Sistema de abstracción de back-ends de almacenamiento para el procesamiento masivo de datos para sistemas HPC y Big Data

Pablo Brox, Javier Garcia-Blas, Jesus Carretero y David E. Singh

21 de Septiembre de 2021

uc3m

ARCOS

CABAHLA

Comunidad de Madrid

UNIÓN EUROPEA
Fondos Estructurales
Invertimos en su futuro
UNIÓN EUROPEA
Fondo Social Europeo
El Fondo Social Europeo invierte en tu futuro

Índice

Motivación

- Hoy en día, la mayoría de las aplicaciones de computación de alto rendimiento (HPC) emplean la interfaz estándar POSIX I/O para acceder a los datos almacenados en el sistema de archivos.
- Este estándar presenta múltiples problemas:
  - El manejo de estructuras de datos complejas,
  - Falta de control sobre los parámetros específicos del sistema de archivos,
  - Dependencia de la capa de datos del sistema operativo.
- Se prevé que las nuevas interfaces de E/S expongan optimizaciones y parámetros de ajuste a los usuarios finales con el objetivo de explotar el rendimiento de las aplicaciones HPC.

Objetivos y características

- Diseño e implementación de una nueva solución basada en contenedores de datos para aplicaciones intensivas en datos.
  - Interfaz de programación de alto nivel que permite sinergias entre los dominios de HPC y BigData,
  - Amplía la interfaz estándar POSIX y oculta la complejidad de tratar con diferentes patrones y fuentes de almacenamiento,
  - Abstracción presentada se basa en una serie de plugins activables que facilitan su extensión a nuevos sistemas de almacenamiento. Optimizaciones:
  - Gestión de memoria mediante operadores de movimiento de C++,
    Programación genérica basada en plantillas (*templates*),
    - Sistema de eliminación de interferencia de E/S simultánea.
    -

Contenedores de Datos

- **Abstracciones de alto nivel:**

  Ocultan la complejidad de usar varios sistemas de almacenamiento. Ofrecen un conjunto estándar de operaciones sencillas y fáciles de usar.

- **Distintos Tipos**:
  - En función de su cardinalidad: de *dataset* y de colección.
  - Dependiendo del tipo de datos: binarios y de texto.
  - Dependiendo de su uso: de entrada y de salida (*flusher*).

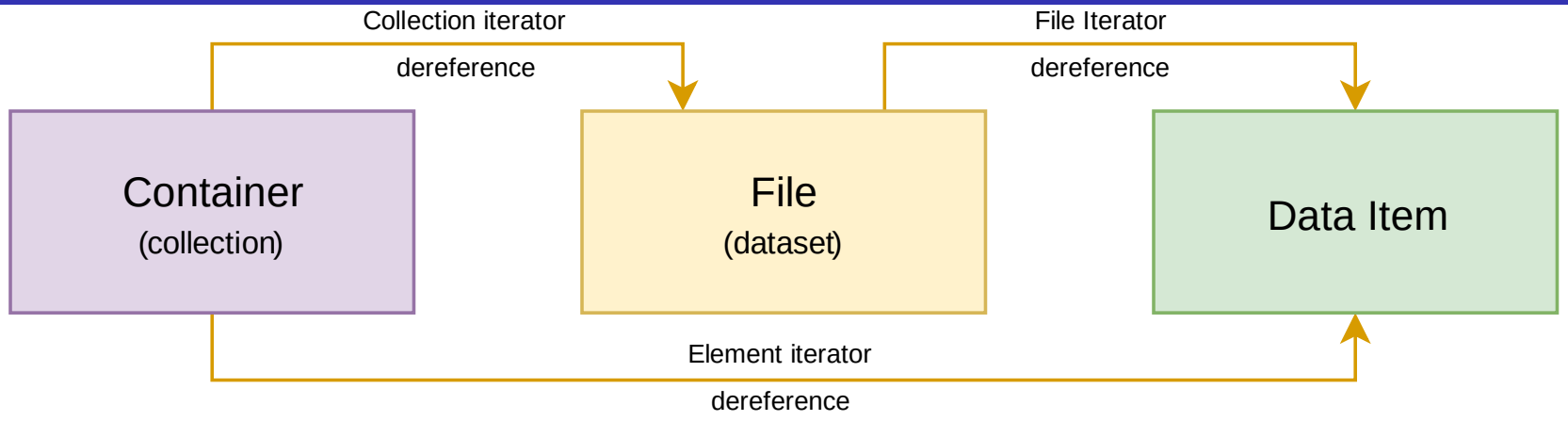
Contenedores de Entrada

-
-

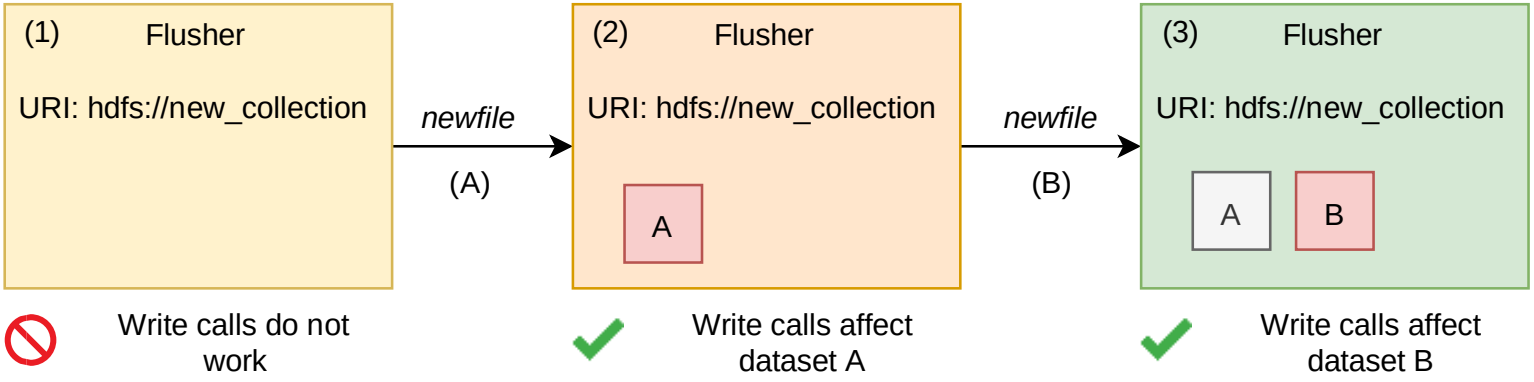
  Se usan para extraer datos ya existentes.

Acceso basado en iteradores.

Contenedores de Salida



- Se usan para escribir bloques binarios o cadenas en un *dataset*.
- Crean *datasets* consecutivamente dentro de una colección.

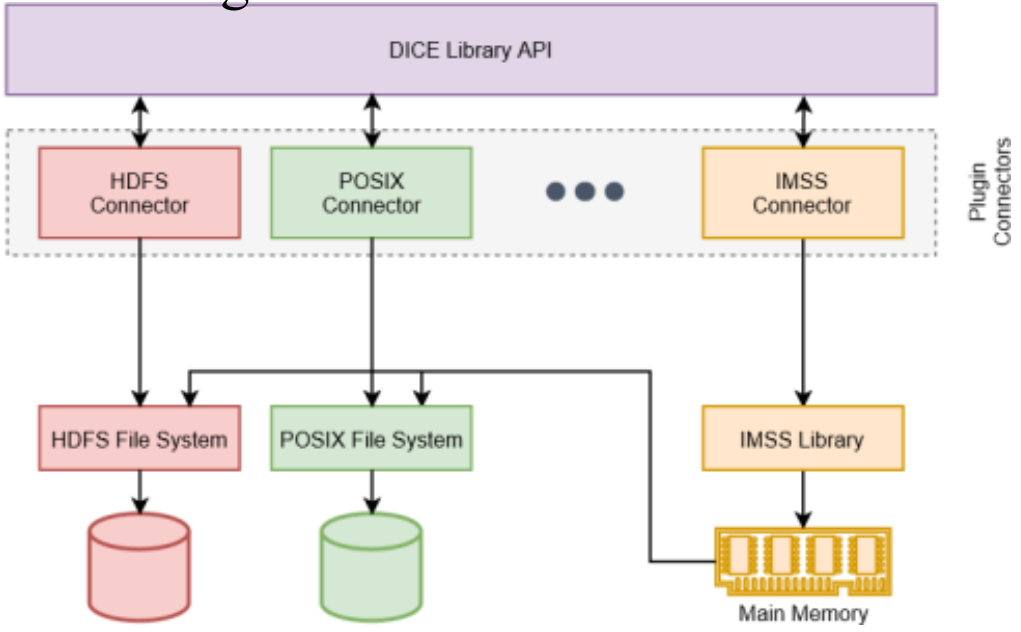


Integración

- *Plug-ins* que posibilitan el acceso a los distintos sistemas de almacenamiento.
- Activables en tiempo de compilación.
- POSIX I/O, HDFS e IMSS.

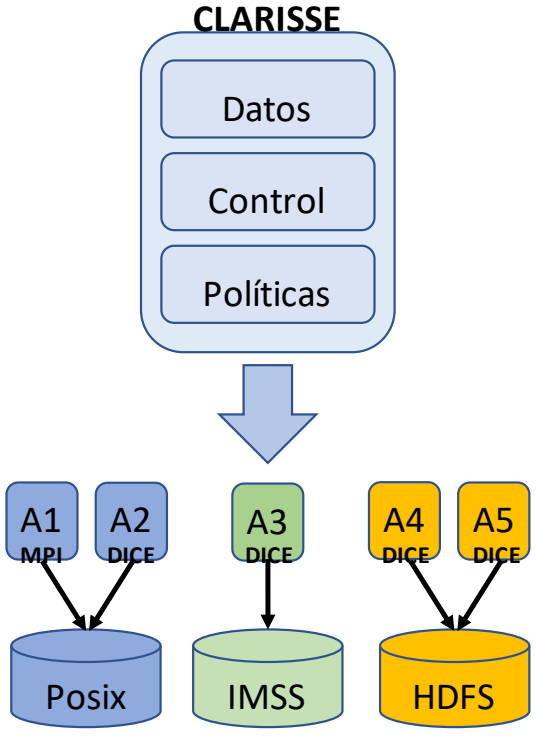
Ejemplo de Uso

- Conteo de 'hashtags' en una colección de *tweets*.



```
//Contendor de entrada para leer los datos
text_in_container in("hdfs://tw",'\n'); //Flusher para
escribir los resultados flusher
results("hdfs://results_ht");

//Expresión regular para detectar '#'
regex rgx("\\#\\w+"); map<string,int> results;

//Iterar todas las líneas for(auto line :
mytest)
    //Buscando el 'hashtag' en cada caso
    for( sregex_iterator it(line.begin(), line.end(), rgx), it_end; it != it_end; ++it ) results[(*it)[0]]++;

//Escribir los resultados para cada 'hashtag' resultsw.newfile("ht_res");

for (auto& kv: results) resultsw << kv.first + ", " + to_string(kv.second) + "\n";
```

Integración con Clarisse



- Integración con CLARISSE, middleware que proporciona coordinación y control de E/S.
-
  - *Gestión de datos* (no utilizada en este

trabajo) permite la creación de una área de intermedia de almacenamiento.
- *Capa de control*: coordinación de las operaciones en E/S entre distintas aplicaciones.
- *Capa de políticas de E/S*: establece un orden en la realización de la E/S.

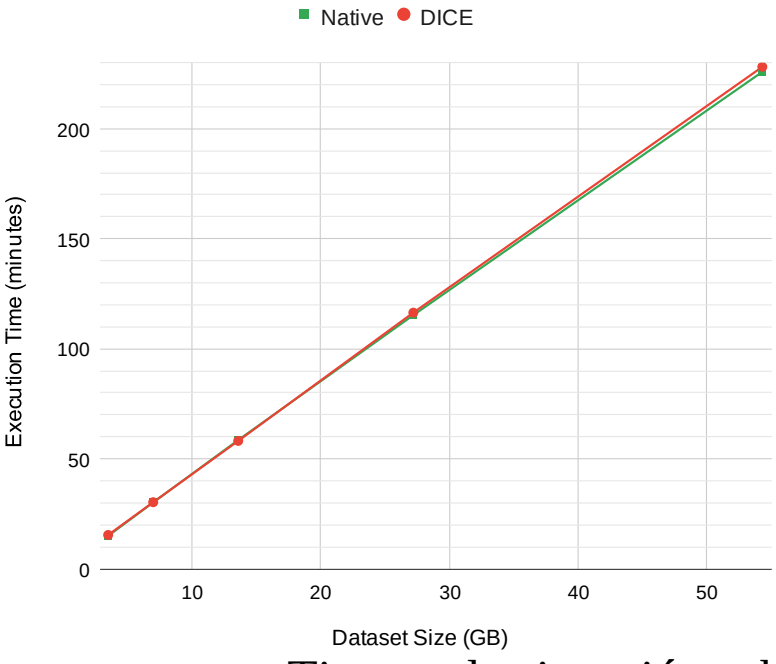
La integración con las aplicaciones se realiza de forma transparente.

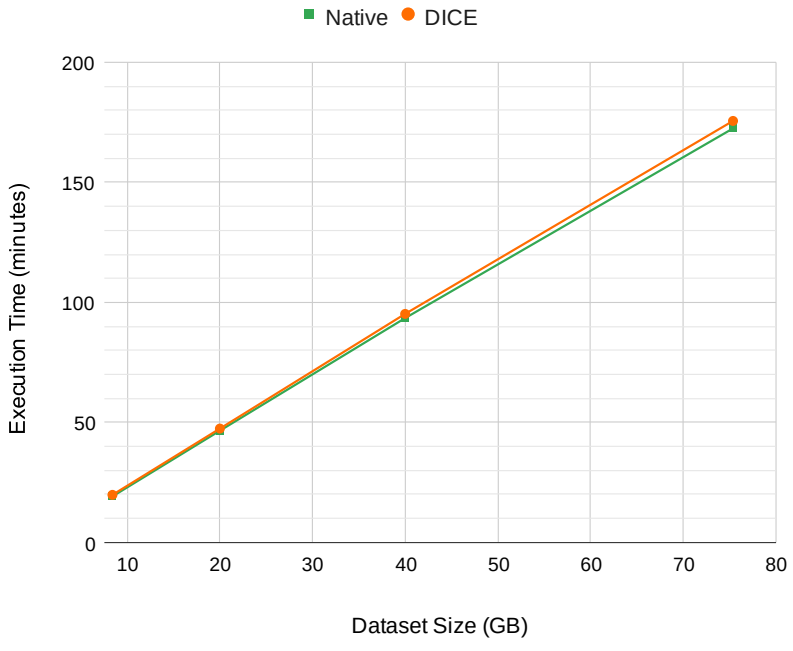
Configuración del Experimento

- Computador en clúster con Ubuntu Server 18.04 LTS.
  - Intel Xeon Gold 6212 con 24 núcleos.
- Sistema de almacenamiento **HDFS** con 26 nodos.
- Sistema de almacenamiento **GLUSTER** con 4 nodos.
- Conectados con red de **10-gigabit**.

Evaluación experimental (I)

- Conteo de ‘hashtags’ en una colección de *tweets*.

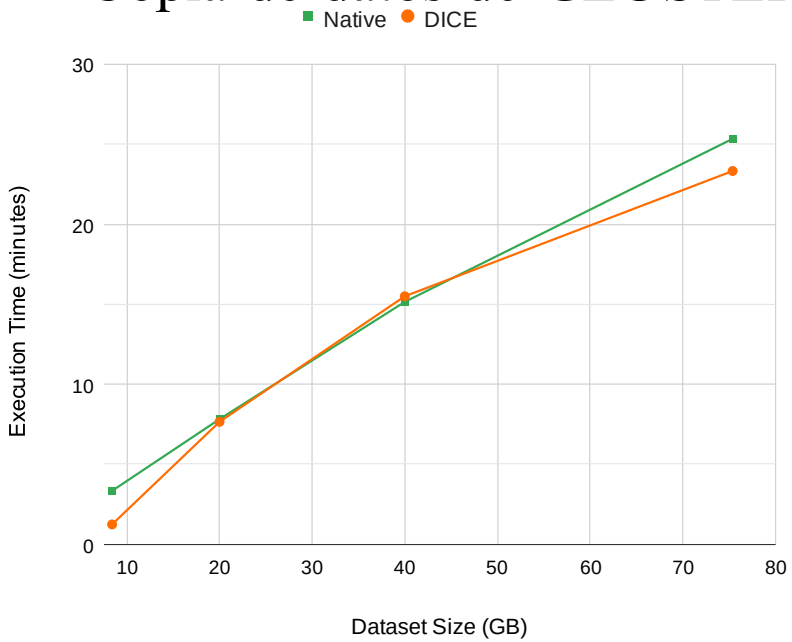


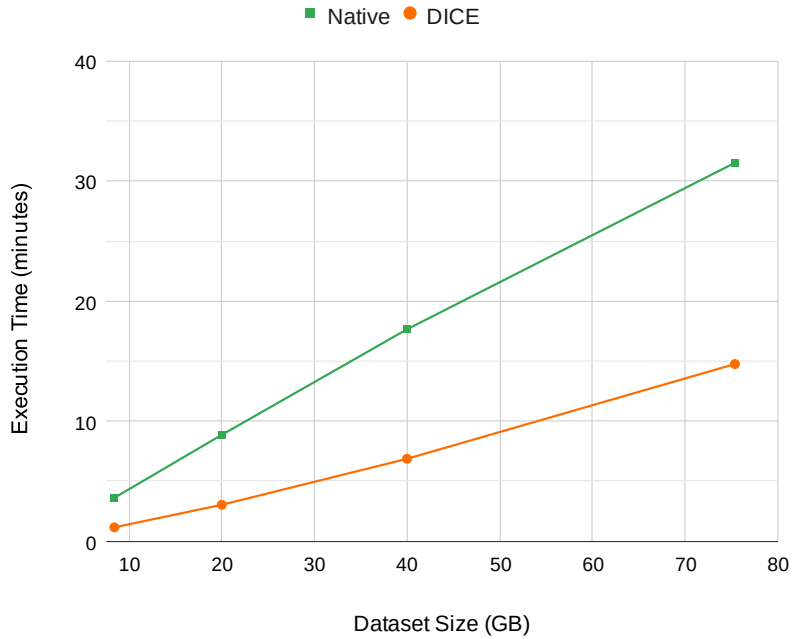


Tiempo de ejecución sobre Tiempo de ejecución sobre HDFS.
GLUSTER.

Evaluación experimental (II)

- Copia de datos de GLUSTER a HDFS.

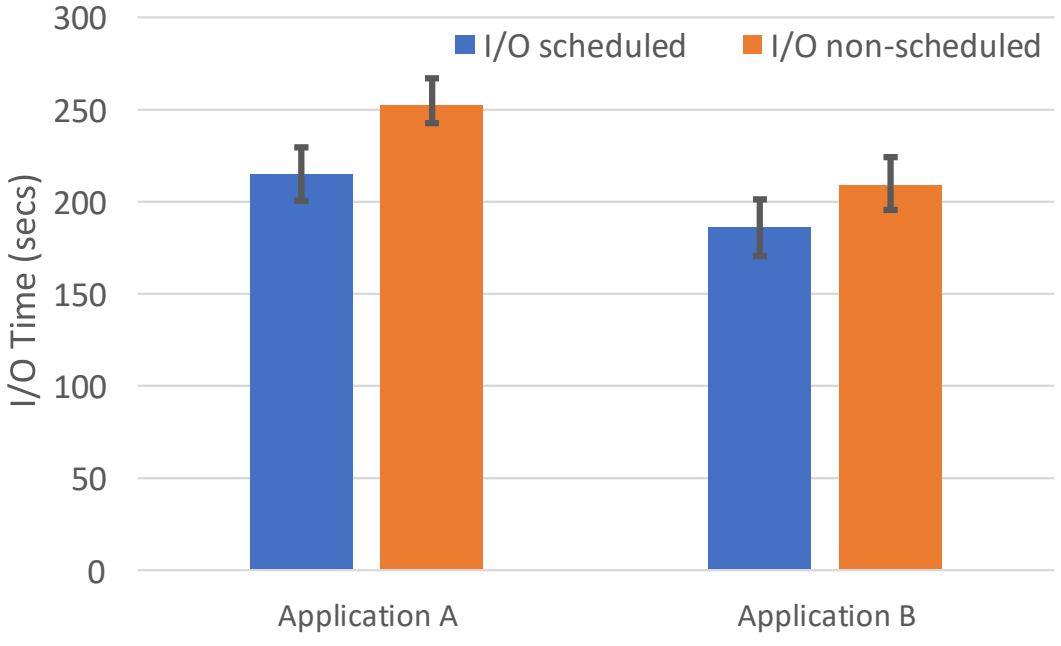




Time for GLUSTER to HDFS copy. Time for HDFS to GLUSTER copy.

Evaluación experimental (y III)

- Eliminación de interferencias en aplicaciones intensas en datos que se ejecutan de forma simultánea.



Conclusiones

- Hemos presentado DICE, una abstracción de acceso a datos genérica para unificar los mundos de HPC y Big Data.
- Mostramos la integración con CLARISSE, que nos permite evitar las interferencias de E/S y reducir la latencia.
- La evaluación experimental ha demostrado ser eficiente en ambos dominios con un sobrecoste menor al 2%.
- Colaboración con UPM en la implementación de un *plug-in* para Rados/Ceph.

Agradecimientos

"ASPIDE: Exascale Programming Models for Extreme Data Processing" con la subvención 801091 de la Comisión Europea.

"Convergencia Big Data-HPC: de los sensores a las Aplicaciones. (CABAHLA-CM)" apoyada por la Comunidad de Madrid.

"New Data Intensive Computing Methods for High-End and Edge Computing Platforms (DECIDE)" Ref. PID2019-107858GB-I00.


ASPIDE


CABAHLA


DECIDE


HORIZON 2020

Sistema de abstracción de back-ends de almacenamiento para el procesamiento masivo de datos para sistemas HPC y Big Data

Pablo Brox, Javier Garcia-Blas, Jesus Carretero y David E. Singh

21 de Septiembre de 2021

uc3m