

¿Qué collector de usar?

Cual collector es realmente el mejor? La respuesta es que depende mucho del tipo de máquina y el área de memoria asignaciones, por ejemplo. En la siguiente tabla, se resumen algunos de los principales beneficios de cada collector mencionados anteriormente:

Colector	Al usar
Serial	Máquinas de un procesador y pequeños heaps
Parallel	máquinas y aplicaciones que requieren alto rendimiento multiprocesador
Concurrent	máquinas y aplicaciones procesador rápido con los SLA estrictos que requieren tiempos de respuesta rápidos
G1	máquinas de servidores multiprocesador con las aplicaciones que utilizan grandes heaps (> 6 GB) y en la necesidad de una latencia mínima GC

Como se puede ver, el colector de serie no es realmente una opción para las soluciones empresariales de hoy en día.

En caso de que su aplicación puede implementar en una máquina multiprocesador y requieren para completar el mayor número posible de transacciones en una ventana de tiempo, el colector paralelo es una buena opción. Este es el caso de las aplicaciones que realizan actividades de procesamiento por lotes, facturación y aplicaciones de nómina.

Tener procesadores rápidos y una aplicación que necesita para servir a cada solicitud por una cantidad de tiempo estricta es generalmente un caso para el colector concurrentes. El colector concurrente es particularmente adecuado para aplicaciones que tienen un relativamente gran conjunto de datos de larga vida, ya que puede recuperar objetos mayores sin una larga pausa. Este es generalmente el caso en las aplicaciones web donde se almacena una cantidad consistente de la memoria en el HttpSession.

El nuevo colector G1 es un útil, pero no es ampliamente utilizado, la sustitución de los colectores paralelos o concurrentes cuando uno más de las siguientes situaciones son notables para una aplicación:

- Más de la mitad del heap se ocupa de datos en tiempo real
- Hay largas GC o compactación pausas (más de la mitad de un segundo)
- La tasa de asignación de objetos o promoción varía mucho