

Políticas de gestión de colas

Introducción

Hasta la actualización del supercomputador en diciembre de 2017 el sistema de gestión de los trabajos en las colas de ejecución se basaba únicamente en el orden de petición al sistema. Es decir, el trabajo solicitado en primer lugar era el primer trabajo en ejecutarse. Mientras hubiera recursos disponibles, las peticiones iban ejecutándose.

Si todos los recursos del supercomputador estaban asignados a peticiones en ejecución, las siguientes peticiones quedaban pendientes a la espera de tener recursos disponibles.

Este sistema de trabajo al principio era suficiente, dado que el número de usuarios del clúster era bajo y no había interferencias notables.

Sin embargo, al aumentar el número de usuarios, así como el número de trabajos lanzados, se vio la necesidad de modificar esta política de gestión de recursos para evitar que un usuario colapsase el supercomputador y que el uso de los recursos fuese un poco más equitativo entre todos los usuarios.

Es por ello que se ha puesto en marcha un sistema de priorización basado en varios factores que se explicará en los siguientes apartados.

Recursos desde el punto de vista de la gestión de trabajos

En el presente documento, y en general siempre que hablemos de supercomputación, se entiende por recurso cualquier elemento del sistema susceptible de ser usado por un trabajo de modo que, durante su uso, cualquier otro trabajo no pueda disponer de él.

Entre estos recursos nos encontramos con los siguientes:

- Unidades de cómputo (núcleos, CPU,...).
- Tiempo de ejecución, medido en segundos de CPU.
- Memoria principal.
- GPU. Procesadores gráficos especializados en cálculos matriciales.
- Licencias de programas.

La lista puede ser mayor, pero los mostrados aquí son los principales.

Colas de ejecución

Nuestro clúster cuenta con dos tipos de nodos. La principal diferencia entre ambos es que el segundo tipo cuenta con tarjetas gráficas nVidia Tesla K20m que pueden ser usadas por programas especializados que hacen uso intensivo de cálculos matriciales.

Dado que este segundo grupo solo cuenta con dos nodos especializados, se ha decidido crear, y mantener separadas, dos colas:

- La cola c_n que agrupa a los otros 48 nodos para cálculos que no requieren las GPUs.
- La cola c_g que reúne los dos nodos restantes para cálculos que sí requieran el uso de GPUs.

Por defecto los trabajos irán a la primera cola y, si alguien necesita el uso de GPUs, puede indicarlo

con la opción correspondiente.

Prioridad multifactor

Después de estudiar los distintos factores aplicables a la prioridad de los trabajos en cola, se llegó a la conclusión de que los factores, y sus pesos relativos, deberían ser los siguientes:

- **Tamaño del trabajo.** Entendido como tal el número de recursos de cualquier tipo solicitados para el mismo. En general se tiende a favorecer los trabajos pequeños que acaban pronto. Su peso relativo es de 1.
- **Edad del trabajo.** Entendida como tal el tiempo que el trabajo lleva en cola sin haber sido ejecutado. A medida que pasa más tiempo, mayor será la prioridad asignada por este factor. Su peso relativo es de 5.
- **Fair-share.** Este factor es una medida de los recursos solicitados recientemente. A medida que se solicitan más recursos, menor es este factor, favoreciendo así que otros usuarios puedan lanzar sus propios trabajos. Su peso relativo es de 10.

Para calcular la prioridad resultante para un trabajo se multiplicaría cada factor por su peso y se sumarían los tres valores, obteniéndose así la prioridad del trabajo.

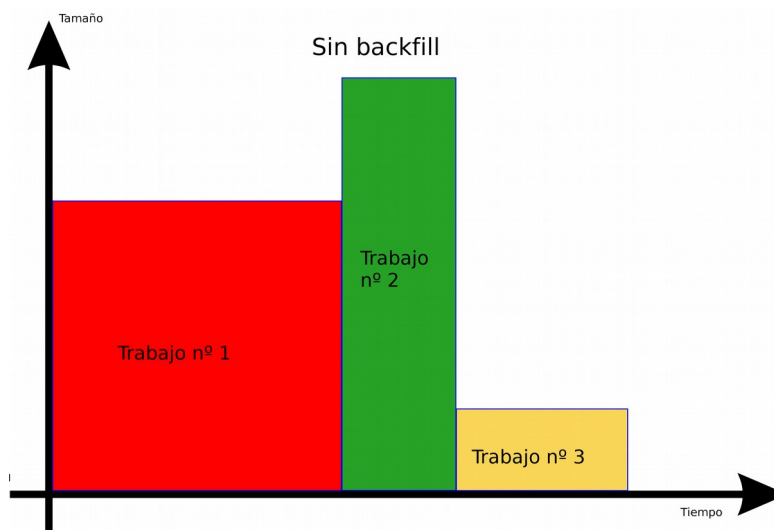
Backfill

Independientemente de lo expuesto arriba, se puede activar (actualmente lo está) el mecanismo de *backfill*, según el cual si un trabajo de menor prioridad puede saltarse la cola sin afectar por ello a los que están por delante de él, el sistema lo adelantará.

Ejemplo:

Veamos una cola de trabajos gestionada sin *backfill*. En la imagen, en el eje de abscisas se representa el tiempo y en el de ordenadas el tamaño del trabajo.

Mientras el trabajo n.º 1 está en ejecución llega el trabajo n.º 2 que, al no haber en el espacio libre, se queda en cola. El trabajo n.º 3 que viene detrás tendrá que esperar a que termine el n.º 2 a pesar de que sí cabría en el hueco.



En el caso de tener los mismos trabajos, pero con la cola usando el mecanismo de *backfill*, el trabajo n.º 3 se adelantaría y empezaría su ejecución inmediatamente.

