Luis Miguel Benitez

994361

Reporte de lectura

Parallel and Distributed Astronomical Data Analysis on Grid Datafarm.

En años recientes, la cantidad de datos obtenidos de observaciones astronómicas ha aumentado a escalas de Peta-bytes. La gran cantidad de información disponible tiene un gran potencial de nuevos descubrimientos, sin embargo el gran problema es que no existe una infraestructura capaz de proveer un ambiente adecuado para el análisis de los datos en su totalidad, por lo que los estudios actuales se realizan solo sobre las observaciones que correspondan a objetos ya conocidos.

Realizar análisis sobre la totalidad de los datos ofrecería la posibilidad de descubrir nuevos objetos, encontrar clases nuevas de objetos ya conocidos, o bien encontrar más datos acerca de tipos de objetos exóticos como emisores de rayos gamma o quásares,

Como ejemplo se pone el telescopio SUBARU, que genera aproximadamente 20 TBs de datos nuevos por año. Asimismo se menciona que para realizar un análisis sobre los datos, se requiere aproximadamente 10 veces el espacio que se necesita para solo almacenarlos.

Para resolver este problema se propone un Grid para almacenar la información obtenida por los instrumentos de observación, y capaz de realizar ciertas operaciones estándar de calibración de datos. Estas operaciones se pueden paralelizar a diferentes niveles, y se explican poniendo como ejemplo el telescopio SUBARU.

Este telescopio tiene una cámara principal compuesta por 10 detectores, y cada uno genera una imagen en formato FITS (un formato específicamente usado para imágenes de observaciones astronómicas). Así que cada observación genera 10 imágenes FITS de aproximadamente 170 MBs.

El primer proceso (*osmed*) de la calibración consiste en substraer a la imagen el sesgo que tenga el convertidor A/D de la cámara. Esta operación analiza las 10 imágenes generadas por los detectores en una sola toma, y genera otras 10 imágenes. La segunda operación (*medianimg*) extrae la mediana de un conjunto de N imágenes. En este caso, se trabaja con 10 conjuntos de N imágenes, cada conjunto corresponde a uno de los detectores en la cámara principal, y contiene las imágenes capturadas por un mismo detector durante N observaciones.

Finalmente la operación *uppercut* elimina ciertos pixeles considerados erróneos de la imagen mediana, y eso concluye la generación de la imagen 'plana'. Estas tres operaciones son agrupadas en el procedimiento Pa.

El siguiente proceso, denominado Pb vuelve a iniciar con la imagen generada por el detector. El primer paso consiste en dividir la imagen del detector entre la imagen plana (*arithimg*) después se corrige la

distorsión óptica (*distcorr*) y finalmente se substrae el fondo de la imagen (*skysb*) es decir, el cielo sin objetos observables.

Los autores proponen Gfarm, un grid capaz de proveer almacenamiento distribuido de archivos, así como procesamiento paralelo y distribuido sobre los mismos. El mecanismo principal en que Gfarm soporta esto es a través de archivos denominados también tipo Gfarm, un proceso de scheduling basado en la ubicación de los archivos, y una semántica de acceso paralelo a archivos.

Un archivo Gfarm es un grupo arbitrario de archivos con ubicación posiblemente distribuida, incluso en dominios administrativos diferentes. El acceso en paralelo se realiza cobre la *vista de archivo local* para el proceso que la requiera, y lo que hace es enviar las instrucciones al procesador que tenga los datos necesarios. El sistema de archivos se encarga de manejar las replicas de los archivos y almacenarlas de acuerdo al espacio disponible, así como de asignar los procesos en base a la carga de cada CPU que tenga una copia de los datos necesarios.

La forma como se implementa es a través de hooks a las llamadas del sistema operativo. La idea es evitar tener que modificar el código de los programas que analizan los datos. En vez de esto, se intercepta la llamada de apertura de archivo al sistema. En caso de que se trate de un archivo Gfarm, entonces esa llamada se maneja a través de los APIs de Gfarm, en caso contrario se deja continuar al sistema operativo.

En el proceso descrito anteriormente, que es con el que se evaluó el rendimiento de este modelo, la forma de alcanzar paralelismo es que cada archivo Gfarm contenga varios archivos FITS, y que cada uno de estos archivos FITS este almacenado en un nodo diferente, para que al procesarse cada nodo pueda trabajar en paralelo sobre los archivos que contenga. Pensando en esto, se incluyeron opciones en los comandos de Gfarm para que al crear un archivo Gfarm, el software intente almacenar cada archivo individual en un nodo diferente.

Las pruebas se realizan en un grid con 30 nodos, parte del Trans-Pacific Grid Datafarm. Cada nodo cuenta con dos CPUs Xeon de 2.8 Ghz, 1 GB de memoria y 1 cuadro discos duros de 200 GBs. Los datos de prueba consistieron en archivos Gfarm, con la particularidad que los archivos FITS que los componían eran todos idénticos, para eliminar irregularidades en el rendimiento entre nodos.

Los resultados sobre rendimiento en I/O se ejecutaron con un archivo Gfarm compuesto por 20 archivos FITS, y ejecutándose la operación osmed, que consiste en restar el sesgo del convertidor AD a la imagen obtenida, y escribirla imagen resultante al sistema de archivos. Se realizaron pruebas variando el número de procesos en paralelo realizando la operación sobre el mismo archivo Gfarm.

En los resultados se mide el ancho de banda tanto de lectura como de escritura en MB/s. En las graficas presentadas se observa que el ancho de banda de lectura crece de manera lineal conforme crece el número de procesos. El ancho de banda crece a un ritmo menor, y cuando los procesos son más de 30, es notorio que el aumento de ancho de banda es menor conforme se agregan más procesos. Esto debido a que hay 30 nodos presentes, y a partir de 30 procesos concurrentes empieza a haber

competencia por escritura. Sin embargo, se alcanzo un ancho de banda de 5.9 GB/s de lectura y 4.0 GB/s de escritura. Estos números indican la cantidad de datos que podrían procesarse bajo esta arquitectura.

Adicionalmente los autores realizan una prueba para medir el impacto del nivel de replicación de los archivos FITS dentro del archivo Gfarm. Para esto se utiliza el mismo archivo Gfarm y la operación *arithimg*, en la cual se divide la imagen obtenida del detector con su imagen 'plana' correspondiente, obtenida por el proceso Pa. En esta prueba solo se ejecutaron dos casos. El primero consiste en que solo uno de los nodos tiene una copia de la imagen 'plana', y en el segundo cada nodo tiene su propia replica local de este archivo. El tiempo de ejecución de arithimg con una sola copia del archivo y 10 procesos concurrentes es de 3.2 segundos, y de 0.5 segundos cuando cada nodo tiene su propia copia. Gfarm tiene la capacidad de de crear replicas locales de un archivo de forma dinámica.

Finalmente los autores mencionan que actualmente los datos se cargan al grid de Gfarm manualmente de un archivo existente, lo cual constituye un cuello de botella en el sistema actual. Se sugiere que idealmente los telescopios y demás dispositivos conectados al grid almacenen sus datos directamente en el grid de Gfarm.

## Referencias

Naotaka Yamamoto, O. T. (2004). Parallel and Distributed Astronomical Data Analysis on Grid Datafarm. *Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing.* Tsukuba,Japan: IEEE.