



Cuba Salud

IV Convención
Internacional de Salud
17-21 de octubre, 2022

Herramienta para la gestión de colas de procesamientos de datos de neurociencias en HPC

Michel Suárez Morales
Dr.C. Arturo Orellana García

Universidad de las Ciencias Informáticas, La Habana, Cuba, contacto: aorellana@uci.cu

Resumen: En el campo de la neurociencia se genera una cuantiosa cantidad de información de investigaciones neurológicas, las cuales pueden ser procesadas gracias a la evolución de las infraestructuras computacionales y las nuevas tecnologías, y mediante el uso de la Computación de alto rendimiento (HPC). Lo anterior es realizado mediante el uso de comandos en una terminal segura (SSH), lo cual requiere un alto conocimiento que la mayoría de los especialistas de este campo no disponen, y pueden ocurrir errores humanos a la hora de ejecutar tareas de neurociencia.

La presente investigación tiene como objetivo desarrollar una herramienta para la gestión de colas de procesamientos de datos de neurociencias en un HPC. Para ello se realizó un análisis de distintas investigaciones previas que permitieron identificar las mejores variantes a tener en cuenta para la propuesta de solución. Como ambiente de desarrollo se definieron un conjunto de herramientas y tecnologías idóneas para el contexto de la investigación como: Python, FastAPI, Javascript, Vue.js, WebSocket, entre otras; además se empleó como herramienta de desarrollo PyCharm Community.

A partir de lo anterior se obtuvo una herramienta amigable e intuitiva que facilita el trabajo de los especialistas, incrementa la productividad y favorece la ejecución de tareas de manera correcta y fácil.

Palabras clave: Procesamiento, datos de neurociencia, clúster, HPC.

I. INTRODUCCIÓN

Con la mejora del hardware, el aumento del ancho de banda de la red y el desarrollo de algoritmos más sofisticados, las arquitecturas paralelas iniciaron una quinta generación que incrementó significativamente la eficiencia de la computación. La canalización (pipelining) y el almacenamiento en caché fueron, a su vez, dos de las técnicas de hardware que aumentaron el rendimiento computacional, conduciendo a mayores velocidades de procesamiento. (1)

La era de los clústeres es casi un eufemismo para indicar que la computación ha entrado en una etapa en la cual el procesamiento, de aplicaciones y del software de base, deja de ser secuencial, para entrar en la era de la concurrencia, que incluye paralelismo, simultaneidad y pipeline. Dado que las computadoras hogareñas y los teléfonos celulares, ya tienen más de un núcleo, es más que evidente que se debe mirar con entusiasmo todo aquello que sea paralelo. El concepto de concurrencia está pensado para que las ejecuciones dejen de ser secuenciales y se ejecuten en alguno de los tres modos que implica la concurrencia; es el afán de resolver las cosas más rápido, y para lograrlo se dividen los problemas en partes que sean independientes pero que a su vez sean cooperantes... o sea: los paradigmas son dividir y cooperar.

Un clúster, si se mira de una manera sencilla, es un conjunto de máquinas que resuelven un problema, dividido en partes, pero que intercambian información para cooperar en busca de un resultado en el menor tiempo operable. La forma de dividirlo es por medio de concurrencia sobre el dominio de datos, es decir, el mismo problema se ejecuta en varias máquinas sobre conjunto de datos distintos o sobre dominio de procesamiento... o sea piezas de software distintas que operan los mismos datos o distintos pero que en todos los casos cooperan para obtener un resultado. Fundamentalmente se requieren éstos para procesar información y obtener resultados en la forma más rápida posible. O sea: El objetivo de un clúster es que el conjunto de nodos, conectados por medio de un enlace, ejecuten subtareas (de una tarea ya dada) con el objetivo de aprovechar aquellas acciones que son concurrentes para superponerlas en tiempo y espacio de tal manera de acortar los tiempos de procesamiento de la tarea original, si esta fuese ejecutada secuencialmente. Esta forma de trabajo debe permitir que el usuario vea al clúster como un sistema único, y que el conjunto de subtareas, que son cooperantes para resolver la tarea, ejecutadas en nodos distintos, sumen las potencias de los nodos hasta obtener una capacidad de procesamiento como si esta tarea fuese ejecutada en un solo nodo igual a la capacidad de la suma de las capacidades de los nodos intervinientes. Está claro que esta capacidad depende fuertemente de la potencia de procesamiento de cada nodo, de la velocidad del enlace, de la forma de administrar los recursos y del algoritmo utilizado para resolver el problema que la tarea representa (abstrae) en su forma computacional. (2)

La tecnología clúster permite a las organizaciones incrementar su capacidad de procesamiento usando tecnología estándar, tanto en componentes de hardware como de software que pueden adquirirse a un costo relativamente bajo. Las aplicaciones pueden distribuirse en más de un equipo, consiguiendo un grado de paralelismo, una recuperación de errores y proporcionando más disponibilidad. El clúster aparece como la imagen de un único sistema para los usuarios finales, las aplicaciones y la red, proporcionando a la vez un único punto de control para los administradores, local o remotamente.(3)

El hecho de que sea necesario un alto conocimiento de línea de comandos o la supervisión de un experto, conlleva a un aumento en los costos y tiempo. La presente investigación tiene como objetivo desarrollar una herramienta para la gestión de colas de procesamientos de datos de neurociencias en HPC.

II. MÉTODO

Para la ejecución de la presente investigación se sigue una estrategia explicativa y se emplearon los métodos: análisis documental para obtener datos e información asociados al objeto de estudio. Se analizaron los documentos bibliográficos referentes a soluciones similares, sistemas de ejecución de procesos en un clúster, su integración, así como estándares y buenas prácticas, lo que permitió establecer los fundamentos teóricos. Mediante el método comparativo: establecer la comparación entre las diferentes librerías y marcos de trabajo, y poder establecer cuáles son los que van a utilizar para el desarrollo de la herramienta.

A. Descripción del funcionamiento de un Clúster

La propuesta ha sido diseñada para el procesamiento de datos de neurociencia a partir del estudio del funcionamiento de un Clúster. Para el uso del mismo, es necesario acceder mediante SSH al nodo maestro y ejecutar el comando o script correspondiente a la tarea a realizar en cuestión. Un script permite a los usuarios interactuar con el sistema, procesando las órdenes que se le indican (4). El clúster cuenta con un sistema de colas encargado de repartir las diferentes tareas por los nodos existentes. De este modo, el usuario, a través de comandos, sólo tendrá que lanzar su aplicación y el gestor de cola se encargará de encontrar nodos libres en los que ejecutar los cálculos. En caso de que estén todos ocupados, el gestor de colas se encargará de poner la tarea en espera y de lanzarla cuando haya recursos disponibles, sin necesidad de que el usuario esté conectado esperando. Las librerías propuestas para el lanzamiento de comandos vía SSH son las siguientes:

- **Pamiko:** Creado puramente en código Python, mientras que tiene extensiones nativas como dependencias, con bajo rendimiento y numerosos errores en comparación con los binarios de OpenSSH y los clientes nativos basados en libssh2. Las versiones recientes han retrocedido en el rendimiento y tienen problemas de bloqueo.
- **AsyncSSH:** Framework creado puramente en Python usando la biblioteca de cliente SSH. La licencia (EPL) no es compatible con GPL, BSD u otras licencias de código abierto y las obras combinadas no pueden ser distribuidas.
- **Ansible:** Una herramienta de gestión y automatización de la configuración que utiliza comandos remotos SSH. Utiliza, en parte, los binarios de Paramiko y OpenSSH. Utiliza subprocesos para la paralelización y adolece del pobre escalado que ofrece este modelo.
- **SSH2-python:** Enlazada con la biblioteca libssh2 escrito en C. No realiza la paralelización desde el primer momento, pero se puede hacer en paralelo a través de la biblioteca de subprocesos de Python con relativa facilidad y, como es un envoltorio para una biblioteca nativa que libera GIL de Python, puede escalar a múltiples núcleos.
- **Parallel-SSH:** Utiliza ssh2-python en su modo nativo sin bloqueo con bucle de eventos y sockets cooperativos proporcionados por gevent para una biblioteca de rendimiento extremadamente alto sin los efectos secundarios de la aplicación de parches mono. Además, utiliza subprocesos nativos para descargar tareas vinculadas a la CPU, como la autenticación, para poder escalar a varios núcleos sin bloquear la E/S de la red.

Esta última cumple en cuanto a rendimiento, adaptabilidad, escalabilidad y mantenibilidad necesario para el lanzamiento de las diferentes tareas desde la herramienta, proporcionando una biblioteca para

ejecutar comandos SSH asíncronos en paralelo con poca o ninguna carga inducida en el sistema al hacerlo con el uso previsto siendo completamente programático y no interactivo. (5)

B. Comparación de diferentes plataformas y sistemas homólogos

El estudio de las plataformas de gestión de datos de neurociencia que tiene como objetivos fundamentales determinar las características que pueden ser reutilizables y recopilar las buenas prácticas y los errores más comunes usados en proyectos ya establecidos en el campo. En la siguiente tabla se muestra la comparación de varias plataformas de almacenamiento y análisis de datos, y brindan información sobre los diferentes aspectos técnicos de cada una.

Tabla 1: Comparación entre diferentes características de las plataformas de neurociencia.

	CBRAIN	LORIS	BIL	C-Big	Neuro-Vault	Neuro-Morpho
Acceso al repositorio	SSL, SSH	SSL	SSH, Bridges-2, Neocortex, Sistema VM	SSH	SSH	SSH
Tipos de datos	Archivos planos, imágenes y metadatos textuales	Archivos planos, imágenes, metadatos textuales y visualizaciones 3D	Imágenes (tiff, jpg-2000), Datos de neuronas(swc)	EEG (EEG-BIDS), MEG(MEG-BIDS)	EEG, MEG(NIFTI)	Archivos planos, imágenes y metadatos textuales (swc)
Tecnologías / Lenguajes	No especificado	Linux, Apache, MySQL, PHP (LAMP)	No especificado.	LORIS, MySQL, React, JavaScript	Python, Django, Docker, JavaScript	MySQL, Java, C ++, Matlab, Apache Tomcat
Código abierto	No	No	No	No	Sí	No
Organización/ Institución a la que pertenece	Canadian Open Neuroscience Platform	Canadian Open Neuroscience Platform	Iniciativa BRAIN, Universidad de Pittsburgh	Canadian Open Neuroscience Platform	Neuroscience Information Framework	Neuroscience Information Framework

Algunas de estas plataformas, como por ejemplo CBRAIN en su repositorio de GitHub (6), hacen uso de Boutiques, un sistema para publicar, integrar y ejecutar automáticamente aplicaciones de línea de comandos en plataformas informáticas. Las aplicaciones de Boutiques se instalan a través de contenedores de software descritos en un lenguaje JSON rico y flexible. (7)

C. Descripción del proceso de lanzamiento de tareas en un clúster

Antes que nada, definamos los roles que interactúan en la herramienta:

- **Desarrollador:** Encargado de crear los scripts, añadir los servidores o clústeres disponibles y administrar el acceso a estos recursos.
- **Especialista:** Lanza los scripts que tiene a disposición con los parámetros correspondientes.

El proceso de lanzamiento de tareas de neurociencias en el clúster se compone de una secuencia fija de pasos. Los cuales son los siguientes:

1. El desarrollador crea los scripts y los diferentes clústeres.
2. El especialista incorpora los diferentes parámetros que requiere el script.
3. El especialista lanza el script en un clúster disponible.

III. RESULTADOS

Teniendo en cuenta lo mencionado anteriormente en la descripción del proceso de lanzamiento de tareas en un clúster. Se propone realizar una herramienta web que sea más amigable e intuitiva y facilite el trabajo que desarrollan los especialistas sin la auditoria de un desarrollador. Se utiliza Visual Paradigm como herramienta de modelado:

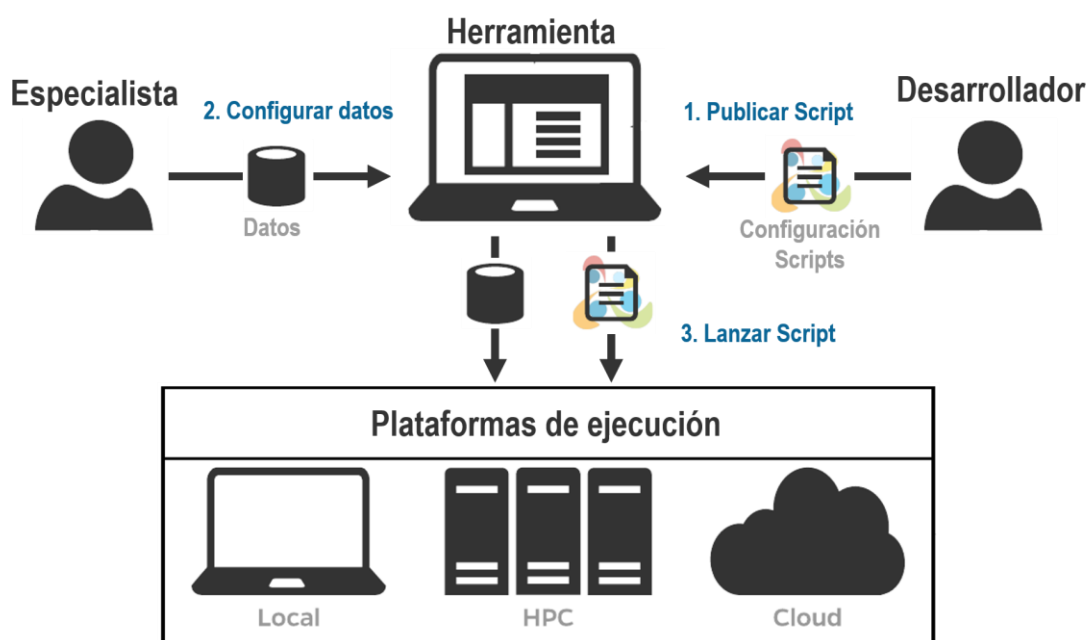


Figura 1. Arquitectura propuesta para el procesamiento de tareas de neurodatos. Fuente: los autores

El funcionamiento de la herramienta comienza a partir del desarrollador; este es el encargado de publicar los scripts que podrán configurar y lanzar los especialistas. Estos scripts pueden ser configurados con parámetros específicos o dinámicos, según las órdenes que se le indican. Una vez creados, es necesario un servidor o clúster donde ejecutarlos, por lo que también deben ser añadidos a la herramienta proporcionando la dirección y credenciales del mismo para su posterior comunicación vía SSH.

Una vez creado todos los componentes de ejecución y acceso al clúster, es hora del lanzamiento de las tareas de neurociencia. Los especialistas disponen de una interfaz amigable e intuitiva que facilita el trabajo, solo incorporan los parámetros de configuración del script y seleccionan el clúster al cual van a lanzar dicho script. Esto incrementa la productividad y favorece la ejecución de tareas de manera correcta y fácil, pues no es necesario que el desarrollador esté presente en el lanzamiento o configuración del script, basta con solo una vez al publicar el script.

Para el desarrollo de esta herramienta se emplearon las siguientes tecnologías y herramientas: Python, Vue.js, WebSocket, FastAPI, Javascript, entre otras. La solución implementada favorece la ejecución de

tareas de manera correcta, fácil e intuitiva. Se espera que con esta herramienta se beneficien los profesionales y los especialistas de la institución, y así incrementar la productividad del personal.

IV. CONCLUSIONES

Se definió el funcionamiento, los protocolos y componentes de un clúster analizando las diferentes librerías que existen para su uso.

Se determinó los diferentes roles y funciones que desempeñaran cada uno en la herramienta de procesamiento, así como el número de pautas a seguir para el lanzamiento de tareas de neurociencia.

Se puntualizaron las tecnologías, herramientas y lenguajes empleados para ejecutar la investigación.

AGRADECIMIENTOS

La investigación que da origen a los resultados presentados en la presente publicación recibió fondos de la Oficina de Gestión de Fondos y Proyectos Internacionales bajo el código PN305LH013-038.

REFERENCIAS

1. Oszlak O. El Estado en la era exponencial [Internet]. Instituto Nacional de Administración Pública; 2020 [citado 14 de julio de 2022]. Disponible en: <http://repositorio.cedes.org/handle/123456789/4549>
2. Bevilacqua R. Breve historia de la evolución de HPC en una institución de investigación en la Argentina. En 2020 [citado 14 de julio de 2022]. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/116778>
3. Rodríguez Pérez JF. Programación Matlab en paralelo sobre clúster computacional: evaluación de prestaciones. febrero de 2010 [citado 14 de julio de 2022]; Disponible en: <https://repositorio.upct.es/handle/10317/1849>
4. Francisco José Fernández Jiménez, Fco. Javier Muñoz Calle. Programación Shell-script en Linux [Internet]. Programación Shell-script en Linux. 2018 [citado 14 de julio de 2022]. Disponible en: http://trajano.us.es/~fjfj/shell/shellscrip.htm#_Toc444081192
5. Comparison With Alternatives — Parallel-SSH 0+unknown documentation [Internet]. [citado 14 de julio de 2022]. Disponible en: <https://parallel-ssh.readthedocs.io/en/stable/alternatives.html>
6. aces/cbrain-plugins-neuro [Internet]. ACElab; 2022 [citado 14 de julio de 2022]. Disponible en: <https://github.com/aces/cbrain-plugins-neuro>
7. Glatard T, Kiar G, Aumentado-Armstrong T, Beck N, Bellec P, Bernard R, et al. Boutiques: a flexible framework to integrate command-line applications in computing platforms. *GigaScience*. 1 de mayo de 2018;7(5):giy016.