Pag 152

Performance

Es el tiempo de respuesta y la habilidad de un sistema de software para satisfacer un requerimiento en un tiempo determinado.

En tiempos actuales el sistema que esperamos un nivel de rendimiento es el sistema web, y el tema principal es el uso de hilos de ejecución donde se diseña para evitar colisiones, se conoce que los eventos “llegan” para ser ejecutados en patrones predecibles o distribuciones estadísticas, los patrones más comunes son: periódicos, estocásticos o esporádicos.

Periódicos: llegan en intervalos regulares de tiempo

Estocásticos: los eventos llegan en un tipo de distribución probabilísticas.

Esporádicos: eventos llegan en un patrón que no es periódico ni estocásticos.

La respuesta del sistema a un evento (estimulo) puede ser medido así:

Latencia: el tiempo entre la llegada del estímulo y la respuesta a ese estimulo

Deadlines in processing: los tiempos asignados de ejecución.

Throughput of system: usualmente el número de transacciones que un sistema puede ejecutar en una unidad de tiempo.

Jitter of the response: la variación permitida en la latencia.

Escenario

|  |  |
| --- | --- |
| Fuente de estimulo | El estímulo (evento), puede ser externo o interno |
| Estimulo | El estímulo que llega en un posible patrón: periódico, estocástico, esporádico |
| Artefacto | El sistema o uno o más de sus componentes |
| Entorno | El sistema puede variar en los siguientes modos de operación: normal, emergente, sobrecarga, carga máxima. |
| Respuesta | El sistema que procesa eventos y puede cambiar el estado (ejemplo: de normal a sobrecarga). |
| Medida de la respuesta | La respuesta es el tiempo que lleva procesar eventos: latencia, deadline, jitter, throughput), miss rate. |

El tiempo de respuesta es influido mayormente por:

Processing time: consume recursos y toma tiempo,

Blocked time: una computación puede ser bloqueada por la contienda por algún recurso y el recurso no es disponible.

Dentro del blocked time

Contención por recursos: algunos recursos solo pueden ser usados por un solo cliente al

momento.

Disponibilidad de recursos: aunque no haya contención por recursos, la computación no

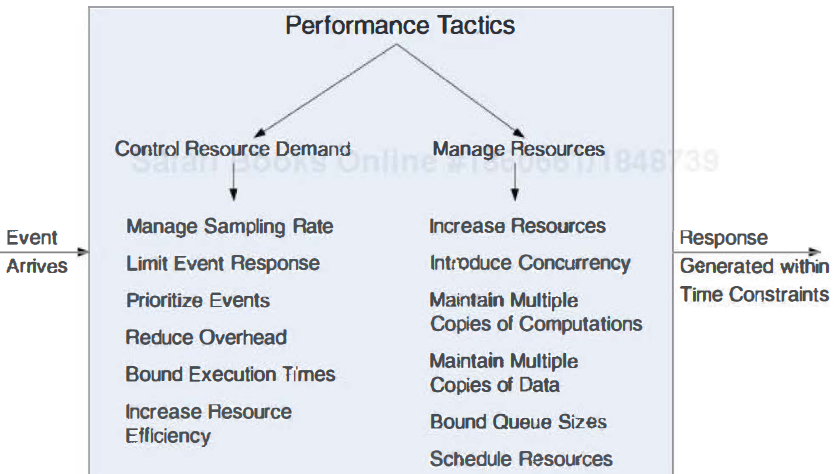
se realiza si el recurso no es disponible.

Dependencia en otra computación: una computación puede tener que esperar porque se

necesita estar sincronizado con otro proceso que no ha terminado y requiere

esperar.

De este modo, se definen categorías de tácticas



1 control resource demand

Táctica que opera en el lado de la demanda para producir una demanda menor que será procesada.

1 Control resource demand

Administrador de la demanda de procesos, se puede asi: limitando la tasa de ejecución, etc

1.1 manage sampling rate

Se puede reducir la demanda pero sacrificando un nivel de fidelidad, el diseño busca mantener niveles predecibles de latencia.

1.2 limit event response

Si los eventos son discretos y llegan muy rápido a ser procesados, los eventos son almacenado en una cola de espera (queue) hasta que sean procesados, un problema puede ser que necesites un queue de mayor tamaño

1.3 priorizar eventos

Si es posible, establecer un esquema de priorizar los eventos y ejecutarlos en base a esa prioridad.

1.4 reduce overhead

El uso de intermediarios como

clases abstractas e interfaces para facilitar la modificabilidad

separación de responsabilidades para facilitar modificabilidad

aumentan el trabajo de computo, es un clásico caso de compensación (trade-off) de modificabilidad-rendimiento.

Otro caso trade-off es el validación de información-rendimiento.

Una solución para reducir este trabajo excesivo (overhead) es co-localizar recursos, co-localizacion significa dirigir componentes que cooperan en el mismo procesador para evitar el tiempo de retraso en una red.

Otras soluciones para overhead son:

1 limpieza de recursos como tablas hash y mapeo de memoria virtual, la limpieza son reinicializar esos recursos

2 ejecutar hijos de ejecución dedicados y dividir en ellos las cargas de trabajo.

1.5 bound execution times

Definir tiempos máximos de ejecución para ciertos eventos, la implementación consiste en reducir el numero de iteraciones de algoritmos, el costo o tradeoffs es un computo menos preciso, se tiene que revisar periódicamente este trade-off para decidir mantenerlo o no.

1.6 incrementar la eficiencia de recursos

Mejorar la eficiencia de algoritmos en áreas críticas para reducir la latencia.

2 administracion de recursos

Aunque la demanda de recursos no es controlable, la administración es posible.

Opera en el lado de la respuesta para hacer los recursos más agiles para satisfacer la demanda.

Hay mecanismos posibles como mantener cierta información en memoria cache.

Otra tácticas de administración de recursos

2.1 Incrementar recursos

Agregar procesadores, memoria y redes mas veloces para reducir la latencia.

2.2 introducir concurrency

Hacer ejecuciones en paralelo y asi el tiempo de bloque es reducido.

2.3 mantener multiples copias de computaciones

Usar multiples servidores y un distribuidor de cargas de trabajo, también se puede preveer ciertas conductas de trabajo basado en las conductas anteriores.

2.4 bound queue sizes

Controla el tamaño de las colas de espera (queues) para maximizar los recursos, se requieren ciertas tácticas para maximizar los trade-offs.

2.5 schedule resources

En cualquier momento que haya una contienda por recursos se espera que se programe una lista de espera optimizada, se programan los CPUs, buffers y redes.

|  |  |
| --- | --- |
| Localización de responsabilidades | Determinar si las responsabilidades incluyen manejo de sobrecargas y áreas criticas por sobrecargas  Determinar responsabilidades para reconocer y procesar solicitudes adecuadamente incluye  -hilos de ejecución  -programar recursos compartidos como queues, buffers, caches  -asegurar el objetivo de rendimiento esperado |

|  |  |
| --- | --- |
| Modelo de coordinacion | Determinar los elementos que se deben coordinar (directa o indirectamente)  -soportar la concurrencia  -poder ejecutar eventos periódicos, estocásticos o esporádicos  -determinar estados apropiados de comunicación como: stateful, stateless, síncrono, asíncrono, entrega garantizada, throughput o latencia. |

|  |  |
| --- | --- |
| Modelo de datos | Determinar las partes del modelo de datos que puedan tener cargas pesadas.  Para las abstracciones de datos determinar  -si mantener multiples copias ayuda al rendimiento  -si la partición de datos ayuda al rendimiento  -si agregar recursos ayuda a reducir cuellos de botella de rendimiento |

|  |  |
| --- | --- |
| Mapeo entre elementos arquitecturales | Determinar si la co-localizacion de algunos componentes ayuda a reducir la sobrecarga (overhead)  Asegurar cuales componentes tienen alta demanda de recursos de procesamiento  Determinar si introducir concurrencia es factible  Determinar si la selección de hilos de ejecución críticos ayuda en reducir la sobrecarga |

|  |  |
| --- | --- |
| Administrar recursos | Determinar que recursos son críticos para el rendimiento  Asegurar que los recursos críticos sean monitoreados durante operación normal y operación con sobrecarga  Otras tareas  -identificar elementos que requieran vigilancia y administracion  -usar modelos de hilos de ejecución y procesos  -priorizacion de recursos y el acceso a los recursos  -programacion de estrategias de bloqueo.  -Agregar recursos adicionales |

|  |  |
| --- | --- |
| Binding time | Para cada elemento que va a ser enlazado determinar  -el tiempo necesario de enlace  -considerar el overhead de usar un método de enlace tardío  Asegurar que los 2 puntos anteriores no reduzcan el rendimiento |

|  |  |
| --- | --- |
| Selección de tecnologia | La tecnología seleccionada alcanzará los objetivos de rendimiento en tiempo ?  Se conoce las características la tecnología que soporta un mejor rendimiento ?  La tecnología seleccionada permite lo siguiente  - implementación de programación de queues de espera  -asignar prioridades  -permite implementar políticas de reducción de demandas  -administrar cargas de trabajo a los CPUs  -otros parámetros relacionados con rendimiento |