En la segmentación de archivos, el archivo se divide en fragmentos de bytes que se almacenan en diferentes OST, lo que permite realizar operaciones de lectura/escritura de forma simultánea.

La técnica de crear bandas puede mejorar el ancho de banda disponible para acceder a los archivos. Sin embargo, también puede generar una sobrecarga debido al incremento en las operaciones de red y la posible congestión del servidor. Por lo tanto, la creación de bandas suele ser beneficiosa principalmente para archivos de gran tamaño.

Es importante tener en cuenta que, debido a que las supercomputadoras cuentan con muchos más nodos que los OSS/OST, el rendimiento de las operaciones de E/S puede variar considerablemente dependiendo de la carga de trabajo de E/S de toda la supercomputadora.

En un programa paralelo, el rendimiento se optimiza cuando cada proceso paralelo accede a una banda diferente de un archivo durante la E/S paralela. Además, para evitar la congestión en la red, cada proceso debe acceder a la menor cantidad posible de OST/OSS. Esto se logra mediante la alineación de bandas. El mejor rendimiento se alcanza cuando los datos se distribuyen de manera uniforme entre los OST y los procesos paralelos acceden al archivo en desplazamientos que coinciden con los límites de las bandas.

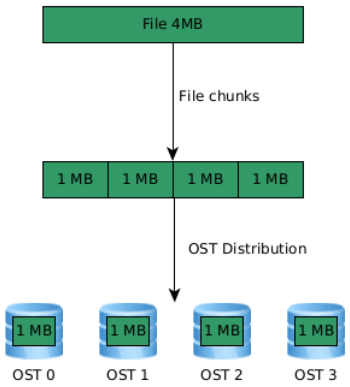

Figura 4. Arquitectura Lustre

4.5 Utilización en la actualidad

Lustre se utiliza en la actualidad en una variedad de entornos y aplicaciones donde se requiere un almacenamiento de archivos distribuido y de alto rendimiento. Algunos de los usos actuales más destacados incluyen:

- Computación de Alto Rendimiento (HPC): Lustre es fundamental en los entornos de HPC, como centros de investigación, laboratorios nacionales y supercomputadoras, donde se realizan cálculos intensivos y se manejan grandes volúmenes de datos.
- Análisis de Big Data: Aunque Hadoop Distributed File System (HDFS) es más común en el ámbito del análisis de big data, Lustre se utiliza en aplicaciones donde se requiere un almacenamiento distribuido de alto rendimiento para procesar grandes volúmenes de datos de manera eficiente.
- Industria del Entretenimiento y Medios: Lustre se utiliza en la industria del entretenimiento y los medios para el almacenamiento de archivos de medios y la edición de contenido multimedia debido a su capacidad para manejar grandes archivos y su rendimiento escalable.
- Bioinformática y Genómica: En el campo de la bioinformática y la genómica, Lustre es utilizado para almacenar y procesar grandes conjuntos de datos genómicos, donde su capacidad para manejar grandes volúmenes de datos y su rendimiento en entornos paralelos son muy útiles.
- Investigación Científica: Lustre es una opción popular en proyectos de investigación científica que requieren almacenamiento de datos distribuido y de alto rendimiento, como en la física de partículas, la astronomía, la meteorología, entre otros.

4.6 Principal industria

Las principales industrias donde Lustre encuentra su aplicación son:

- Investigación Científica: Lustre es ampliamente utilizado en centros de investigación y laboratorios nacionales donde se realizan experimentos científicos y análisis de datos complejos.
- Industria de la Energía: En la industria energética, Lustre se utiliza para el análisis de datos sísmicos,

modelado geofísico y simulaciones de reservas de petróleo y gas.
- Finanzas: En la industria financiera, Lustre se utiliza para análisis de riesgos, modelado financiero y procesamiento de datos en tiempo real para operaciones comerciales.
- Industria Aeroespacial y de Defensa: Lustre se utiliza en la industria aeroespacial y de defensa para el procesamiento de imágenes satelitales, simulaciones de vuelo y análisis de datos de radar.
- Investigación en Salud: En el campo de la investigación médica, Lustre se utiliza para el análisis de datos genómicos, modelado de proteínas y simulaciones de procesos biológicos.

En resumen, Lustre es ampliamente utilizado en una variedad de industrias donde se requiere un almacenamiento de archivos distribuido, de alto rendimiento y escalable para satisfacer las demandas de grandes conjuntos de datos y aplicaciones intensivas en computación.

4.7 Consejos

- Si es posible, evite el uso ls -l, ya que la información sobre la propiedad y los metadatos de permisos se almacenan en los MDT y los metadatos del tamaño del archivo están disponibles en los OST. Úselo lsen su lugar si no necesita información adicional.
- Evite guardar una gran cantidad de archivos en un solo directorio, mejor dividirlos en más directorios.
- Si es posible, evite acceder a una gran cantidad de archivos pequeños en Lustre.
- Si una aplicación abre un archivo para leer, ábralo solo en modo lectura.
- Una regla general es utilizar como franja la raíz cuadrada del tamaño del archivo en GB. Si el archivo tiene 90 GB, la raíz cuadrada es 9,5, así que utilice al menos 9 OST.

5. PEER TO PEER (P2P)

En la era digital, las redes Peer-to-Peer (P2P) han surgido como un fenómeno revolucionario que desafía las convenciones establecidas en la comunicación y el intercambio de recursos en línea. Desde sus modestos comienzos como plataformas de intercambio de archivos hasta su expansión hacia dominios tan diversos como la criptomoneda y la computación distribuida, el P2P ha demostrado ser mucho más que una simple tecnología: es un paradigma que transforma fundamentalmente la forma en que concebimos la conectividad en línea.[4]

Figura 5. Red Peer-to-Peer

5.1 Definiendo la Red P2P:

Básicamente una red informática P2P se refiere a una red que no tiene clientes y servidores fijos, sino una serie de nodos que se comportan a la vez como clientes y como servidores de los demás nodos de la red. Este modelo de red contrasta con el modelo cliente-servidor tradicionalmente empleado en las aplicaciones de Internet. Así, en una red P2P todos los nodos se comportan igual y pueden realizar el mismo tipo de operaciones; pudiendo no obstante diferir en configuración local, velocidad de proceso, ancho de banda y capacidad de almacenamiento.[4]

Cuando un usuario ingresa a esta red P2P, establece conexiones directas con otros nodos. Cada nodo, a su vez, desempeña un papel dual como cliente y servidor, permitiendo la recopilación y almacenamiento de información y contenido disponible para compartir. Se trata, esencialmente, de un programa diseñado para conectar a los usuarios a través de una red sin servidores, facilitando la descarga de música, películas, libros, fotos y software entre los participantes, de manera gratuita. La compartición de archivos ocurre directamente "de computador a computador"simplemente por tener acceso al sistema.[24]

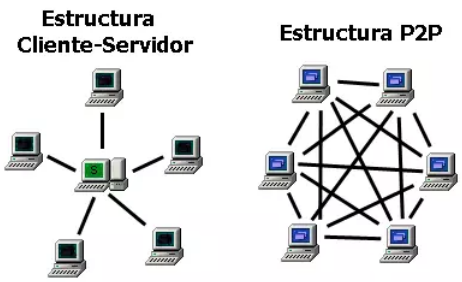

Figura 6. Estructura Cliente-Servidor vs Estructura P2P

5.2 Elementos de las redes P2P

El elemento fundamental de toda red P2P es un par o un igual, y es la unidad de procesamiento básico de cualquier red P2P. Un par es una entidad capaz de desarrollar algún trabajo útil y de comunicar los resultados de ese trabajo a otra entidad de la red, ya sea directa o indirectamente. Existen dos tipos de pares:

- Pares simples: Sirven a un único usuario final, permitiéndolo proporcionar servicios desde su dispositivo y empleando los servicios ofrecidos por otros pares de la red. Los pares suelen tener una naturaleza dinámica y heterogénea, es decir se conectan a la red de forma intermitente y tienen capacidades muy distintas.
- Superpares: Ayudan a los pares simples a que encuentre otros pares o a otros recursos de los pares. Los pares lanzan solicitudes de búsqueda de recursos a los superpares y los superpares les indican donde conseguirlos. Generalmente los superpares tienen una naturaleza estática, se encuentran conectados normalmente a la red y son fácilmente accesibles.

Otro elemento es el concepto de grupo de pares, Un grupo de pares es un conjunto de pares formado para servir a un interés común u objetivo dictado por el resto de pares implicados. Los grupos de pares pueden proporcionar

servicios a sus pares miembro que no son accesibles por otros pares de la red P2P[24]

5.3 Estructuras y Protocolos:

Los sistemas P2P son un nivel avanzado de aplicaciones de redes virtuales que incorporan superposiciones de topologías y protocolos de enrutamiento específicos. La topología de cada red define cómo los nodos están conectados entre sí, mientras que el protocolo de enrutamiento determina cómo pueden intercambiar mensajes para compartir información y recursos.

En función de su topología y del grado de uso del esquema cliente-servidor las podemos clasificar en:

- Redes P2P centralizadas: Las transacciones se realizan a través de un único servidor que actúa como punto de enlace entre los nodos. Era el caso de los primeros sistemas que se popularizaron, como Napster y compañía.
- Redes P2P híbridas o semicentralizadas: Existe un servidor central que gestiona y administra los recursos disponibles y maneja el encaminamiento de paquetes entre nodos pero sin almacenar información (por ejemplo: BitTorrent y eDonkey)
- Redes P2P descentralizadas: no requieren de la figura de un servidor central y todas las comunicaciones se gestionan directamente entre los usuarios.[17]

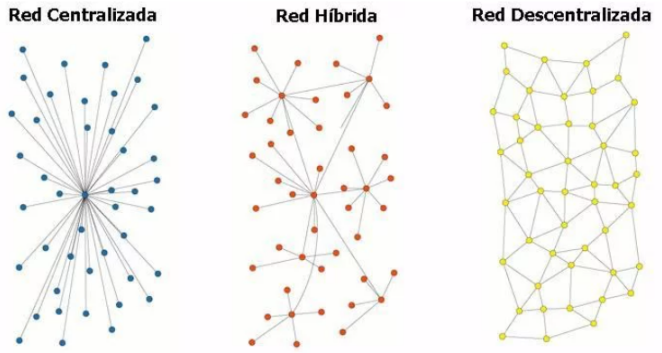

Figura 7. Clasificación de las redes P2P

5.4 Tablas Hash Distribuidas (DHT)

Las tablas hash distribuidas son un componente esencial de las redes P2P, y han tenido un efecto revolucionario en la descentralización de estas redes. Las topologías caóticas de red de la primera generación de protocolos P2P ha sido sustituida por arquitecturas más escalables y con mejores propiedades gracias a las DHT. Una DHT básicamente realiza las funciones de una tabla Hash, que básicamente se resumen en dos operaciones, almacenar el par valor y clave en la tabla, y dada una clave buscar su valor. Un ejemplo típico de Tabla Hash es un diccionario donde las palabras son claves y sus definiciones los valores.[5] Lo que se pretende con las DHT es distribuir el almacenamiento y la búsqueda de la tabla hash a múltiples máquinas. A diferencia de un modelo cliente servidor en el que se basan las arquitecturas de replicación de datos, todos los nodos son iguales que pueden unirse y dejar la red libremente. A pesar del aparente caos producido por el cambio de los miembros en la red, DHT garantiza su funcionamiento.

5.5 Evolución del sistema P2P

El concepto de Peer-to-Peer (P2P) tiene sus raíces en el desarrollo de redes de computadoras y la evolución de la comunicación digital. [17]

1. Década de 1960 - ARPANET: Los primeros vestigios del P2P pueden rastrearse hasta el desarrollo de ARPANET, la red precursora de Internet. Financiado por la Agencia de Proyectos de Investigación Avanzada de Defensa (ARPA) de los Estados Unidos, sentó las bases para la comunicación entre computadoras distribuidas.
2. Década de 1970 - Redes Descentralizadas: Se produjo un crecimiento en la investigación y desarrollo de redes descentralizadas que permitieran la comunicación entre nodos de manera más eficiente.
3. Década de 1980 - Desarrollo de Protocolos: El concepto de igualdad entre nodos y la idea de compartir recursos de manera directa comenzaron a tomar forma. El desarrollo de protocolos que permitieran la comunicación punto a punto allanó el camino para la evolución hacia el P2P.
4. Década de 1996 -Aparece *Hotline Connect*, desarrollado por Adam Hinkley. El primer programa con el sistema de comunicación P2P fue creado para su uso en universidades. Pero no tardó en verse su utilidad entre ordenadores particulares para el intercambio de música en formato mp3 y software de todo tipo.
5. Año 1999: El 1 de junio se pone en marcha Napster, creado por Shawn Fanning y Sean Parker.
6. Años 2000: Justin Frankel y Tom Peeper desarrollan Gnutella, segunda generación de P2P, la primera red descentralizada, en las que ningún servidor hace de puente, de manera que cerrando aquel pueda acabarse con el sistema. Gnutella dará lugar al servidor Limewire.
7. Año 2001: Napster alcanza los 13 millones y medio de usuarios, un juez decide cerrarlo porque considera ilegal el uso que se hace de la música. Sus clientes se pasan a las numerosas redes descentralizadas que se desarrollan a partir de este momento. Nace BitTorrent, protocolo de comunicaciones para el intercambio de archivos, creado por el programador Bram Cohen.
8. Año 2002: Nace eMule, que pertenece a la segunda generación de servidores de la red eDonkey.
9. Año 2007: Los programas más populares de descarga de archivos son Ares, Limewire, BitTorrent y eMule, aunque aparecen nuevos sistemas de pago más eficientes y seguros que permiten mantenerse dentro de la legalidad, como Rapidshire, Sendspace, Megaupload o Filefactory.

6. NAPSTER

Napster, fundado por Shawn Fanning y Sean Parker en 1999, representó una innovación revolucionaria en la distribución de archivos de música en línea. Este servicio pionero, basado en la tecnología peer-to-peer (P2P), transformó la manera en que las personas compartían y accedían a la música digital. Napster se destacó al utilizar servidores centralizados como vastos almacenes de archivos MP3,

permitiendo a los usuarios acceder y descargar música de manera rápida y sencilla.[27]

La esencia de Napster radicaba en su enfoque colaborativo: para descargar música, los usuarios debían compartir sus propios archivos MP3 con la comunidad. Este modelo democratizó el acceso a la música digital, desplazando a métodos anteriores como las aplicaciones FTP, IRC o Usenet, que resultaban engorrosas y lentas en comparación con la eficiencia del P2P.[21]

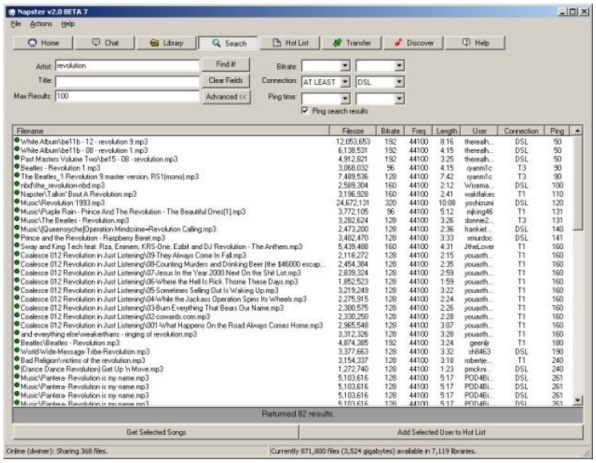

Figura 8. Interfaz de Napster

La historia de Napster se convierte así en un hito fundamental para comprender la evolución de la distribución digital de música, al mismo tiempo que desencadenó considerables desafíos legales asociados con la piratería en línea. La plataforma dejó un legado perdurable al cambiar la forma en que las personas interactúan con la música en el entorno digital, influenciando la transformación de la industria musical y el surgimiento de nuevas plataformas de distribución legales.[21]

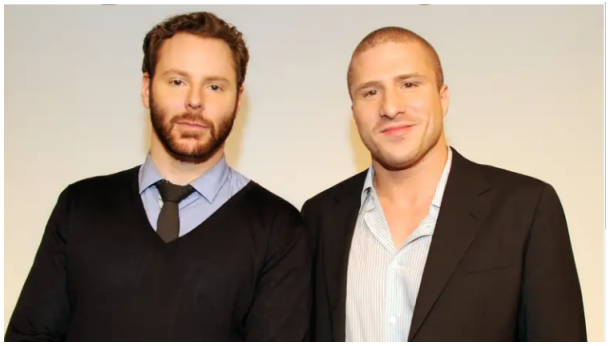

Figura 9. Fundadores De Napster: Sean Parker, Shawn Fanning

Nueve meses después de su estreno, Napster contaba con 10 millones de usuarios. Pasados 18 meses, la abismal cifra de 80 millones de usuarios registrados y la descarga de 10.000 canciones por segundo58 aterrorizaban a la industria del entretenimiento. En febrero de 2001, Napster se encontraba en la cima más alta de su popularidad.

Cuando las empresas discográficas se percataron del peligro que se les aproximaba, ya era demasiado tarde. Napster había logrado fundar una cultura de libre distribución de música entre millones de personas. Naturalmente, los efectos fueron desastrosos para los artistas, productores y las casas disqueras. Si tan solo el uno por ciento de sus usuarios hubiese pagado por las descargas realizadas a la fecha, las ganancias hubiesen sido de medio millón de dólares por cada canción. Sin embargo, en febrero de 2001 Fanning fue llevado a juicio y perdió. Así, en AM Records Inc. vs. Napster Inc., los demandantes alegaron que el demandado no sólo era responsable de contribuir directamente a la infracción del *copyright contributory infringement*, sino, además, por el hecho de terceros *vicarious liability*, es decir, sus usuarios. Con base en estas teorías, los peticionarios lograron que una Corte de Distrito Federal en Estados Unidos le imputara a Napster una condena preliminar a pagar perjuicios.[21]

Figura 10. Logo de Napster

Sin embargo, en julio de 2001, Napster fue cerrado después de que un tribunal de apelaciones confirmara la orden de cierre. Aunque el servicio original cerró, su impacto en la industria de la música y la forma en que la gente consumía música en línea fue significativo. A pesar de su cierre, el modelo P2P introducido por Napster influyó en el desarrollo de futuras plataformas de intercambio de archivos y servicios de música en línea legal. Las discográficas y artistas también se vieron obligados a adaptarse a la era digital y buscar nuevas formas de distribución y monetización de la música.[7]

Después de varios cambios de propiedad, Napster fue relanzado como una plataforma de música en streaming legal. Sin embargo, ya no tenía la influencia y la notoriedad que tuvo en sus días de intercambio de archivos P2P.

Arquitectura Napster La arquitectura de Napster se basaba en un modelo peer-to-peer (P2P) que permitía a los usuarios compartir archivos de música directamente entre ellos sin la necesidad de un servidor central para almacenar todos los archivos.

- Cliente Napster: Los usuarios debían descargar e instalar el software cliente de Napster en sus computadoras. Este software proporcionaba una interfaz fácil de usar para buscar, compartir y descargar archivos de música en formato MP3.
- Índice Centralizado: Aunque Napster es a menudo asociado con la descentralización, la característica central de su arquitectura era un índice centralizado. El índice central contenía información sobre qué archivos estaban disponibles y qué usuarios los tenían. Cuando un usuario buscaba una canción, el índice central facilitaba la búsqueda, mostrando una lista de usuarios que poseían la canción deseada.

- Conexiones P2P:Una vez que un usuario identificaba otro usuario que tenía la canción que buscaba, se establecía una conexión P2P directa entre ellos. La descarga o intercambio de archivos ocurría directamente entre los clientes, sin pasar por un servidor central para la transferencia de datos.
- Intercambio de Archivos: La característica fundamental de Napster era la capacidad de compartir archivos MP3. Los usuarios compartían sus propias bibliotecas de música al permitir que otros descargaran canciones de su colección.
- Proceso de Autenticación: Napster requería que los usuarios se registraran y autenticaran en el sistema antes de poder utilizar el servicio. Esta autenticación permitía mantener un registro de las actividades de los usuarios y garantizar ciertas normas en la comunidad.[7]

Figura 11. Descarga de música

7. MIDDLEWARE P2P

Middleware es el conjunto de servicios informáticos distribuidos que, dentro de cualquier entorno de procesamiento determinado, permite a los clientes y al servidor comunicarse e interoperar entre sí de la manera más conveniente, flexible y correcta posible. La función del middleware es facilitar la tarea de diseñar, programar y administrar aplicaciones distribuidas proporcionando un entorno de programación distribuida simple, consistente e integrado. La interoperabilidad es la clave para un sistema de middleware.[22]

Específicamente, definimos el middleware P2P como una capa de software distribuido que abstrae la complejidad y heterogeneidad de los sistemas P2P subyacentes con su multitud de servicios, protocolos y aplicaciones. Actualmente las aplicaciones P2P más importantes y populares son las de intercambio de datos. La capa de middleware implementa una abstracción fundamental por encima de los sistemas P2P subyacentes y proporciona servicios comunes de intercambio de datos para aplicaciones de alto nivel.[15]

7.1 Características

- Abstracción de la Red: El middleware P2P abstrae la complejidad de la comunicación entre nodos en una red P2P. Proporciona interfaces y funciones que permiten a los desarrolladores interactuar con la red sin preocuparse por los detalles de la implementación subyacente.
- Gestión de Recursos: Facilita la gestión de recursos distribuidos en una red P2P. Esto puede incluir la administración de nodos, la asignación de tareas y la coordinación de actividades entre los participantes de la red.
- Descubrimiento de Nodos: Proporciona mecanismos para descubrir y conectarse con otros nodos en la red P2P. Esto implica la implementación de servicios de búsqueda y registro que facilitan la ubicación eficiente de nodos en la red.
- Seguridad y Autenticación: Ofrece funciones relacionadas con la seguridad y la autenticación. Esto puede incluir la implementación de mecanismos para proteger la integridad de los datos, autenticar nodos y garantizar la confidencialidad de la información transmitida.
- Gestión de Transacciones: Facilita la gestión de transacciones distribuidas en una red P2P. Esto puede abordar la coherencia de datos, la atomicidad de las operaciones y la tolerancia a fallos.
- Escalabilidad: Ayuda a garantizar la escalabilidad de las aplicaciones P2P. Esto implica la capacidad de gestionar eficientemente un número creciente de nodos en la red sin comprometer el rendimiento.
- Integración con Aplicaciones: Permite la integración sencilla de las capacidades P2P en aplicaciones más amplias. Los desarrolladores pueden utilizar el middleware P2P para construir aplicaciones distribuidas sin tener que lidiar directamente con la complejidad de la red P2P.

El middleware P2P juega un papel crucial en la simplificación del desarrollo de aplicaciones distribuidas, permitiendo a los desarrolladores centrarse en la lógica de la aplicación sin tener que preocuparse por los detalles específicos de la red P2P subyacente.[19]

8. BITTORRENT

Es un protocolo de comunicación utilizado para compartir archivos (particularmente archivos grandes) a través de Internet de forma descentralizada. Esto se logra utilizando la capacidad de carga de los pares que están descargando un archivo. Un aumento considerable en los descargadores solo resultará en un aumento modesto en la carga en el servidor que aloja el archivo [14, 20].

8.1 Historia

BitTorrent fue creado en 2001 por Bram Cohen, un programador estadounidense frustrado por los largos tiempos de espera que experimentaba con aplicaciones que usaban otros protocolos como FTP. En 2002 lo presentó en una conferencia. Su objetivo era proporcionar a las personas una forma rápida y sencilla de distribuir e intercambiar software de Linux en línea. Sin embargo, pronto se vio el potencial de BitTorrent para la distribución de películas y programas de televisión, lo que llevó a un crecimiento explosivo [20].

8.2 Arquitectura

La arquitectura de BitTorrent normalmente consta de los siguientes elementos:

- Un archivo de metainformación estática (*archivo torrent*).
- Un rastreador (*tracker*).
- Un descargador original (*seed*).
- El descargador final (*leecher*).

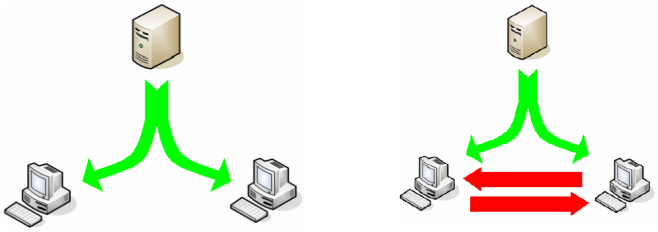

Figura 12. Flujo básico de información en protocolo BitTorrent

Para publicar un archivo, se crea un *archivo torrent* que contiene información sobre el archivo, como su nombre, tamaño, hash y la URL del *tracker*. Este archivo se distribuye para que los usuarios puedan descargarlo. El torrent se crea utilizando un programa, comúnmente incluido en los clientes de BitTorrent. Para descargar (*leech*) o para sembrar (*seed* un archivo, se necesita un cliente de BitTorrent, que es una aplicación que administra el proceso de descarga.

La descarga de BitTorrent se inicia abriendo el archivo torrent en el cliente de BitTorrent, o bien, a través de un enlace magnético (magnet links). El rastreador mantiene un registro de pares que están descargando un archivo y les ayuda a encontrarse. Este grupo de pares que comparten el mismo torrent representa un enjambre (*swarm*). El archivo original se divide en piezas más pequeñas, y los datos descargados se verifican utilizando códigos hash para garantizar su autenticidad.

A medida que el usuario descarga piezas del archivo, también se convierte en un par que puede compartir esas piezas con otros usuarios que deseen descargar el mismo archivo. Este proceso de compartir y descargar continuará hasta que todos los usuarios hayan descargado el archivo completo. Una vez que un usuario ha descargado todas las partes del archivo, puede optar por convertirse en una semilla y seguir compartiendo el archivo completo con otros usuarios en la red. Algunos sitios web de rastreadores fomentan la siembra al penalizar a los pares que no siembran sus archivos después de que se completan las descargas [20].

9. BOINC (BERKELEY OPEN INFRASTRUCTURE FOR NETWORK COMPUTING)

BOINC es un sistema de software para cómputo voluntariado: les permite a las personas donar tiempo en sus computadoras domésticas (Windows, Mac, Linux) y teléfonos inteligentes (Android) a proyectos de investigación científica. Se ejecuta con la mínima prioridad de procesamiento y limita la cantidad de memoria utilizada para evitar una paginación excesiva. El desarrollo de BOINC tiene sus orígenes en 2002, por un grupo basado en la Universidad de California, Berkeley [1].

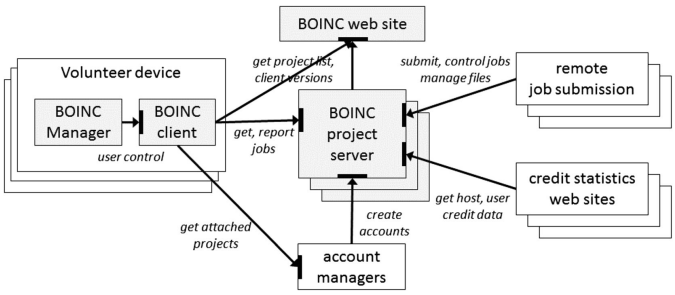

Figura 13. Componentes e interfaces RPC del sistema BOINC

9.1 Estructura BOINC

En terminología BOINC, un proyecto es una entidad que usa BOINC para realizar cómputo. Cada proyecto tiene un servidor, basado en el software servidor BOINC, que distribuye tareas a dispositivos voluntarios. Los proyectos son autónomos e independientes, y pueden operar en sitios web para su propia gestión, así como entablar comunicación con los voluntarios y visualización de resultados.

Un voluntario es un dueño de un dispositivo de cómputo que desee participar en cómputo voluntariado. Esto lo realizan por medio de a) instalar el cliente BOINC en su dispositivo; b) seleccionando proyectos para apoyar y crear cuentas para cada proyecto; y c) adjuntando el cliente BOINC a cada cuenta. A cada vínculo se le puede asignar una cantidad de recursos, indicando la cantidad de cómputo relativo a realizar, a largo plazo, para el proyecto [2].

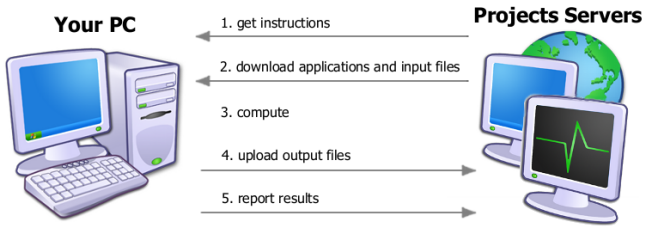

Figura 14. Funcionamiento de BOINC desde el punto de vista de un voluntario

De manera más simple y desde la perspectiva del voluntario, la participación en un proyecto BOINC funciona de la siguiente manera:

1. El dispositivo voluntario recibe un conjunto de tareas del servidor de planificación del proyecto dependiendo de los recursos del dispositivo. Los proyectos admiten varias aplicaciones, y el servidor puede enviar tareas de cualquiera de ellas.
2. El dispositivo voluntario descarga archivos ejecutables y de entrada desde el servidor de datos del proyecto. Nuevas versiones de las aplicaciones se descargan de manera automática.
3. El dispositivo voluntario ejecuta las aplicaciones, produciendo archivos de salida y los carga al servidor de datos
4. Posteriormente el dispositivo voluntario reporta al servidor de planificación sobre las tareas terminadas y recibe trabajos nuevos.

Este ciclo se repite de manera indefinida y automáticamente por BOINC. El voluntario no necesita hacer nada [18].

9.2 UNAM@Home

Figura 15. M.C. Alejandro Velázquez Mena

Fue un proyecto liderado por el M. C. Alejandro Velázquez Mena, actual jefe de la División de Ingeniería Eléctrica en la Facultad de Ingeniería de la UNAM. Haciendo uso de BOINC se implementó la plataforma UNAM@Home. El proyecto se llevó a cabo en un periodo de 2015 a 2017. En este tiempo se participó en 23 proyectos aportando procesamiento de cómputo, siendo el más destacable el descubrimiento de un número primo de más de 1,000,000 de dígitos en octubre de 2016. También se creó el primer proyecto de la Facultad de Ingeniería, encontrar más dígitos del número de Euler y se apoyó al proyecto *Serpent*, que buscaba modelar partículas nucleares y analizarlas [26, 23].

MENA_COMP_DIE_FI_UNAM

	Credits:	BSrac:	Rank:	Rank%:
Collatz	6,730,808,915	0	79	95.666
PrimeGrid	960,078,678	0	115	96.518
GPUGRID	505,505,825	0	156	91.545
Moo!	332,165,530	0	83	90.247
Asteroids	182,094,720	0	53	97.480
WCG	174,164,874	517	256	99.091
BitcoinUt	153,380,350	0	138	59.884
Einstein	153,146,829	0	435	96.394
MilkyWay	151,688,262	0	225	95.298
SETI	91,863,367	0	352	99.457
yoyo	24,922,962	0	57	95.605
SAT@home	22,427,365	0	11	97.566
Rosetta	21,311,942	0	357	97.185
POGS TSN	20,197,621	0	60	93.428
SIMAP	15,982,422	0	46	98.166
WUProp	13,406,652	0	25	96.580
Malaria	9,464,463	0	55	97.781
NmbrField	9,090,923	0	87	89.403
DENIS@Hom	6,522,065	0	47	90.656
ATLAS@Hom	3,821,182	0	18	96.498
LHC	3,821,182	0	238	95.748
LHC-dev	3,821,182	0	239	-251.471
EDGeS	1,024,252	0	53	92.251
CPDN	814,117	0	1,354	83.230
Acoustics	93,198	0	126	59.486
	9,587,797,708	**63,159**	**140**	**99.875**

BOINC STATS — Team Stats boincstats.com

Figura 16. Distribución de créditos BOINC a proyectos de parte de UNAM@Home

Tras su finalización formal en 2017, la plataforma en su modo cliente sigue aportando a diversos proyectos bajo el nombre de equipo MENA_COMP_DIE_FI_UNAM, amasando un total 9,587,797,412.53 créditos BOINC, forma en que los contribuyentes pueden mantener un registro del tiempo de CPU donado a los proyectos. Esto lo coloca en segundo lugar a nivel país en cuanto a contribuciones a los proyectos de la infraestructura, representando al 48.93 % de las aportaciones provenientes de México [30, 31].

10. FOLDING@HOME

Figura 17. Dr. Vijay Pande y Dr. Gregory Bowman con su equipo

Es un proyecto de cómputo distribuido que comenzó en octubre de 2000 en la Universidad de Stanford, desde el 2019 hasta la actualidad está siendo administrado por el Dr. Gregory Bowman en la Universidad de Pennsylvania. El proyecto se centra en entender el plegamiento de proteínas, las enfermedades resultantes del mal plegamiento de éstas y agregaciones, así como el desarrollo de nuevas formas de tratamiento mediante la computación. Para esto hace uso de la computación distribuida, permitiendo a los voluntarios conducir experimentos diseñados y asignados por el equipo de Bowman desde Pennsylvania, al mismo tiempo, los usuarios pueden observar y monitorear los resultados de las simulaciones, o simplemente continuar con sus vidas y dejar a sus dispositivos realizar el cómputo en el trasfondo. Fundamentalmente se realizan muestreos adaptivos y Modelos de Markov (Markov State Models) [25, 13, 8].

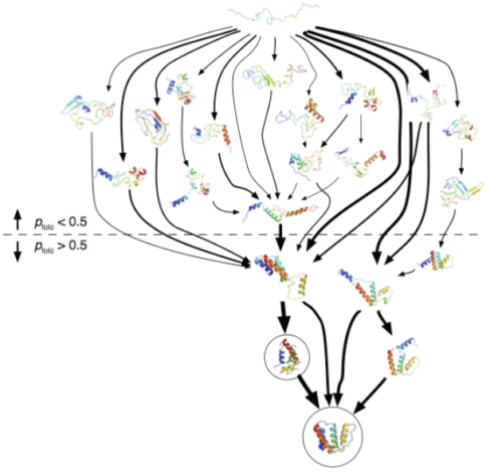

Figura 18. MSM para la proteína ACBP

Cabe destacar que el proyecto Folding@Home no hace uso de la infraestructura BOINC. En enero de 2006 se lanzó una iniciativa en el cliente BOINC y se realizaron pruebas Alpha con un grupo pequeño, pero se encontraron con problemas significativos. Otros problemas posteriores provocaron que en junio de 2006 el equipo de Folding@Home decidiera detener la implementación del proyecto en BOINC y continuase con su propia plataforma [11].

Entre las enfermedades que el equipo de Bowman investiga se encuentran: [9]

- Cáncer: Cáncer de mama, p53, epigenética, cáncer de riñón.
- Enfermedades infecciosas: Dengue, enfermedad de Chagas, virus del Zika, Hepatitis C, virus del Ébola, COVID-19.
- Enfermedades neurológicas: Enfermedad de Alzheimer, enfermedad de Huntington, enfermedad de Parkinson.

11. RELACIÓN CON LOS SISTEMAS DISTRIBUIDOS

En el ecosistema de sistemas distribuidos, Hadoop con su paradigma de MapReduce emerge como una solución prominente para el procesamiento de datos a gran escala. Hadoop abstrae la complejidad de los sistemas distribuidos al proporcionar una infraestructura que gestiona la distribución y replicación de datos de manera transparente para los usuarios finales.

MapReduce, el modelo de programación utilizado en Hadoop, divide las tareas en unidades más pequeñas que se distribuyen y ejecutan en paralelo en un clúster de nodos, permitiendo así un procesamiento rápido y eficiente de grandes conjuntos de datos. Esta abstracción facilita a los desarrolladores centrarse en el desarrollo de lógica de aplicación sin preocuparse por la complejidad subyacente de la infraestructura distribuida.

Hadoop y MapReduce se basan en varios principios y conceptos de los Sistemas Distribuidos para funcionar:

- **Almacenamiento distribuido:** Hadoop y MapReduce utilizan el Hadoop Distributed File System (HDFS) para almacenar grandes conjuntos de datos en varios nodos de un clúster. HDFS implementa técnicas de replicación y fragmentación para distribuir los datos y garantizar la disponibilidad y la escalabilidad.
- **Procesamiento Distribuido:** Hadoop utiliza MapReduce, para ejecutar tareas en paralelo en varios nodos de un clúster. MapReduce divide las tareas en unidades más pequeñas que se ejecutan en paralelo, lo que permite procesar grandes conjuntos de datos de manera eficiente.
- **Escalabilidad:** Hadoop y MapReduce están diseñados para escalar horizontalmente, lo que significa que se puede agregar más nodos al clúster para aumentar la capacidad de almacenamiento y procesamiento. Esta característica es fundamental para manejar el crecimiento continuo de los datos.
- **Tolerancia a Fallos:** Hadoop y MapReduce están diseñados para ser tolerante a fallos, lo que significa que puede continuar funcionando incluso si algunos nodos del clúster fallan. Esta característica es importante para garantizar la alta disponibilidad del sistema.

En el ámbito de los sistemas distribuidos de almacenamiento de archivos, Lustre destaca como una solución robusta y escalable diseñada para entornos que requieren un rendimiento excepcional y un manejo eficiente de grandes conjuntos de datos. Lustre se ha establecido como una opción popular en campos como la computación de alto rendimiento (HPC), la investigación científica y la industria del entretenimiento, donde el acceso rápido a grandes volúmenes de datos es esencial para operaciones críticas. Con su capacidad para distribuir archivos entre múltiples nodos de almacenamiento (OST) y administrar metadatos a través de servidores de metadatos (MDS), Lustre permite un acceso paralelo y simultáneo a los datos, lo que mejora significativamente el rendimiento y la escalabilidad del sistema.

Además, Lustre ofrece características avanzadas de administración, incluida la capacidad de configurar las franjas (stripes) para optimizar la distribución de datos y el rendimiento del sistema. Sin embargo, su implementación y mantenimiento pueden ser complejos y requieren una comprensión profunda de la infraestructura subyacente. Los administradores del sistema deben gestionar cuidadosamente aspectos como la configuración de servidores, la monitorización del rendimiento y la resolución de problemas para garantizar un funcionamiento eficiente y fiable del sistema.

Por otro lado, P2P representa un paradigma para la construcción de sistemas distribuidos y aplicaciones en los cuales datos y recursos computacionales son provistos por varios huéspedes en Internet, todos los cuales participan en la provisión de un servicio uniforme. Un problema clave es la UBICACIÓN de los objetos de datos en muchos huéspedes y el subsecuente ACCESO a los mismos de manera de balancear la carga y asegurar la disponibilidad sin agregar sobrecarga. Una primera aplicación que se presenta en este documento es Napster,lanzado en 1999, fue un pionero en la aplicación de conceptos de sistemas distribuidos en el ámbito de la distribución de música en línea.La arquitectura de Napster se basaba en un modelo P2P que permitía a los usuarios compartir archivos directamente entre sí, eliminando la necesidad de un servidor centralizado para almacenar toda la música. Cada usuario que participaba en la red actuaba como un nodo distribuido, compartiendo y accediendo a archivos de música. En Napster, cada usuario era un nodo autónomo en la red. Cada nodo tenía su propia colección de archivos MP3 y decidía qué compartir con los demás usuarios. Este enfoque descentralizado permitía la autonomía individual de cada participante.

Otro caso, es el middleware P2P que proporciona una capa de abstracción que oculta la complejidad de la comunicación y la interacción entre nodos en una red P2P. Esto facilita a los desarrolladores la creación de aplicaciones distribuidas sin preocuparse por los detalles específicos de la red subyacente.

Otra aplicación del P2P se encuentra en el protocolo BitTorrent, el cual elimina la dependencia a un único servidor centralizado para la distribución de archivos, BitTorrent

aprovecha la capacidad de carga de múltiples pares que están descargando o compartiendo el mismo archivo. Esto permite una distribución más eficiente y descentralizada de los archivos, ya que cada par contribuye con parte de su ancho de banda y capacidad de almacenamiento para ayudar en la distribución de los archivos.

BOINC es un sistema dedicado a la computación voluntaria, una subcategoría del cómputo distribuido, pues coordina la participación de voluntarios de todo el mundo para contribuir con su poder de cómputo a proyectos de investigación científica. Además, BOINC permite la distribución de tareas y datos entre los dispositivos de los voluntarios, lo que lo convierte en un ejemplo de un sistema distribuido de cómputo a gran escala. De manera similar, Folding@Home utiliza la potencia de cómputo de los voluntarios para realizar simulaciones de plegamiento de proteínas y entender mejor su funcionamiento. Dentro de la Facultad de Ingeniería en la Universidad Nacional Autónoma de México, el proyecto UNAM@Home además de realizar sus propias aportaciones con recursos de cómputo a proyectos en BOINC, sirvió para introducir y generar contenido didáctico para la asignatura propuesta de Sistemas Distribuidos, que se planeó introducir para el plan de estudios 2016-1 [23].

12. CONCLUSIÓN

Consideramos a los tópicos propuestos un conjunto de herramientas para expandir nuestro conocimiento de los sistemas distribuidos. Por ejemplo, logramos dilucidar algunos conceptos asociados a los sistemas distribuido generales pero aplicados a los tópicos vistos en este documento. La importancia de la interconexión de nodos nos permitió comprender el intercambio de información entre computadoras que trabajan en un mismo clúster de dispositivos; la escalabilidad, entre la que destaca la horizontal, pues añadiendo más dispositivos, los sistemas son flexibles y no provocan colisiones o fallos; la tolerancia a fallos, debido a que hay redundancia en los datos y en Hadoop, por ejemplo, hay mucha repetición de la información con ayuda de los sistemas de archivos distribuidos. Por último, y la opción que consideramos más importante, la transparencia, ya que el usuario, al estar interactuando con el software de su preferencia, no se da cuenta de los posibles fallos que hay detrás: hay un ocultamiento en dónde se obtiene la respuesta requerida y eso es lo único que importa. Es decir, todo es visto como una sola entidad, por más que detras allá una infinidad de dispositivos interconectados. Afirmamos, como propone Tanenbaum, que un sistema distribuido es una colección de computadoras independientes que aparecen ante los usuarios del sistema como una única computadora. Su soporte, para dicho fin está sustentado en dos importantes innovaciones tecnológicas: el microprocesador y las redes de datos. [3]

13. ENLACE AL VIDEO

El video correspondiente se aloja en el siguiente vínculo de YouTube:
https://www.youtube.com/watch?v=ZI9cafFbJvE

REFERENCIAS

[1] D. P. Anderson. *BOINC in Retrospect*. https://continuum-hypothesis.com/boinc_history.php. Ene. de 2022.

[2] D. P. Anderson. "BOINC: A Platform for Volunteer Computing". En: *Journal Of Grid Computing* 18.1 (nov. de 2019), págs. 99-122. DOI: 10.1007/s10723-019-09497-9.

[3] Francisco Asís. *Sistemas distribuidos*. 2014.

[4] Oscar Blancarte. *Arquitectura Peer To Peer (P2P)*. 2020. URL: https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/p2p.

[5] John F. Buford, Heather Yu y Eng Keong Lua. "Chapter 2 - Peer-to-Peer Concepts". En: *P2P Networking and Applications*. Ed. por John F. Buford, Heather Yu y Eng Keong Lua. Boston: Morgan Kaufmann, 2009, págs. 25-44. ISBN: 978-0-12-374214-8. DOI: https://doi.org/10.1016/B978-0-12-374214-8.00002-7. URL: https://www.sciencedirect.com/science/article/pii/B9780123742148000027.

[6] CSC. *Sistema de archivos Lustre*. 2024. URL: https://docs.csc.fi/computing/lustre/.

[7] Ballesteros Diego. *Napster: La Revolución Musical y su Legado en la Era Digital*. 2023. URL: https://medium.com/@diegoballesteros.dev/napster-la-revoluci%C3%B3n-musical-y-su-legado-en-la-era-digital-d58be1383d82.

[8] *Dig deeper – Folding@home*. https://foldingathome.org/dig-deeper/.

[9] *Diseases – Folding@home*. https://foldingathome.org/diseases/.

[10] EOFS. *Acerca del sistema de archivos Lustre*. 2024. URL: https://www.lustre.org/about/.

[11] *FAH on BOINC – Folding@home*. https://foldingathome.org/faqs/high-performance/fah-on-boinc/.

[12] Apache Software Foundation. *Apache Hadoop*. ://HADOOP.APACHE.ORG/.

[13] A. Gardner. *Folding@home: How You, and Your Computer, Can Play Scientist*. https://www.pennmedicine.org/news/news-blog/2023/may/folding-at-home. Mayo de 2023.

[14] E. Gregersen. *BitTorrent — File sharing, Peer-to-Peer Networking*. https://www.britannica.com/technology/BitTorrent. Feb. de 2024.

[15] "Middleware P2P para la Sincronización de Eventos Discretos en una Simulación Distribuida de Sistemas que evolucionan en el tiempo". En: 2012.

[16] Helena. *¿Qué es Hadoop y para qué sirven en Big Data*. ://AYUDALEYPROTECCIONDATOSES/BIGDATA/HADOOP/.

[17] CurioSfera Historia. *Historia del P2P: origen e inventor*. 2023. URL: https://curiosfera-historia.com/historia-del-p2p-inventor/#google_vignette.

[18] *How BOINC works - BOINC*. https://boinc.berkeley.edu/wiki/How_BOINC_works.

[19] Taylor Ian J. *From P2P to Web Services and Grids Peers in a Client/Server World*. Springer, 2005.

[20] J. Arne Johnsen, Erik Karlsen y S. Sæther Birkeland. *Peer-to-peer networking with BitTorrent*.

https://web.cs.ucla.edu/classes/cs217/05BitTorrent.pdf. Dic. de 2005.

[21] García Jose. *Napster: inicio, auge y caída del servicio que puso en jaque a la industria musical*. 2019. URL: https://www.xataka.com/historia-tecnologica/napster-inicio-auge-caida-servicio-que-puso-jaque-a-industria-musical.

[22] Minglu Li. *Grid and Cooperative Computing*. Springer, 2003.

[23] A. Velázquez Mena. *Incorporación de las arquitecturas de computo distribuido en la asignatura de sistemas distribuidos en la Facultad de Ingeniería*. https://www.innovacioneducativa.unam.mx:8443/jspui/handle/123456789/5364. Mar. de 2020.

[24] Ramón Jesús Millán Tejedor. *Domine las redes P2P : "Peer to Peer.ºrígenes, funcionamiento y legislación del P2P, selección y configuración del acceso de banda ancha a internet"*. Creaciones Copyright, 2006.

[25] *Project Timeline – Folding@home*. https://foldingathome.org/project-timeline/. 2000.

[26] R. Ovando Trejo. *Hallazgo numérico en la FI*. https://www.comunicacionfi.unam.mx/mostrar_nota.php?id_noticia=876. Ene. de 2017.

[27] Eduardo Secondo Varela Pezzano. *Tecnología de redes P2P*. Vlex, 2015.

[28] T. White. *Hadoop: The Definitive Guide (4th ed.)*. O'Reilly Media, 2015.

[29] J.A.R.N. Zezinho. *Hadoop - MapReduce Applications. Big Data for executives and professionals*. HTTPS://MEDIUM.COM/XNEWDATA/HADOOP-MAPREDUCE-APPLICATIONS-29CC6AA813D4. 2023.

[30] W. De Zutter. *BOINC combined - team stats - MENA_COMP_DIE_FI_UNAM — BOINCstats/BAM!* https://www.boincstats.com/stats/-1/team/detail/138744386/projectList.

[31] W. De Zutter. *BOINC combined - team stats - MENA_COMP_DIE_FI_UNAM — BOINCstats/BAM!* https://www.boincstats.com/stats/-1/team/detail/138744386/overview.