


Rendimiento y escalabilidad

Teoría de colas y aplicación



USAC - Análisis y diseño de sistemas 2 - 1er semestre 2016
Ing. Ricardo Morales

Teoría de colas

- ▶ Enfrentarse efectivamente al rendimiento y escalabilidad de un sistema de software depende no solo de las medidas cuantitativas, sino también del análisis cuantitativo basado en teorías probadas, como la teoría de colas
- ▶ La teoría de colas fue desarrollada para calcular y optimizar la eficiencia de cualquier sistema que alcanza sus objetivos al consumir múltiples recursos de forma óptima



Beneficios teoría de colas en rendimiento y escalabilidad

► Puede ayudar a:

- Entender los conceptos de rendimiento y escalabilidad de software mas formalmente
- Entender el rendimiento y escalabilidad de software mas científicamente
- Identificar cuellos de botella en el rendimiento y escalabilidad del software, mas eficientemente desde un punto de vista práctico
- Analizar las causas raíz de problemas de rendimiento y escalabilidad de software mas objetivamente
- Obtener con mejores guías para dimensionar productos de software



Caso teoría colas

- ▶ La pregunta, para construir un caso en el que se muestre qué problemas típicos puede ayudar a resolver la teoría de colas, en el contexto del rendimiento y escalabilidad de software, es:
- ▶ Para una tasa dada de arribo λ y una tasa dada de servicio μ , ¿cuántos servidores paralelos (m) son requeridos para que el sistema pueda operar bajo circunstancias estables?



Caso teoría colas (II)

- ▶ El problema puede ser respondido requiriendo que la intensidad de carga del sistema (ρ), definida como $\rho = \lambda / m\mu$ debe ser menor que 1 o $\lambda / m\mu$, donde m es el número mínimo de servidores paralelos requeridos para mantener una operación estable del sistema
- ▶ El encolamiento es un proceso dinámico. El número de solicitudes que arriban a una cola es estocástico o aleatorio en naturaleza, y el tiempo de servicio tampoco es constante



Aplicando teoría de la probabilidad a sistemas de colas

- ▶ Un sistema de colas tiene 2 procesos mutuamente acoplados: el proceso de arribo y el proceso de servicio. Estos 2 procesos son estocásticos o aleatorios hasta cierto punto:
 - ▶ Un proceso de arribos es caracterizado por el numero de arribos durante un período de tiempo dado y el tiempo inter arribos entre 2 arribos adyacentes. Ambos, el número de arribos y el intervalo entre arribos, son aleatorios, lo que hace al proceso de arribo un proceso estocástico en naturaleza
 - ▶ Ya que el proceso de arribo es estocástico, cualquier proceso subsecuente dirigido por el proceso de arribos es estocástico también. Esto justifica porque el proceso de servicio es estocástico también
 - ▶ Ya que el tiempo de servicio es aleatorio, el tiempo de respuesta y el desempeño de un sistema de colas que depende en el tiempo de servicio es aleatorio también



Procesos de Markov

- ▶ Cuando se aplica un proceso de Markov al sistema de colas mas simple, caracterizado por ciertos patrones de arribo y servicio, implica que:
 - ▶ El número de arribos sigue la distribución de Poisson
 - ▶ Los tiempos inter arribo siguen la distribución exponencial
 - ▶ Los tiempos de servicios siguen la distribución exponencial también



¿Qué significa cada distribución en el proceso de Markov?

- ▶ La distribución de Poisson es una distribución de probabilidad discreta que representa el proceso de arribo aleatorio, con la probabilidad de tener exactamente k eventos ocurriendo en un período de tiempo fijo con una tasa conocida de λ para la ocurrencia de esos eventos
 - ▶ En el contexto de rendimiento de software esos eventos pueden ser solicitudes de un usuario a un web server
- ▶ La distribución exponencial es usada para modelar el tiempo entre eventos independientes, tales como el tiempo inter arribos, el tiempo de servicio y el tiempo de respuesta



Notación de Kendall

- ▶ Basado en el tipo de proceso de arribo, el tiempo de servicio y otras características de un sistema de colas, Kendall desarrolló un conjunto de notaciones para definir diferentes tipos de colas de forma simbólica

Símbolo	Semántica
α	Tipo de distribución de probabilidad para proceso de arribo
σ	Tipo de distribución de probabilidad para tiempo de servicio
m	Numero de servidores en el centro de colas
β	Tamaño de buffer o capacidad de almacenamiento del centro de colas
N	Tamaño de población permitido, finito o infinito
Q	Tipo de política de servicio (FIFO, etc.)



Modelos de colas para sistemas de colas en red

- ▶ Las colas pueden estar encadenadas, para formar un sistema de colas de red, donde las salidas de una cola entran a la siguiente cola, estos sistemas pueden ser clasificados en abiertos y cerrados
 - ▶ Los sistemas de colas abiertos tienen una entrada externa y un destino final externo
 - ▶ Los sistemas de colas cerrados son completamente contenidos y los clientes circulan continuamente, sin dejar el sistema
- ▶ Un modelo de colas es un procedimiento acerca de como calcular algunas métricas de rendimiento del sistema de colas. Para el software, nos interesa el tiempo de respuesta y el desempeño



Modelos asociados a teoría de colas

- ▶ Se presentan a continuación los siguientes 3 temas asociados a la teoría de colas
 - ▶ Ley de Little, que muestra como las 3 mayores métricas de colas – desempeño, tiempo de respuesta y tamaño de la cola – se correlacionan entre si
 - ▶ Modelo abierto $M/M/1$ que asume que los clientes entran y salen del sistema. Es un modelo tratable analíticamente, por ello popular. Es lo mínimo que se debería entender de teoría de colas
 - ▶ Modelo cerrado $M/M/m/N/N$ que asume que hay un número limitado de clientes en el sistema



Ley de Little

- ▶ Indica que el número de clientes, esperando y recibiendo servicio, es igual al producto de desempeño y el tiempo de respuesta $N_i = X_i R_i$
- ▶ Donde X_i es el desempeño en el nodo i de la cola
- ▶ Ya que el tiempo de respuesta puede ser expresado como $R=W+S$, donde W representa el tiempo de espera y S el tiempo de servicio, la ecuación anterior se puede descomponer en:

$$N_{i_wait} = X_i W_i$$

$$N_{i_busy} = X_i S_i$$



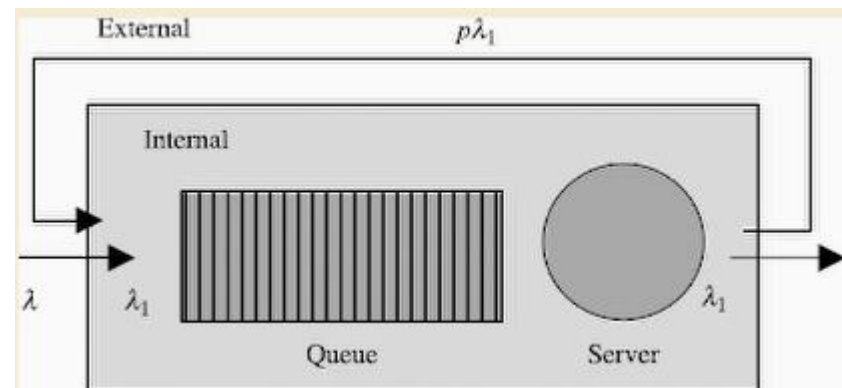
Modelo M/M/1 (abierto)

- ▶ Este modelo inicia con las siguientes 3 suposiciones:
 - ▶ Una tasa de arribo λ es conocida. Usualmente no es un problema, ya que la tasa de arribos dirigen la carga de un sistema
 - ▶ El sistema está corriendo bajo condiciones de equilibrio, que significa que el desempeño promedio X_0 es igual a la tasa de arribo λ . Esto es una forma de indicar que no se pierden transacciones en el sistema.
 - ▶ La demanda de servicio del recurso en cuestión es conocida y definida como $D_i = V_i S_i$, donde i representa al i ésimo nodo y V_i y S_i son el número de visitas al nodo y tiempo de servicio promedio requerido por visita, respectivamente



Modelo M/M/1 (II)

- ▶ Para derivar las fórmulas de este modelo, se considera un nodo de cola con retroalimentación, lo que significa que algunos clientes pueden regresar y visitar la cola mas de una vez
- ▶ En el contexto de software, retroalimentación significa múltiples visitas a un recurso requerido para completar una transacción a nivel del sistema



Modelo M/M/1 (III)

- ▶ El tiempo de respuesta está definido por

$$R_i = W_i + S_i = S_i / (1 - U_i)$$

- ▶ Donde W_i es el tiempo de espera, S_i es el tiempo de servicio y U_i es la utilización del recurso
- ▶ Sabiendo el desempeño del sistema X_0 y la demanda de servicio de i ésimo nodo D_i , podemos calcular la utilización de recurso U_i : $U_i = X_0 D_i$
- ▶ Con D_i y U_i conocidos, podemos calcular el tiempo de residencia R'_i en el nodo i : $R'_i = V_i R_i = D_i / (1 - U_i)$
- ▶ El tiempo promedio de respuesta total del sistema es: $R_0 = \sum_{i=1}^k R'_i$



Modelo M/M/1 (IV)

- ▶ En resumen, el modelo M/M/1 nos permite calcular el tiempo de respuesta R_0 para un sistema de software OLTP con una tasa λ dada de arribo de transacciones y servicio de demanda D_i previamente medido o tiempo de servicio S_i , sin retroalimentación
- ▶ La demanda de servicio es uno de los elementos mas básicos para aplicar el modelo para calcular el rendimiento de sistemas de software
- ▶ Tomar en cuenta que el desempeño del sistema es una métrica de rendimiento a ser calculada para batch jobs, no para sistemas OLTP
- ▶ Con los sistemas OLTP el desempeño es el mismo que la tasa de arribo bajo condiciones de equilibrio
- ▶ El desempeño para batch jobs es calculado usando el modelo cerrado M/M/m/N/N



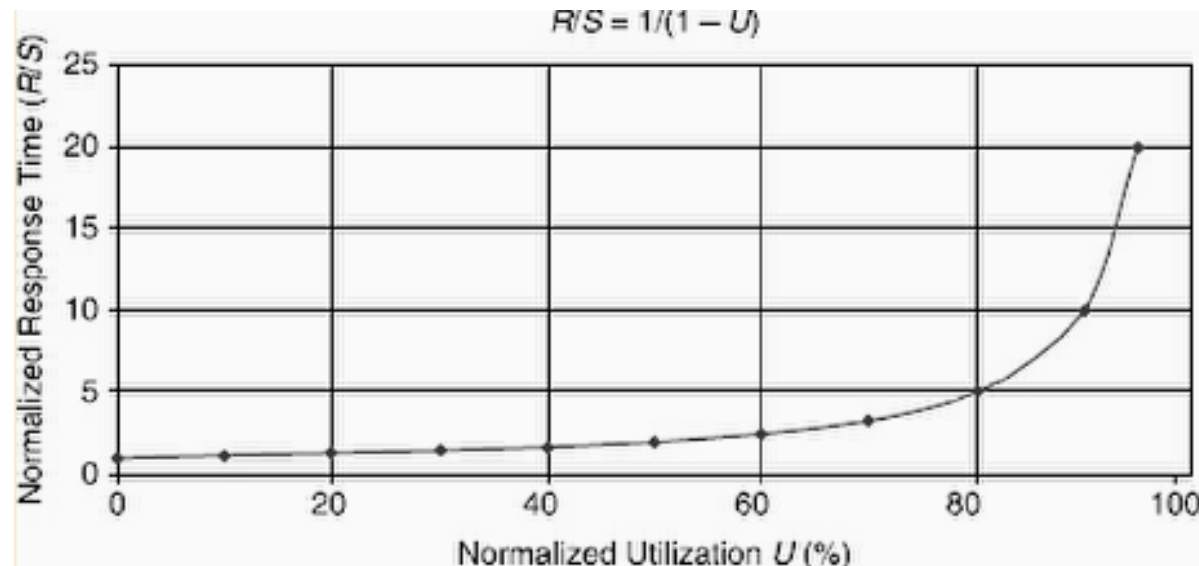
Sistemas de colas con retroalimentación y sin retroalimentación

Fórmula	Con retroalimentación	Sin retroalimentación
Desempeño del sistema	$X_0 = \lambda$	$X_0 = \lambda$
Desempeño local	$X_i = V_i X_0$	$X_i = X_0$
Demanda de servicio	$D_i = V_i S_i$	$D_i = S_i$
Utilización	$U_i = X_0 D_i$	$U_i = X_0 S_i$
Tiempo de residencia	$R'_i = V_i R_i = D_i (1 - U_i)$	$R'_i = R_i = S_i / (1 - U_i)$
Tiempo de respuesta del sistema	$R_0 = \sum_{i=1}^k R'_i$	$R_0 = \sum_{i=1}^k R_i$



Utilización, tiempo de servicio y tiempo de respuesta

- ▶ La relación entre estos 3 elementos es la siguiente ecuación $R = S / (1 - U)$
- ▶ La gráfica muestra que tan rápido puede crecer el tiempo de respuesta con un incremento en la utilización



-
- ▶ Asumiendo que el tiempo de servicio $S=1$ segundo, el tiempo de respuesta R crecerá a 2, 2.5, 3.3, 5, 10 y 20 segundos, si la utilización se incrementa a 50%, 60%, 70%, 80%, 90% y 95%
 - ▶ Esta es la razón por la cual se indica que la utilización de CPU para un sistema OLTP debe mantenerse bajo el 70%, para que el tiempo de respuesta del sistema no exceda mas de 3 veces el tiempo de servicio



-
- ▶ Es interesante notar que el rendimiento y escalabilidad de un sistema de software puede ser mejorado por las siguientes opciones:
 - ▶ Múltiples líneas de colas separadas paralelas, este escenario corresponde a “scale out” a múltiples servidores
 - ▶ Una sola cola multi servidor, este escenario corresponde a “scale up” con múltiples procesadores en un servidor

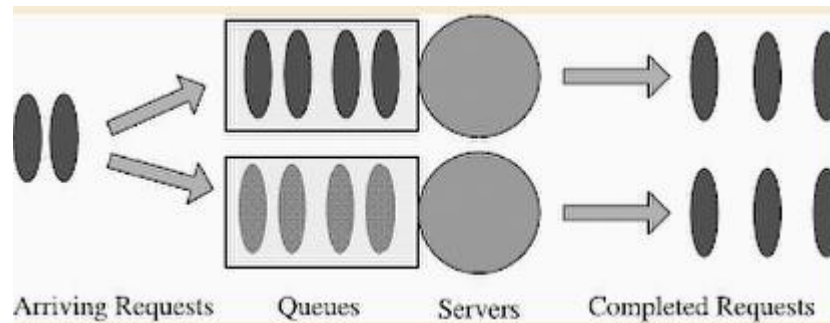


Colas paralelas múltiples vs cola única con múltiples servidores

- ▶ Para colas paralelas múltiples y partiendo de que

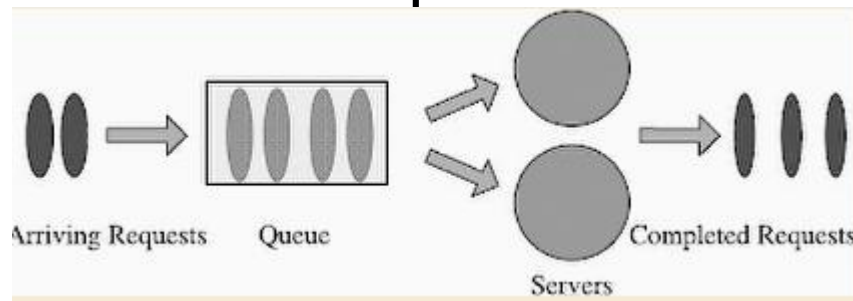
$$\rho = U/m:$$

$$R = S / (1 - \rho)$$

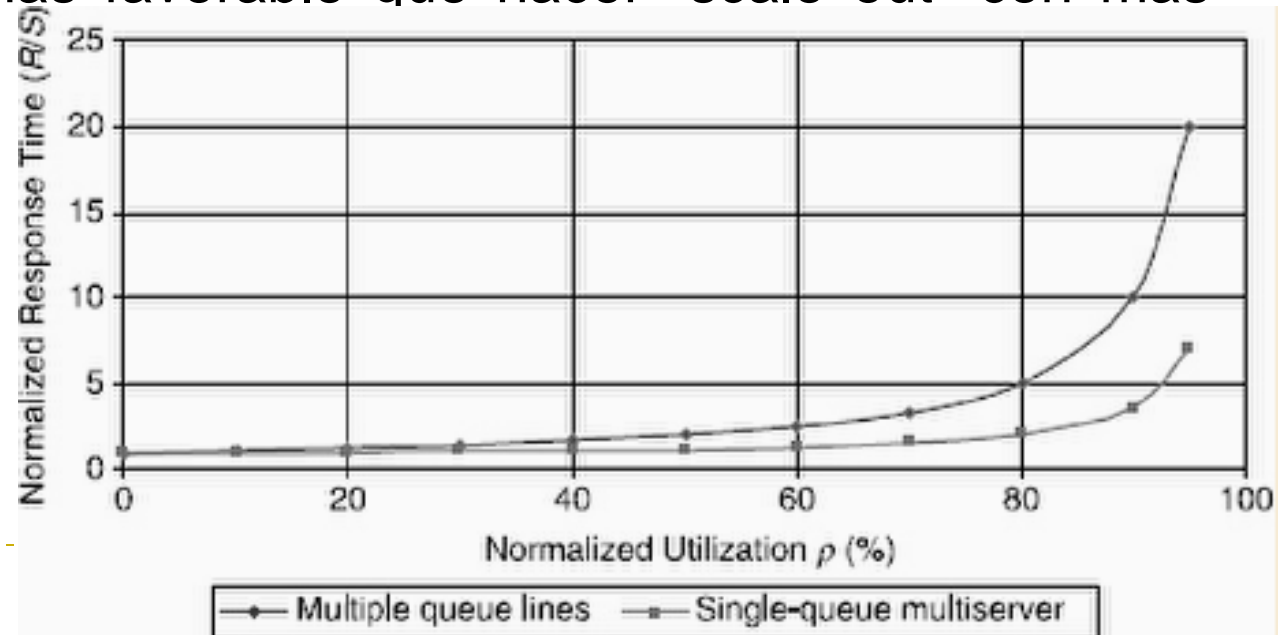


- ▶ Para una cola única con múltiples servidores

$$R = S / (1 - \rho^m)$$



- ▶ La siguiente gráfica compara ambos escenarios con $m=4$ para ambos escenarios
- ▶ Para la misma utilización de $\rho=50\%$, el tiempo de respuesta se degrada en un 100% en el escenario con múltiples colas, pero solo 14% en el escenario de una cola
- ▶ Este análisis soporta la práctica bien conocida de que hacer “scale up” con mas procesadores o procesadores mas poderosos es mas favorable que hacer “scale out” con mas servidores



Modelo cerrado M/M/m/N/N

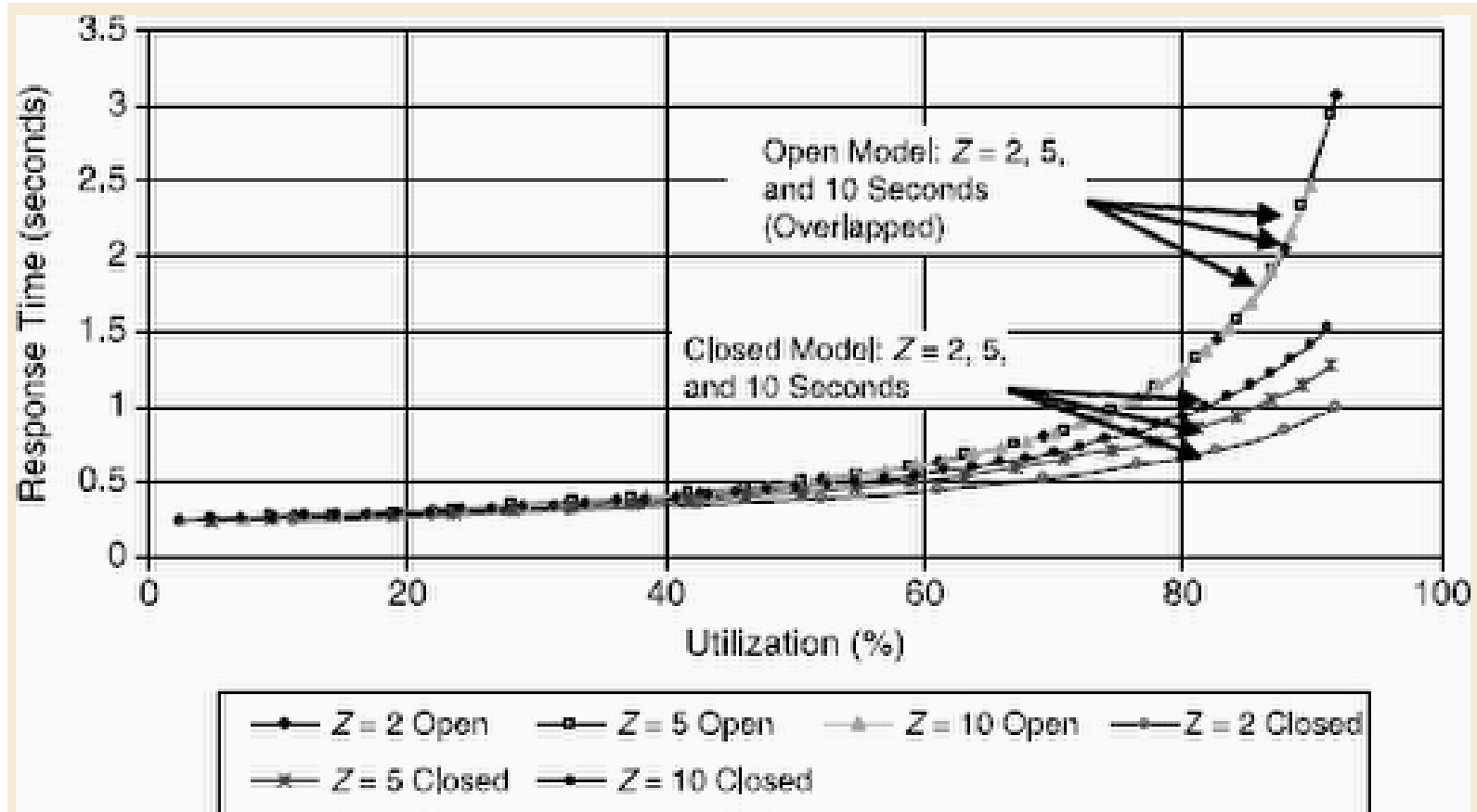
- ▶ Las siguientes ecuaciones describen este modelo:

$$R'_i[n] = D_i(1 + Q_i[n-1])$$
$$X[n] = \frac{n}{Z + \sum_{i=1}^m R'_i[n]}$$
$$Q_i[n] = X[n]R'_i[n]$$

- ▶ Donde n es la longitud de la cola a nivel de sistema o número de clientes en el sistema
- ▶ $R'_i[n]$ el tiempo de residencia en el nodo i
- ▶ $X[n]$ el desempeño del sistema
- ▶ $Q_i[n]$ la longitud de la cola en el nodo i
- ▶ Z el tiempo de pensar



Comparación entre modelos



Validez de los modelos abiertos

- ▶ Las fórmulas analíticas derivadas de los modelos abiertos son válidas bajo las siguientes condiciones:
 - ▶ El sistema no está cerca de la saturación o la utilización está bajo el 70%
 - ▶ La cola del sistema esta vacía o baja, lo que significa que el sistema no está corriendo cerca de la saturación y que puede tomar mas carga sin ver un impacto significativo en su rendimiento

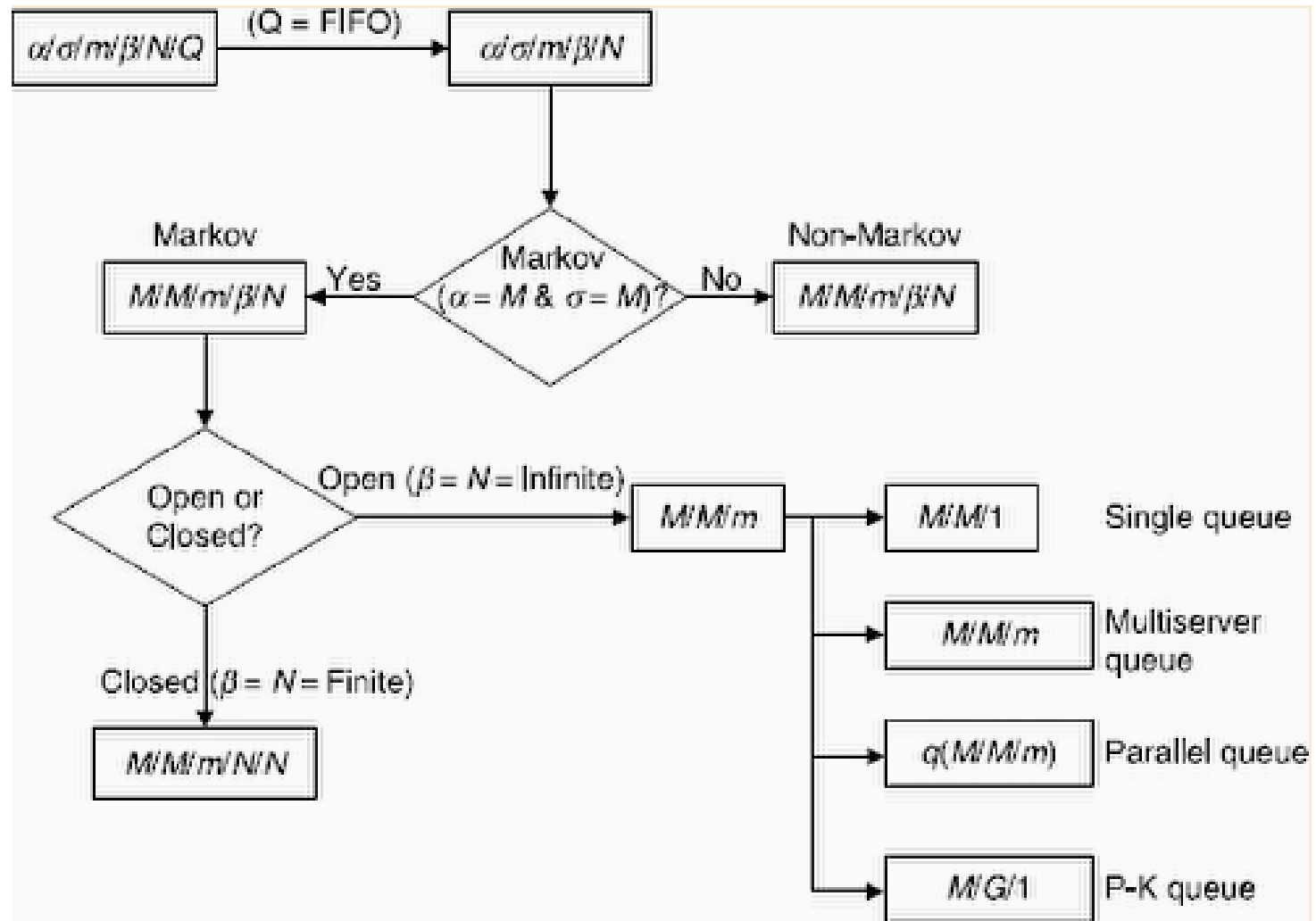


Cuellos de botella en rendimiento y escalabilidad en un sistema de software

- ▶ Típicamente, para un sistema que utiliza múltiples recursos, la demanda total de servicio es la suma de todas las demandas en los diferentes nodos de colas
- ▶ El recurso que contribuye mas a la demanda de servicio total, es definido como un cuello de botella
- ▶ Sin embargo, debido a que puede ser difícil medir la demanda de servicio, una alternativa es usar la utilización para identificar cuellos de botella
- ▶ Identificar los cuellos de botella del sistema es primer paso para optimizar y afinar el rendimiento escalabilidad del software



Genealogía de modelos de colas



Proceso ingeniería

- ▶ Un sistema de software raramente va a tener un buen rendimiento y escalabilidad sin pasar a través de un riguroso proceso de ingeniería de rendimiento y escalabilidad
- ▶ Dicho proceso de ingeniería generalmente incluye 2 actividades principales: optimización y afinamiento
- ▶ En el contexto de rendimiento y escalabilidad, la optimización se refiere a los esfuerzos para identificar y eliminar diseños e implementaciones internas ineficientes
- ▶ Afinamiento se refiere a los esfuerzos para establecer la configuración óptima para cada posible parámetro externo de configuración
- ▶ Ambos deben ser incorporados en los ciclos de desarrollo



Tipos de datos

- ▶ Los datos de prueba de rendimiento y escalabilidad generalmente se refieren a datos en las siguientes categorías
 - ▶ Métricas de rendimiento, tales como tiempo de respuesta y desempeño que califiquen el rendimiento del sistema bajo pruebas
 - ▶ Datos de utilización de recursos, obtenidos durante las pruebas a través de los diferentes recursos como CPU, disco, red y memoria
 - ▶ Datos de perfilamiento, tales como perfilamiento de apis y reportes estadísticos de base de datos

