

Distributed visualisation of oceanographic data using Apache Spark and LustreFS

Leonardo Astudillo Villalón & Camilo Nuñez Fernández

Departamento Informática
Universidad Federico Santa Maria

August 15th, 2020

Introducción

Introducción

- A medida que la tecnología aumenta y mejora, los datos generados por los instrumentos actuales, demandan un mayor uso de las tecnologías existentes.
- Unas de las principales fuentes de datos, son los sistemas sensoriales oceanográficos, capaces de captar mediciones de su entorno y transmitirlos en tiempo real.
- El procesamiento de estos datos demanda una plataforma capaz de recibir estas grandes cantidades de datos y procesarlos en tiempo real para su visualización y posterior análisis.
- Frente a esto, el presente trabajo busca contribuir en el pipeline de almacenamiento y procesamientos de estos datos, desarrollando una plataforma capaz de procesar y almacenar estos datos, utilizando las tecnologías de *Apache Spark*, *LustreFS* y *Jupyter Notebook*.

Background y Trabajos Relacionados

Background y Trabajos Relacionados

Background

- Las técnicas actuales de Big Data involucran múltiples métodos que van desde el almacenamiento eficiente de los datos a gran escala, hasta su procesamiento optimizado en unidades de computo limitadas pero potentes.
- *LustreFS* es el filesystem distribuido por excelencia para el almacenamiento eficiente de los datos a gran escala. Se encuentra en muchos de los clusters de HPC en el mundo, con decenas de petabytes(PB) de almacenamiento y cientos de gbps en sus redes.
- *Apache Spark* es un framework potente para clusters el cual funciona por medio de colecciones de elementos o RDD los cuales realizan cálculos en la memoria volátil de los clusters.

Background y Trabajos Relacionados

Trabajos Relacionados

- Tanto *LustreFS* como *Apache Spark* son utilizados en ambientes de producción e investigación, sin embargo, no fue posible encontrar trabajos indexados y publicados que los relaciones conjuntamente.
- Pero sí existen publicaciones que son compatibles en su implementación. Tal es caso de *Jovial*, una plataforma fullstack orientada al procesamiento de datos astronómicos, desarrollada por [1].
- Por otro lado *Apache Spark* ha servido en el creciente de desarrollo del computo en el análisis de datos oceanográficos, como puede ver visto en el trabajo de [2].
- Ambas investigaciones sirvieron de base para la elaboración de este proyecto.

Diseño del sistema

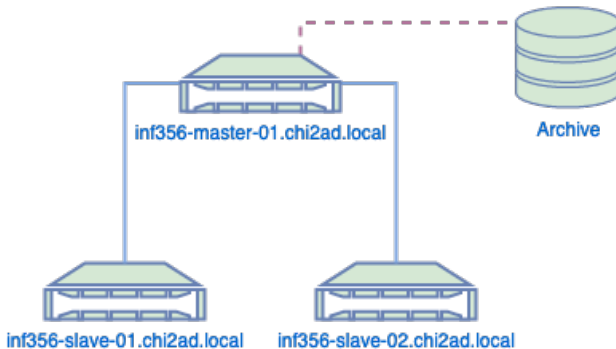
- Se busca desarrollar una plataforma fullstack dividida en tres componentes principales.
- Un modulo de almacenamiento, el cual corresponde al archive dispuesto en *LustreFS*, un modulo de computo con *Apache Spark*, un componente de frontend para edición orientada en *Jupyter*.
- Para el deployment de la arquitectura, se utilizaron 3 máquinas virtuales, cada máquina fue virtualizada por medio de Ovirt 4.1, con una configuración de 4 CPU, 8GM de RAM, y 50 GB en disco local. Las máquinas cuentan con una intefaz de InfiniBand virtualizada usando SR-IOV para tener acceso a la LNet de LustreFS. El sistema operativo usado fue Debian 10.

- Encargado de almacenar los datos oceanográficos que se descarguen desde el sistema Global RTOFS (Global Real-Time Ocean Forecast System).
- Para obtener y almacenar los datos, se programó un crawler en Python para descargar datos del sistema Global RTOFS.
- Acceso directo al `archive`.

- La implementación de *Apache Spark* fue por medio de un nodo maestro y dos workers; a su vez, el nodo maestro fue conectado al `archive`.
- Por defecto se usaron los compiladores e interpretes que contiene *Apache Spark* en sus módulos, particularmente Python 3 y PySpark.
- El objetivo de *Apache Spark* en el desarrollo de este proyecto fue crear las `SessionContext` con los cuales se logro paralelizar el procesamiento de las imágenes.

Diseño del sistema

Arquitectura



Diseño del sistema

Componente de Frontend

- El principal elemento de este componente es la implementación de *Jupyter Notebooks* para el desarrollo del código.
- En esta parte se desarrollaron los notebooks necesarios para el procesamiento de las imágenes.
- El componente de frontend tiene conexión directa al `archive`, debido a que los notebooks fueron ejecutados en un directorio dedicado dentro del filesystem.

Experimento

Experimento

Etapas de la tarea

Para el desarrollo de este experimento, primero se comentará las etapas de la tarea unitaria:

- Recolección de datos
- Procesamiento
- Visualización

Experimento

Recolección de datos

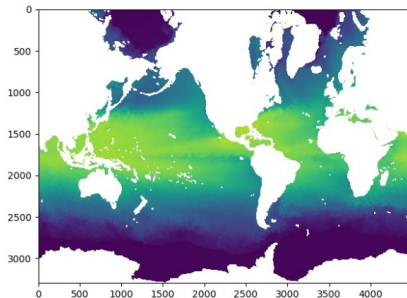
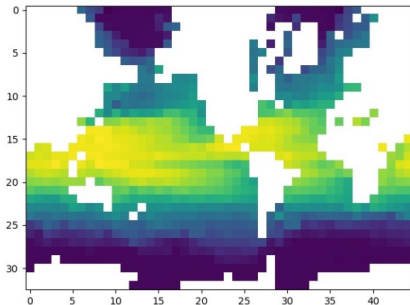
- Existe un web crawler encargado de obtener datos a las 15:00 hrs todos los días del sistema Global RTOFS
- Los datos obtenidos son archivos netCDF (.nc) [3]
- Estos registros contienen mediciones y metadata asociada a estas mediciones
- Cada archivo pesa alrededor de 0,5[GB] y son recibidos aproximadamente mas de 100 archivos

- Se utiliza la herramienta de Jupyter Notebook con kernel de python
- Se procesan los archivos con la librería *netCDF4*, la cual identifica los tipos de mediciones y se extraen los registros de temperatura de la superficie del mar (*Sea_Surface_Temperature*)
- Se utiliza la librería *xarray* para estructurar los datos y ser manejado por la librería *pandas*.

Experimento

Visualizaciones

- Se hace utiliza la función plot de la librería *matplotlib*
- Se genera dos tipos de imágenes, una de baja resolución y una de alta resolución



Experimento

Paralelización con Pyspark

- Esta tarea unitaria es ejecutada concurrentemente con todos los archivos nc que llegan al sistema
- Se utiliza el wrapper de Apache Spark en python llamado Pyspark para invocar el SessionContext
- Se paraleliza la tarea de procesamiento y visualización para cada archivo nc entregado por el web crawler con la función parallelize()

- Esta propuesta es más viable que un sistema clásico de procesamiento serial
- Varias propiedades que tiene este sistema permiten generar dashboards mas completos, mostrando de mejor manera la información del momento.

Conclusión

Conclusiones Generales

- Se construyó esta plataforma distribuida con *Apache Spark* y *LustreFS* con el fin de procesar datos oceanográficos en tiempo real.
- Se aprovechó de manera exitosa herramientas como el filesystem *LustreFS* para el almacenamiento de datos y la herramienta de procesamiento distribuido *Apache Spark* para coordinar las tareas de computación entre dos nodos workers y un nodo master
- Se cumplió el objetivo de presentar un ambiente de data science con Jupyter Notebook

Conclusiones

Posibles cambios y Trabajo Futuro

- La propuesta presentada es un prototipo en ambiente de testing lo que en un futuro se podría trabajar con un entorno de producción real
- Para un trabajo futuro se plantea la opción de utilizar *JupyterHub* para habilitar múltiples usuarios concurrentes, parecido al proyecto de Cloud Service hecho por ChiVO labs [4]

- El desarrollo de esta investigación fue gracias a la infraestructura proporcionada por el Chilean Virtual Observatory, y el asesoramiento de su grupo de system administrators.

Referencias



Araya, M., Osorio, M., Díaz, M., Ponce, C., Villanueva, M., Valenzuela, C., and Solar, M., “Jovial: Notebook-based astronomical data analysis in the cloud,” *Astronomy and Computing* **25**, 110 – 117 (2018).



Dahal, J., Ioup, E., Arifuzzaman, S., and Abdelguerfi, M., “Distributed streaming analytics on large-scale oceanographic data using apache spark,” (2019).



Rew, R. and Davis, G., “Netcdf: an interface for scientific data access,” *IEEE computer graphics and applications* **10**(4), 76–82 (1990).



Sr., H. A. F., Sr., D. O., Núñez, C., Sr., M. S., and Bugueno, M., “ChiVOLabs: cloud service that offer interactive environment for reprocessing astronomical data,” in [Software and Cyberinfrastructure for Astronomy V], Guzman, J. C. and Ibsen, J., eds., **10707**, 787 – 796, International Society for Optics and Photonics, SPIE (2018).