

Caso de Estudio 3 – Análisis de Desempeño Sistema de Gestión Empresarial y Operativa de una Compañía Transportadora

Objetivos

- Evaluar las características de desempeño de la infraestructura computacional que soporta el despliegue de una aplicación.
- Simular condiciones de carga y configuraciones de software con el fin de analizar el comportamiento de una aplicación.

Descripción General

En esta tercera parte del caso se aborda la problemática del desempeño de la infraestructura computacional para el despliegue de la aplicación desarrollada en el caso 2. Exploraremos las relaciones entre el desempeño de la aplicación y las características de la infraestructura, ante diferentes niveles de carga.

Problemática

En el despliegue de una solución computacional es de vital importancia tener en cuenta la infraestructura subyacente con el fin de ofrecer un desempeño adecuado durante el ciclo de vida de la solución.

Actividades

1. **Preparación.** Vamos a simular escenarios para analizar el comportamiento de la aplicación y de los recursos de la máquina ante diferentes niveles de carga.

Use como punto de inicio el código que usted mismo desarrolló para el Caso 2.

- 1.1. Modifique el código para generar una aplicación multithread. Cada thread debe explorar una parte del espacio de búsqueda. Para simplificar el problema nos concentraremos solamente en el peor caso para una cadena de longitud 7.
- 1.2. Diseñe e implemente monitores de desempeño para medir los siguientes indicadores: (1) uso de CPU cada 5 minutos (explore el método `getSystemCpuLoad`, al final del enunciado, para medir uso de CPU) y (2) Tiempo de ejecución por thread.

2. **Identificación de la plataforma.** Defina la máquina en la que correrá la aplicación e identifique las siguientes características. Tenga en cuenta que su aplicación deberá correr en la misma máquina para todos los experimentos (si cambia de máquina, los resultados no serán comparables).

- Arquitectura (32 o 64 bits)
- Número de núcleos (*cores*)
- Velocidad del procesador
- Tamaño de la memoria RAM
- Espacio de memoria asignado a la JVM

3. **Comportamiento de la aplicación con diferentes estructuras de administración de la concurrencia.** Este escenario tiene como objetivo evaluar cambios en el comportamiento de la aplicación ante diferencias en la carga:

- Genere escenarios diferentes cambiando el número de threads y los algoritmos. Para el número de threads usaremos 1, 2, 4 y 8. Para los algoritmos de generación de código criptográfico de hash usaremos MD5, SHA256 y SHA512. Es decir, tenemos 12 escenarios (4 valores de threads X 3 algoritmos)
- Para cada uno de los escenarios mida los indicadores establecidos en el punto 1.2. Repita cada experimento 5 veces.
- Consolide los datos en un archivo Excel (una hoja por escenario) y evalúe la calidad de los datos en cada escenario.
- Cree las siguientes gráficas por algoritmo:
 - a. # threads vs. tiempo de ejecución por thread

- b. # threads vs. porcentaje de uso de la CPU
- Analice cuidadosamente las gráficas y determine cómo varía el comportamiento de la aplicación (en términos de los indicadores tiempo y CPU) con respecto a número de threads.

Entrega. Archivo zip o tar con:

- Código fuente de la aplicación.
- Archivo Excel con los datos recopilados.
- Informe con: (a) Descripción detallada de la implementación de los monitores, (b) identificación de la plataforma y (c) respuestas al punto 3 (todas las gráficas y la conclusión).
- Los datos y el informe **deben** estar en el subdirectorío docs del servidor. **Al comienzo del informe, escriba los nombres y carnés de los integrantes del grupo.** Si un integrante no aparece en el documento entregado, el grupo podrá informarlo posteriormente. Sin embargo, habrá una penalización: la calificación asignada será distribuida (dividida de forma equitativa) entre los integrantes del grupo.
- El trabajo se realiza en grupos de **2** personas. No debe haber consultas entre grupos. El grupo responde solidariamente por el contenido de todo el trabajo, y lo elabora conjuntamente (no es trabajo en grupo repartirse puntos o trabajos diferentes). Se puede solicitar una sustentación a cualquier miembro del grupo sobre cualquier parte del trabajo. Dicha sustentación será parte de la calificación de todos los miembros.
- El proyecto debe ser entregado por Sicua+ por uno solo de los integrantes del grupo.
- **La fecha límite de entrega es 20 de noviembre 2020, a las 11:50 p.m.**

Referencias

- *Java Threads (Third Edition)*. Oaks, S., & Wong, H. O'Reilly, 2004.
- *Presentaciones del curso (Análisis de desempeño)*.

Medida de CPU

<https://stackoverflow.com/questions/18489273/how-to-get-percentage-of-cpu-usage-of-os-from-java/21962037>

```
public double getSystemCpuLoad() throws Exception {
    MBeanServer mbs = ManagementFactory.getPlatformMBeanServer();
    ObjectName name = ObjectName.getInstance("java.lang:type=OperatingSystem");
    AttributeList list = mbs.getAttributes(name, new String[]{ "SystemCpuLoad" });

    if (list.isEmpty()) return Double.NaN;

    Attribute att = (Attribute)list.get(0);
    Double value = (Double)att.getValue();

    // usually takes a couple of seconds before we get real values
    if (value == -1.0) return Double.NaN;
    // returns a percentage value with 1 decimal point precision
    return ((int)(value * 1000) / 10.0);
}
```