

Escalabilidad y Paralelización mediante el uso de Hadoop Distributed File System

Damián P. Barry¹, Carlos E. Buckle¹, Rodrigo Jaramillo¹,
Ignacio Real¹, Fernando G. Tinetti²

¹Depto. de Informática, Fac.de Ingeniería, Universidad Nacional de la Patagonia San Juan Bosco.
Puerto Madryn, Argentina
+54 280-4472885 – Int. 117.

damian_barry@unpata.edu.ar, cbuckle@unpata.edu.ar, rodrigojaramillo@gmail.com, nachoreal@hotmail.com

²III-LIDI, Facultad de Informática - Universidad Nacional de La Plata
²Investigador Comisión de Investigaciones Científicas de la Prov. de Bs. As.
La Plata, Argentina

Resumen

En el marco del proyecto de investigación TECNICAS DE RECUPERACION DE INFORMACION EN GRANDES VOLUMENES DE DATOS HETEROGENEOS CON BASES DE DATOS NOSQL, el presente trabajo se orienta a evaluar configuraciones de clusters utilizando Hadoop Distributed File System (HDFS) para comprobar las capacidades de disponibilidad, escalabilidad y paralelización en la recuperación de información.

Dicha evaluación permitirá establecer las capacidades necesarias con las que debería contar un File System Distribuido, tanto desde la perspectiva de almacenamiento y técnicas de indexación, como de distribución de las consultas, paralelización, escalabilidad y rendimiento en ambientes heterogéneos.

Para ello se diseñarán arquitecturas tanto centralizadas como distribuidas, y se realizarán las correspondientes verificaciones, estableciendo los porcentajes de mejora de rendimiento para cada arquitectura.

Palabras Clave: Recuperación de Información, HDFS, MapReduce, cluster, Paralelización, Escalabilidad, Rendimiento.

Contexto

Esta presentación corresponde al proyecto de investigación “Técnicas de recuperación de información en grandes volúmenes de datos heterogéneos con bases de datos No-Sql” desarrollado por docentes y alumnos de la Facultad de Ingeniería de la Universidad Nacional de La Patagonia San Juan Bosco (UNPSJB) Sede Puerto Madryn. El proyecto es financiado por la Secretaría de Ciencia y Técnica de dicha Universidad y se vincula con el III-LIDI Facultad de Informática de la Universidad Nacional de La Plata a través del director quien se desempeña como investigador en dicho laboratorio.

Introducción

La popularidad de las redes sociales, los sistemas de gestión de contenidos (CMS, Content Management Systems) como portales en general y como plataformas de colaboración, el crecimiento en la producción de información dentro de las organizaciones ya sea por producción de los sistemas o por la digitalización de información existente han generado un crecimiento exponencial en los volúmenes de información producidos, pasando de hablar de gigabyte de información a hablar con total normalidad del orden de los petabyte^{[13][14][15]}.

Esta situación y las aplicaciones en cuestión suelen requerir mayor cantidad de recursos que los que están disponibles en una sola unidad de cómputo. El desafío se encuentra por lo tanto en producir infraestructura de cómputo que permita re-utilizar las unidades existentes, normalmente heterogéneas entre sí, de forma colaborativa, alcanzando capacidades de disponibilidad y escalabilidad acordes al incremento en el procesamiento de la información^{[1][5]}.

La re-utilización de equipamiento de bajo costo de forma colaborativa que permita abordar esta problemática, requiere de técnicas de sistemas distribuidos, donde cada servidor cuente con capacidad de almacenamiento y cómputo local que permita que procesos y acceso a datos puedan ser distribuidos y balanceados en el cluster heterogéneo^[8].

Indagando distintas alternativas para solucionar la distribución de información y de las búsquedas en un ambiente heterogéneo y escalable se definieron un conjunto de propiedades deseables que debiera cumplir una solución^{[1][2][3][16]}: Rendimiento, Tolerancia a Fallas y Ejecución en ambientes heterogéneos. Por otra parte, las soluciones NoSQL para administrar grandes volúmenes de información se basan normalmente en la conformación de un sistema de nodos heterogéneos.

Existen diversas técnicas que permiten configurar ambientes heterogéneos y/o mixtos.

En nuestro caso particular de DFS, hemos concentrado el esfuerzo del presente trabajo en el Framework programado en Java llamado Hadoop para almacenar y procesar grandes volúmenes de datos a través de grupos de unidades de cómputo y almacenamiento, llamado Hadoop Distributed File System (HDFS).

El HDFS además de ser un sistema de archivos distribuido, escalable y portable, resuelve el manejo de disponibilidad y confiabilidad replicando los datos en múltiples servidores^[8].

Dentro de algunas de las características a destacar de HDFS, podemos mencionar:

Replicación con balanceo de carga

La arquitectura debe garantizar un conjunto de nodos con la información replicada y consistente en todos los nodos. En este caso el motor de búsqueda cuenta con un pooling de nodos de datos en los cuales buscar la información. La arquitectura misma no paraleliza las búsquedas, simplemente distribuye la carga entre los nodos. Como los nodos son independientes y auto-suficientes son capaces de responder consistentemente a la consulta realizada ya que la responsabilidad de recuperación está en el nodo de datos y la responsabilidad de distribución de carga en el balanceador. La desventaja de este método es que ni resuelve el problema espacial de la información ni paraleliza la búsqueda^[3].

Scatter and Gather

El método realiza un broadcast de la búsqueda de información requerida en sus nodos conocidos, realizando una dispersión de la misma. Cada nodo (independiente de los demás) tiene la capacidad de elaborar una respuesta con la información que contiene dicho nodo. Todas las respuestas se concentran en el nodo que realizó la dispersión y éste es responsable de consolidar las mismas en una única (y consistente) respuesta a la petición. Una ventaja adicional del método es que a su vez los nodos de datos pueden ser dispersores en nuevos nodos (conocidos por él). Conformando de esta forma una red de nodos independientes que contienen información.

Las ventajas en este caso son el particionamiento de la información y la paralelización de las búsquedas. La desventaja es una sobrecarga en la distribución de la información, especialmente si se desea realizar con alguna lógica de segmentación en particular: Geográfica, tipo de contenido, atributos ontológicos, etc. En este último caso se requiere conocimiento e información sobre los contenidos (datos) a ser almacenados, siendo en algunos casos relativamente compleja su resolución, especialmente ante la aplicación de reglas ontológicas sobre los contenidos^{[2][3][7]}.

Para este caso es interesante poder aplicar la técnica de Map / Reduce que es una buena técnica para procesar gran volumen de datos en paralelo. El modelo provee un

mecanismo de particionamiento de información que permite distribuir “inteligentemente” de acuerdo a reglas pre-definidas los datos en distintos nodos auto-contenidos.

Map /reduce es una técnica que implica paralelizar los datos clasificados por claves. Esto permite además paralelizar el procesamiento en las búsquedas, donde la clave de la técnica se basa en la inteligencia de separación de los datos. A su vez una ventaja adicional radica en el ahorro de espacio en el resultado de las claves compartidas al reducirlas dentro de un documento^{[8][9][17][18]}.

Líneas de Investigación y Desarrollo

Las principales líneas de investigación se podrían resumir en la lista siguiente:

- Determinar la factibilidad y aplicabilidad de los métodos teóricos en los entornos prácticos estudiados, especialmente en lo referido a la distribución de información e information retrieval.
- Proponer mejoras o nuevas técnicas y/o reformulaciones a las técnicas existentes para el manejo de recursos, en lo que se refiere a las técnicas de distribución y recuperación.
- Implementar y validar las técnicas y métodos propuestos sobre plataformas de desarrollo concretas.
- Seleccionar material bibliográfico y generar una base de conocimiento sobre las técnicas y métodos empleados en los esquemas de particionamiento, replicación, distribución e indexado en la infraestructura apache Hadoop.

Resultados y Objetivos

- Diseñar distintas arquitecturas del cluster con HDFS variando la cantidad de nodos del cluster y analizar los tiempos de respuestas.
- Investigar y seleccionar uno o varios métodos de particionamiento y replicación sobre Apache Hadoop.
- Verificar el esquema de escalabilidad y desempeño del cluster.
- Evaluar las distintas opciones de parametrización de la plataforma y evaluar distintas arquitecturas de configuración en Hadoop para una optimización.
- Definir métricas que permitan obtener conclusiones relevantes respecto a las técnicas y métodos implementados.
- Definir y desarrollar uno o varios métodos de pruebas de stress sistematizadas para someter a comparación las distintas configuraciones y parametrizaciones.

- Armar un banco de pruebas que permita comprobar las distintas implementaciones y métodos utilizados en el DFS seleccionado.

En la actualidad se ha logrado configurar un cluster HDFS de 8 equipos heterogéneos con un funcionamiento estable. Para ello fue necesario superar los inconvenientes en el armado de este tipo de infraestructura desde el punto de vista espacial en nuestra Unidad Académica como de conseguir los recursos necesarios para poder llevarla adelante.

El cluster hadoop cuenta en la actualidad con la siguiente configuración de equipos inter-conectados mediante un switch de 1Gb:

- PC1: Procesador: AMD Turion(tm) X2 Dual-Core Mobile RM-72 - Memoria: 4 GB - Disco: 320.1 GB. NameNode y DataNode.
- PC2: Procesador: Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz - Memoria: 10 GB- Disco: 500.1 GB. DataNode.
- PC 3: Procesador: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz - Memoria: 16 GB - Disco: 500.1 GB. DataNode.
- PC 4: Procesador: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz - Memoria: 8 GB - Disco: 1000.0 GB. DataNode.
- PC 5: Procesador: Intel(R) Celeron(R) D CPU 3.20GHz - Memoria: 685 MB - Disco: 80.0 GB. DataNode.
- PC 6: Procesador: Intel(R) Celeron(R) D CPU 3.20GHz - Memoria: 877 MB - Disco: 80.0 GB
- PC 7: Procesador: Intel(R) Celeron(R) D CPU 3.20GHz - Memoria: 685 MB - Disco: 80.0 GB. DataNode.
- PC 8: Procesador: Intel(R) Core(TM)2 Duo CPU E4400 @ 2.00GHz - Memoria: 1 GB - Disco: 160.0 GB. DataNode.

Sobre el cluster descrito se está comenzando a diseñar las pruebas de carga y stress que permitan determinar conclusiones sobre las distintas opciones de configuración del cluster Hadoop, respecto de la disponibilidad y la escalabilidad.

Las pruebas de carga se realizarán utilizando las herramientas siege y sar de linux. Los resultados serán publicados oportunamente.

Formación de Recursos Humanos

En lo referido a Formación de Recursos Humanos este proyecto propone las siguientes metas:

- Consolidar mediante el proyecto, un grupo de investigación de la Universidad Nacional de la Patagonia San Juan Bosco sede Puerto Madryn, sobre la disciplina Bases de Datos NoSql. Este grupo se integra actualmente de 4 profesores, 1

JTP y 1 Auxiliar, además participan del mismo 4 alumnos del ciclo superior de los cuales 2 alumnos están realizando el presente trabajo de Tesis.

- Fomentar, incentivar y difundir las tareas de investigación.
- Fomentar la realización de las Tesinas de Grado en el marco del proyecto de investigación. En particular el presente trabajo es el resultado de la ejecución del trabajo final de carrera de 2 de los autores
- Mejorar la formación de recursos humanos altamente calificados, con capacidades de investigación y desarrollo. Especialmente orientados a la inserción laboral altamente calificada en nuestro medio productivo.
- Contribuir a la creación en un futuro Centro o Instituto en investigación informática.
- Interactuar con otros grupos de investigación de las sedes de la universidad y de otras universidades, en tareas conjuntas de investigación y desarrollo, como también en la formación de recursos humanos.
- Incrementar el número de proyectos acreditados y de trabajos publicados por la universidad y la sede.

Referencias

- [1] M.T. Özsu & P. Valduriez. "Principles of Distributed Database Systems, 2nd edition". Prentice-Hall, 1999. Sitio web: <http://softbase.uwaterloo.ca/~tozsu/ddbook/notes.html>
- [2] David Taniar & Clement H. C. Leung & Wenny Rahayu & Sushant Goel. "High Performance Parallel Database Processing and Grid Databases". John Wiley & Sons, 2008.
- [3] P. Valduriez. "Data Management and Parallel Processing". Chapman and Hall, 1992.
- [4] M.Cohn, "Succeeding with Agile: Software Development Using Scrum", Pearson Education, 2010.
- [5] Ahmed K. Elmagarmid & Marek Rusinkiewicz & Amit Sheth. "Management of Heterogeneous and Autonomous Database Systems". Morgan Kaufmann Publishers, 1999.
- [6] Kristina Chodorow & Michael Dirolf. "MongoDB: The Definitive Guide". O'Reilly, 2010.
- [7] Satnam Alag. "Collective Intelligence in Action". Manning Publication, 2009.
- [8] Jason Venner. "Pro Hadoop". Apress, 2009.
- [9] Tom White. "Hadoop: The Definitive Guide", O'Reilly, 2011.

- [10] Michael McCandless & Erik Hatcher. "Lucene in Action, Second Edition: Covers Apache Lucene 3.0". Manning Publication, 2010.
- [11] David Smiley & Eric Pugh. "Solr 1.4 Enterprise Search Server". Packt Publishing, 2009.
- [12] Erik Hatcher, Otis Gospodnetić. "Lucene in Action", 2nd. ed, Manning Publications Co. 2004.
- [13] WhiteHouse.gov Goes Drupal,
<http://personaldemocracy.com/node/15131>
- [14] Thoughts on the Whitehouse.gov switch to Drupal,
<http://radar.oreilly.com/2009/10/whitehouse-switch-drupal-opensource.html>
- [15] Cal Henderson: "Building Scalable Web Sites", O'Reilly Media, 2006
- [16] Ricky Ho: Scalable System Design Patterns, Pragmatic Programming Techniques.
<http://horicky.blogspot.com/2010/10/scalable-system-design-patterns.html>
- [17] Azza Abouzeid, Kamil BajdaPawlikowski, Daniel Abadi1, Avi Silberschatz, Alexander Rasin: HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads.
- [18] Jeffrey Dean and Sanjay Ghemawat: MapReduce: Simplified Data Processing on Large Clusters. Google Inc.
- [19] Michael Stonebraker: The Case for Shared Nothing. University of California, Berkeley, Ca.