**Quicksort Paralelo**

Duran, Tatiana. Espeche, Marcos. Maya, Facundo. Prato, Joaquín.

Universidad Tecnológica Nacional, Facultad Regional Mendoza

Nota del autor

[Incluya información sobre becas o ayudas y una dirección postal completa.]

Resumen

[El resumen ha de tener una longitud de un párrafo de entre 150 y 250 palabras. sin sangría. Los títulos de sección, como la palabra Resumen anterior, no se consideran títulos, por lo que no se usa formato de título en negrita. En su lugar, use el estilo Título de sección. Este estilo inicia automáticamente la sección en una nueva página, por lo que no es necesario que agregue saltos de página. Tenga en cuenta que todos los estilos de esta plantilla están disponibles en la pestaña Inicio de la cinta, en la galería de estilos.]

Palabras clave: [Haga clic aquí para agregar palabras clave.]

Decisiones de diseño: (no olvidar mencionar que consideramos usar descomposicion recursiva)

Nuestra primera decisión fue el método de descomposición que usamos. Nos decantamos por utilizar un esquema de descomposición por dominio. Es el que mejor se adecua a la naturaleza del problema, debido a que el algoritmo a realizar por cada nodo es siempre el mismo pero sobre distintos conjuntos de datos, por lo que, además, podemos afirmar que dentro de las taxonomías estudiadas usamos un SPMD (*Single Program Multiple Data*). Respeto al algoritmo, implementamos un esquema *Master-Worker*, en el cual el *master* es el que recibe el arreglo inicial, define qué porciones del arreglo entregarle a cada *worker*, y luego recibir sus resultados para terminar de ordenar el arreglo. A su vez, cada *worker* ordena el arreglo recibido. Por lo tanto, nuestra implementación se desvirtua un poco de la idea original del algoritmo *quicksort*, pero nos parece que es una implementación más limpia, fácil de entender, y con un mejor balanceo de carga que con un esquema recursivo.

Cabe aclarar que el programa está hecho para usar al menos dos nodos, y que con una cantidad de datos significativamente pequeña comparada con la cantidad de nodos el rendimiento del programa se verá disminuido. Es más rentable implementar paralelismo cuando hay una cantidad de datos relativamente grande respecto de la cantidad de nodos.

Nos percatamos, cuando decidiamos cómo implementar las comunicaciones entre nodos, de la posibilidad de combinar *sends* y *recieves* bloqueantes y no bloqueantes. El nodo *master* puede hacer envíos no bloqueantes, ya no hay dependencia entre las porciones del arreglo principal enviadas a cada *worker*. Sin embargo, la recepción de datos en los *workers* sí debe ser bloqueante, porque no puede empezar a trabajar si no terminó de recibir el arreglo que tiene que ordenar.

Una vez que cada *worker* recibe su porción de arreglo, aplicará el algoritmo quicksort secuencial sobre él, con la distinción de que uno de los *workers* tendrá una porción de arreglo más grande (en caso de que la división del arreglo principal por la cantidad de *workers* no sea exacta). Una vez terminados este proceso de ordenamiento, cada *worker* tendrá su porción de arreglo ordenada, pero esto no quiere decir que las porciones de arreglo estén ordenadas entre sí.

Para poder reconstruir el arreglo ordenado, los *workers* envían al *master* sus porciones de arreglo ya ordenadas, y el *master* las recibe y las organiza en pares de “listas”. Luego, el *master* envía 1 par de listas a cada *worker*, quedando potencialmente algunso workers libres de trabajo (dependiendo de la cantidad de listas y de nodos presentes).

Cada *worker* que recibe 2 listas, irá recorriéndolas y reconstruyendo 1 solo subarreglo ordenado a partir de las listas. Una vez termina, reenviará este nuevo subarreglo al *master*, el cual tomará este subarreglo y lo enviará con otro más a otro *worker*. Este proceso de reconstrucción binaria continuará hasta que se haya reconstruido el arreglo original, el cual quedará ordenado.

ACLARACION PARA SECCION MEJORAS

Si bien la descomposición de dominio fue la elegida para desarrollar el programa, inicialmente se consideró la descomposición recursiva: el primer nodo tomaría el arreglo, encontraría un pivote, ordenaría elementos menores que el pivote a su izquierda y mayores a su derecha, y luego enviaría a un nuevo nodo la porción del arreglo restante a la izquierda del pivote, quedándose con la porción derecha para la siguiente iteración de quicksort. Tanto el nuevo nodo como el original, volverían a empezar este proceso, involucrando más y más nodos hasta llegar al límite de nodos con los que el programa se estuviese ejecutando. A partir de ese momento, cada nodo avanzaría con el quicksort secuencialmente hasta ordenar su porción, y empezarían a devolver “hacia arriba” (pensando en una jerarquía de árbol invertido) su porción de arreglo ya ordenada, quedando al final el arreglo ordenado.

Este primer enfoque nos gustó porque es fiel al espíritu del algoritmo quicksort, el cual es recursivo por naturaleza. Sin embargo, a la hora de plantear la sincronización de procesos (como por ejemplo, verificar si quedan nodos disponibles para ver si se puede enviar una porción de arreglo a otro nodo o no, y a cuál nodo enviarla), nos encontramos con muchos problemas a la hora de implementarlo. Además, el balanceo de carga de cada nodo era muy desequilibrado, ya que el primer nodo ejecutaba muchísimo procesamiento de ordenamiento, el segundo menos de la mitad que el primero, y así sucesivamente. Fueron éstas consideraciones lo que nos llevó a pensar en otro tipo de descomposición, llegando así a la descomposición de dominio que presentamos en la sección principal de este informe.

Sin embargo, si se pudiesen resolver estos problemas (por ejemplo, asignación dinámica de la carga según la cantidad de nodos activos, priorización de la partición de arreglos a aquellos que tengan porciones de arreglos más grandes, etc.), la descomposición recursiva podría funcionar, y tiene la gran ventaja de que arrojaría el resultado final (el arreglo ordenado) sin ningún tipo de procesamiento extra. Esto podría significar una importante mejora de rendimiento, siempre y cuando se puedan resolver los problemas de sincronización y balanceo de carga explicados.

Otro punto de mejora reside en la reconstrucción del arreglo: en la implementación presentada, algunos nodos podrían quedar ociosos durante este proceso, provocando una pérdida de eficiencia en el programa. Si este proceso de reconstrucción pudiese ser replanteado de tal manera que aprovechase todos los nodos disponibles, la eficiencia del programa aumentaría.

Quicksort Paralelo

[En el cuerpo del artículo se usa una sangría de primera línea de 1,27 cm con espaciado doble. En el estilo APA se usan hasta cinco niveles de título que se muestran en los párrafos siguientes. Tenga en cuenta que la palabra Introducción no ha de usarse como título inicial, ya que se supone que el artículo empieza con una introducción.]

# [Título 1]

[Los dos primeros niveles de título tienen supropio párrafo, como se muestra aquí. Los títulos 3, 4 y 5 son encabezados usados al principio del párrafo.]

## [Título 2]1

[Para agregar una tabla de contenido (TDC), aplique el estilo de título adecuado solo al texto del título al principio de un párrafo y se mostrará en la TDC. Para hacerlo, seleccione el texto del título Después, en la pestaña Inicio, en la galería de Estilos, haga clic en el estilo que necesite.]

### [Título 3].

[Incluya un puntoal final de un encabezado. Tenga en cuenta que, si corresponde, se pueden incluir párrafos consecutivos con sus propios títulos.]

#### [Título 4].

[Al usartítulos, no omita niveles. Si necesita usar un título 3, 4 o 5 sin texto antes del siguiente título, agregue un punto al final del título y, después, inicie un nuevo párrafo para el subtítulo y su texto.] (Apellidos, año)

##### [Título5].

[Al igual que el resto de las secciones del artículo, las referenciasempiezan en su propia página. La página de referencias que aparece después se crea con la característica Citas y bibliografía, que está disponible en la pestaña Referencias. En esta característica se incluye una opción de estilo que da formato a las referenciaspara la 6.ª edición de APA. También puede usar esta característica para agregar citas en el texto que estén vinculadas a la fuente, como las que se muestran al final de este párrafoy en el párrafo anterior. Para personalizar una cita, haga clic con el botón derecho y luego Editar cita.] (apellido, año)

Referencias

Apellidos, n. s. (Año). Título del artículo. *Título del diario*, Páginas desde - hasta.

Apellidos, n. s. (Año). *Título del libro.* Nombre de la ciudad: Nombre del editor.

Notas al pie

1[Agregue notas al pie, si corresponde, en su propia página después de las referencias. Para los requisitos de formato de APA, simplemente escriba sus propias referencias y notas al pie. Para dar formato a una referencia de nota al pie, seleccione el número y, después, en la Galería de estilos de la pestaña Inicio, haga clic en Referencia de nota al pie. En el cuerpo de una nota al pie, como en este ejemplo,se usa el estilo de texto Normal. (Nota: Si elimina esta nota al pie de ejemplo, no se olvide de eliminartambién su referencia en el texto. Está al final del párrafo Título 2 de ejemplo de la primera página del contenido del cuerpo de esta plantilla).]

Tablas

Tabla 1

[Título de tabla]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Encabezado de columna | Encabezado de columna | Encabezado de columna | Encabezado de columna | Encabezado de columna |
| Encabezado de fila | 123 | 123 | 123 | 123 |
| Encabezado de fila | 456 | 456 | 456 | 456 |
| Encabezado de fila | 789 | 789 | 789 | 789 |
| Encabezado de fila | 123 | 123 | 123 | 123 |
| Encabezado de fila | 456 | 456 | 456 | 456 |
| Encabezado de fila | 789 | 789 | 789 | 789 |

Nota: [Coloque todas las tablas del artículo en una sección de tablas, después de las referencias (y, si corresponde, de las notas al pie). Use una página nueva para cada tabla e incluya un número de tabla y un título de tablapara cada una, como se muestra en esta página. Todo el texto explicativo aparece en una nota de tabla después de la tabla, como en esta. Use el estilo de tabla o ilustración, disponible en la galería de estilos de la pestaña Inicio, para agregar el espaciado entre la tabla y la nota Las tablas en el formato de APA pueden usar un interlineado de una línea o de 1,5 líneas. Incluya un título para cada fila o columna, incluso si el contenido parece obvio. Se configuró un estilo de tabla predeterminado para esta plantilla que cumple con las normas del estilo APA. Para insertar una tabla, en la pestaña Insertar, haga clic en Tabla.]

Título de ilustraciones

Ilustración 1. [Incluya todas las ilustraciones en su propia sección, después de las referencias (y, si corresponde, de las notas al pie y las tablas). Incluya un título numerado para cada ilustración. Use el estilo de tabla o ilustración para agregar fácilmente espaciado entre la ilustración y el título.]

Para obtener más información sobre todos los elementos del formato de estilo APA, vea el Manual de estilo de la APA, 6.ª edición.