

Trabajo Práctico 2

Consideraciones generales

El trabajo debe realizarse en una o más Notebooks de Python. En cada ejercicio deben estar explicados de forma detallada y justificados todos los pasos y procedimientos realizados para llegar a la resolución del mismo.

La entrega debe efectuarse por Campus, y sólo un miembro del grupo debe realizarla. En la misma se deben incluir tanto el código/desarrollo utilizado para resolver el trabajo práctico como las diapositivas utilizadas en la presentación.

Ejercicio 1

Un servidor recibe solicitudes las cuales son procesadas de una por vez en el orden de llegada (política FIFO).

Se determinó que en 10 milisegundos existe una probabilidad $p = 1/40$ que llegue una nueva solicitud y una probabilidad $q = 1/30$ que una solicitud termine de ser procesada y deje el sistema.

Se desea estudiar la cantidad de solicitudes en el servidor considerando tanto las que están en cola esperando ser procesadas como la solicitud que está siendo procesada.

- Determine la matriz de transición de estados explicando cómo se obtiene la misma.
- Simule, utilizando el generador de números al azar desarrollado en el tp 1, la evolución del sistema a lo largo de 1.500 segundos. Suponga que el servidor comienza sin estar procesando solicitudes.
- Realice un gráfico mostrando la cantidad de solicitudes en el servidor en cada instante de tiempo.
- Realice un histograma mostrando cuantas veces el sistema estuvo en cada estado.
- Determine el % de tiempo que el servidor se encuentra sin procesar solicitudes.

Ejercicio 2

Junto con el mismo grupo asignado para el ejercicio 3 se deben simular las 2 alternativas planteadas. Cada grupo debe elegir una arquitectura y simularla, luego entre ambos deben concluir la mejor opción a implementar.

Resolver planteando una simulación de tipo Montecarlo.

Se está diseñando un web service, el cual cada vez que es invocado consulta a una base de datos.

Se estima que el tiempo que transcurre entre cada llamada al servicio se puede modelar según una distribución exponencial con media $= 4$ segundos

Se debe decidir la arquitectura de base de datos a utilizar entre las dos siguientes:

- Utilizar 2 bases de datos distribuidas.

Con probabilidad $p = 0.7$ las solicitudes son atendidas por la base 1 y con probabilidad $q = 1 - p$ son atendidos por la base de datos 2.

El tiempo que demora cada base de datos en atender una solicitud sigue una distribución exponencial con medias, $1 = 0,7$ seg y $2 = 1$ seg respectivamente.

- Utilizar 1 base de datos central.

En este caso la demora en resolver una solicitud sigue una distribución exponencial con $= 0,8$ segundos

Simular para cada opción 100.000 solicitudes procesadas, determinando:

- El tiempo medio de espera entre que la solicitud llega y puede ser procesada (suponer que ninguna conexión cae por timeout).
- La fracción de las solicitudes que no esperaron para ser procesadas.

Ejercicio 3

En base al trabajo “SIMULATION OF LOAD BALANCING ALGORITHMS: A Comparative Study”, y utilizando Simpy, se pide simular un balanceador de carga siguiendo las siguientes técnicas:

- Difusa (propuesta en el trabajo)
- La técnica asignada a cada grupo, la cual se encuentra publicada en la planilla de grupos.
- Nueva técnica propuesta por el grupo

Cada grupo debe trabajar con otro equipo, donde cada uno implementará, salvo la estrategia Difusa, estrategias diferentes, para luego ser comparadas y analizar los resultados en conjunto.

Cada grupo debe proponer distintos parámetros para la simulación siendo estos:

- Tasa de arribos de requerimientos
- Tiempo de procesamiento por el balanceador, o tiempo en derivar el pedido
- Tiempo de procesamiento del requerimiento en el servidor destino
- Cantidad de servidores (se debe incluir la opción de 5 servidores evaluada en el trabajo)

Este ejercicio apunta a ser un trabajo de exploración, donde puedan modificar los parámetros buscando umbrales donde la decisión entre los distintos métodos no sea tan clara.

Suponer procesos de Poisson

Simular al menos 2 situaciones para cada técnica, comparando los resultados obtenidos.

Generar distribuciones para las variables tiempo en espera y tiempo de atención en los servidores, y medidas de resumen como media, mediana, y desvío estándar para poder comparar las alternativas.

Se debe elegir al menos 1 ejercicio de los 2 siguientes

Ejercicio 4

El siguiente sistema dinámico es conocido como “Modelo de la telaraña”.

Mediante el mismo se puede modelar la formación de precios de productos cuya oferta se establece en función del precio de mercado en el período de tiempo anterior.

$$\begin{cases} Q_t^{\text{demanda}} = a - bP_t \\ Q_t^{\text{oferta}} = dP_{t-1} - c \\ Q_t^{\text{demanda}} = Q_t^{\text{oferta}} = Q_t \end{cases} \quad \begin{array}{l} \text{Con:} \\ a=9 \\ b=1,1 \\ c=0,4 \\ d=1 \\ P_0=8 \end{array}$$

Se pide:

- Realizar un análisis de estabilidad de los puntos de equilibrio
- ¿Cómo afectan las condiciones iniciales al sistema? Justifique.
- Simular, utilizando el generador de números al azar implementado en el TP 1 el sistema para distintas condiciones iniciales e interprete los resultados.
- Graficar la variable precio en función del período (t) para 100 períodos.
- Graficar las trayectorias en el espacio de fases del sistema a través de 100 períodos para pares ordenados (Q_t , P_t)
- Simular, utilizando el generador de números al azar implementado en el TP 1, el sistema distintos valores de los parámetros a,b,c,d y P_0 y vuelva a realizar los ítems a,b,c,d, y e para cada juego de parámetros probados.

Ejercicio 5

Los números pseudo aleatorios son ampliamente utilizados en algoritmos de Machine Learning.

Para los grupos que se quedaron con ganas de utilizar el generador de números al azar implementado en el TP 1, en un caso real, les pedimos implementar el algoritmo KMeans, y aplicarlo al dataset entregado junto a este trabajo práctico.

Pasos del algoritmo:

- 1- Determinar la cantidad de cluster/grupos a utilizar. Para este caso utilizar 6.
- 2- Elegir los centroides a utilizar en cada grupo. Estos centroides se inicializan en coordenadas aleatorias.
- 3- Calcular la distancia de cada punto a cada centroide. Utilizar distancia euclídea. Agrupar cada observación con aquel centroide más próximo
- 4- Recalcular los centroides de cada grupo como la media de todos los puntos del cluster. Esto significa que se actualiza la posición del centroide para reflejar la nueva agrupación.
- 5- Repetir los pasos 3 y 4 hasta que alguno de los criterios de stop se cumplan:
 - Los centroides dejan de cambiar. Después de múltiples iteraciones, los centroides de cada clúster no cambian. Por lo que se asume que el algoritmo ha convergido.
 - Los puntos dejan de cambiar de clúster. Cuando se observa que no hay un intercambio de clústers se asume que el modelo está entrenado.
 - Límite de iteraciones. Se fija un número máximo de iteraciones que queremos que nuestro algoritmo ejecute antes de pararlo. Cuando llega a ese número se detiene el entrenamiento.

Evaluar el comportamiento del algoritmo utilizando distintos centroides iniciales (utilizando distintos números al azar generados).

Grafique cada observación y los grupos generados.