Artículo
Bernardo Linarez · 4 ago, 2020 Lectura de 9 min

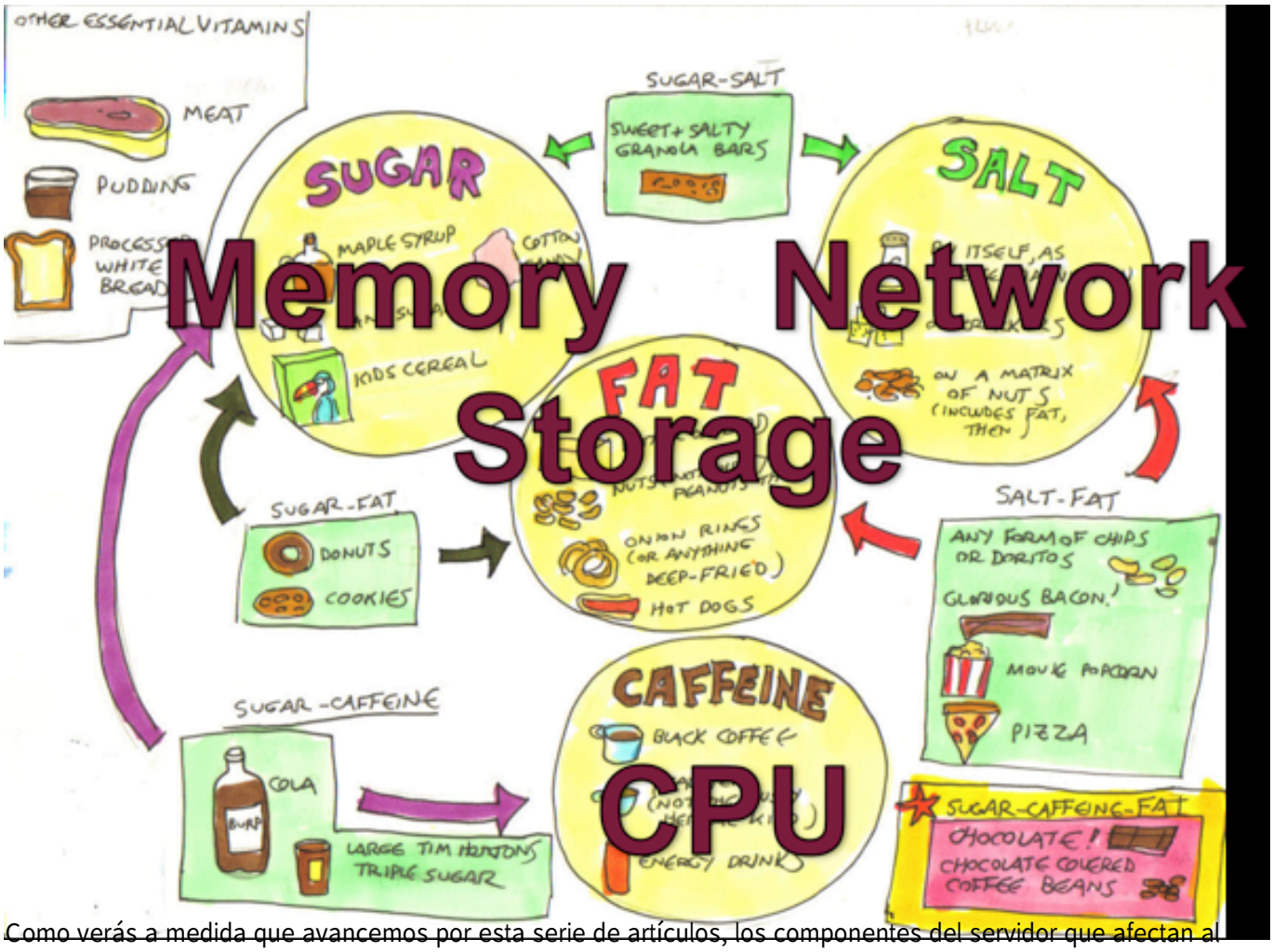
Plataformas de datos de InterSystems y su rendimiento - Parte 2

En la última publicación programamos recogidas de métricas de rendimiento usando pButtons, a lo largo de 24 horas. En esta publicación, analizaremos algunas de esas métricas clave que se están recogiendo y cómo se relacionan con el hardware del sistema subyacente. También empezaremos a explorar la relación entre las métricas de Caché (o de cualquiera de las plataformas de datos de InterSystems) y las métricas del sistema. Veremos también cómo usar estas métricas para entender el pulso diario de tu sistema y diagnosticar problemas de rendimiento.

Aquí puedes ver un listado con otros artículos de esta serie >>

Editado en octubre de 2016...
Añado un ejemplo de script para extraer datos de pButtons a un archivo .csv
Editado en marzo de 2018...
Las imágenes habían desaparecido, las volví a añadir.

Grupos alimenticios de hardware



Como verás a medida que avancemos por esta serie de artículos, los componentes del servidor que afectan al rendimiento pueden categorizarse como:

- CPU
- Memoria
- Almacenamiento de entrada y salida
- Red de entrada y salida

Si alguno de estos componentes está bajo una carga excesiva, el rendimiento del sistema y la experiencia de usuario seguramente se vean disminuidas. Estos componentes también están todos relacionados entre sí, los cambios en un componente pueden afectar a otro, a veces con consecuencias imprevistas. He visto un caso en el que corregir un cuello de botella de E/S en una matriz de almacenamiento causó que el uso de CPU saltara a 100%. Esto resultó en una experiencia de usuario aún peor, ya que el sistema de pronto quedó libre para hacer más trabajo, pero no tenía los recursos de CPU para atender el aumento en la actividad de usuario y el rendimiento.

También veremos como la actividad del sistema Caché tiene un impacto directo sobre los componentes del servidor. Si los recursos de E/S de almacenamiento son limitados, un cambio positivo que se puede hacer es aumentar la memoria del sistema y aumentar la memoria para **búferes globales de Caché**, lo que a su vez puede reducir la **E/S de lectura de almacenamiento del sistema** (¡pero quizás aumentar la CPU!).

Una de las métricas más obvias del sistema para monitorizar frecuentemente o para revisar cuándo reportan problemas los usuarios es el uso de CPU. Debemos mirar *top* o *nmon* en Linux o AIX, o *Windows Performance Monitor*. Como la mayoría de los administradores de sistemas revisan frecuentemente los datos de la CPU, especialmente si se presentan de forma gráfica, un rápido vistazo nos da una buena idea de la salud actual del sistema: qué es normal y qué es un pico repentino de actividad que podría ser anormal o indicativo de un problema. En este artículo revisaremos brevemente las métricas de CPU, y nos centraremos en las métricas de Caché. Comenzaremos por ver los datos de *mgstat* y entenderemos que mirar los datos de forma gráfica nos permite obtener, de un vistazo, una idea de la salud del sistema.

Introducción a mgstat

mgstat es uno de los comandos de Caché incluidos y ejecutados en pButtons. mgstat es una gran herramienta para recoger métricas básicas de rendimiento para ayudarte a entender la salud de tus sistemas. Observaremos los datos de mgstat recogidos de un pButtons de 24 horas, pero si quieres capturar datos fuera de mgstat de pButtons, también puede ejecutarse a demanda de forma interactiva o como tarea en segundo plano desde el terminal de Caché.

Para ejecutar mgstat a demanda desde el namespace %SYS, el formato general es.

```
do mgstat(sample_time,number_of_samples,"/file_path/file.csv",page_length)
```

Por ejemplo, para ejecutar una tarea en segundo plano para una ejecución de una hora con período de muestreo de 5 segundos y salida a un archivo csv.

```
job ^mgstat(5,720,"/data/mgstat_todays_date_and_time.csv")
```

Por ejemplo, para mostrar en pantalla pero omitiendo algunas columnas, usa la entrada dsp132. Te dejaré como tarea revisar la salida para entender la diferencia.

```
do dsp132^mgstat(5,720,"",60)
```

Puedes encontrar información detallada sobre las columnas de mgstat en la *Guía de monitarización de Caché* en la documentación de InterSystems:

InterSystems online documentation

Analizando los datos de mgstat

pButtons fue diseñado para compilarse en un único archivo HTML que se pueda navegar y empaquetar fácilmente para enviar a los especialistas del Centro de Soporte Internacional (WRC), que podrán diagnosticar problemas de rendimiento. Sin embargo, cuando ejecutas pButtons por tu cuenta y quieres mostrar los datos gráficamente, puedes volver a separarlos en un archivo csv para convertirlo en un gráfico, por ejemplo con Excel, usando un script de línea de comandos o simplemente cortando y pegando.

En este artículo, profundizaremos solo en algunas de las métricas de mgstat para mostrar como, incluso con un rápido vistazo, podemos hacernos una buena idea sobre si el sistema está funcionando bien, o si hay problemas reales o potenciales que afectarán a la experiencia del usuario.

Glorefs y CPU

El siguiente gráfico muestra el uso de CPU del servidor de la base de datos en un sitio donde se ejecuta una aplicación hospitalaria con una tasa muy elevada de transacciones. Presta atención al pico de actividad de la mañana, cuando hay muchos pacientes externos, con una caída a la hora del almuerzo y un descenso mantenido durante la tarde y labnoche. En este caso, los datos provienen de Windows Performance Monitor *(Total)\% Processor Time* - la forma del gráfico encaja con el perfil de un día laborable. No hay picos ni "valles" inusuales, lo que es normal para este sitio. Al hacer el mismo análisis para tu sitio, podrás empezar a tener una base de referencia de la actividad "normal". Un pico muy grande, especialmente si es prolongado, puede ser indicativo de un problema. En una publicación futura me enfocaré en el CPU.

Como referencia, este servidor de base de datos es un Dell R720 con dos procesadores E5-2670 de 8 núcleos, el servidor tiene 128 GB de memoria y 48 GB de búferes globales.

El siguiente gráfico muestra más datos de mgstat: Glorefs (referencias globales) o accesos a la base de datos para el mismo día que la gráfica de uso de CPU. Glorefs indica la cantidad de trabajo que se hace para procesar la carga de trabajo actual. Si bien las referencias globales consumen tiempo de CPU, no siempre consumen otros recursos del sistema, como lecturas físicas, por la forma en que Caché usa el recurso compartido de búfer de memoria global.

Como es típico en aplicaciones de Caché, hay una correlación muy fuerte entre Glorefs y uso de CPU.

> Otra forma de analizar estos datos de CPU y gloref es decir que *reducir las glorefs reducirá el uso de CPU*, lo que permitirá el uso de servidores con menos núcleos o escalar más sobre los sistemas existentes. Puede haber formas de reducir las referencias globales si hacemos que la aplicación sea más eficiente. Volveremos a este concepto en otros artículo posteriores.

PhyRds y Rdratio

La forma del gráfico obtenido con los datos de mgstat *PhyRds* (Physical Reads, lecturas físicas) y *Rdratio* (Read ratio, tasa de lectura) también puede aportar información sobre qué esperar del rendimiento del sistema y ayudar con la planificación de capacidad. En futuras publicaciones hablaré con más detalle sobre el almacenamiento de entrada y salida para Caché.

PhyRds son simplemente IOPS de lectura física desde el disco a las bases de datos de Caché. Deberías ver los mismos valores reflejados en las métricas del sistema operativo para discos lógicos y físicos. Recuerda al mirar las IOPS del sistema operativo que también podrían estar mostrando IOPS provenientes de aplicaciones que no son de Caché. Dimensionar el almacenamiento y no tener en cuenta las IOPS esperadas es una receta para el desastre. Necesitas conocer las IOPS que tu sistema realiza en horas pico para realizar una adecuada planificación de capacidad. El siguiente gráfico muestra las *PhyRds* entre la media noche y las 15:30.

Observa el gran salto en lecturas entre las 05:30 y las 10:00. Hay otros picos menores a las 11:00 y justo antes de las 14:00. ¿Qué crees que los causa? ¿Ves este tipo de picos en tus servidores?

Rdratio es un poco más interesante: es la tasa de lectura de bloques lógicos por lectura de bloque físico. Así que es una tasa de cuántas lecturas son de búferes globales (lógicos) desde la memoria y cuántas son del disco, lo que es órdenes de magnitud más lento. Un *Rdratio* alto es algo bueno. Que caiga cerca de cero por largos períodos no es bueno.

Observa que al mismo tiempo que hay una alta cantidad de lecturas, *Rdratio* baja a cerca de cero. En este sitio me pidieron investigar cuando el departamento de TI comenzó a recibir llamadas de usuarios que se quejaban de que el sistema se volvía lento por largos períodos. Esto había estado sucediendo al parecer de forma aleatorio desde hacía semanas, y me pidieron analizar el sistema.

> ***Como pButtons había sido programado para ejecutarse diariamente durante 24 horas, fue relativamente sencillo analizar varias semanas de datos para empezar a ver un patrón de PhyRds elevados y bajos Rdratio, correlacionados con las llamadas al soporte técnico.***

Tras un mayor análisis, pude rastrear la causa hasta un trabajador por turnos nuevo, que estaba ejecutando varios informes metiendo parámetros "erróneos" junto con consultas mal redactadas, sin los índices adecuados, lo que causaba la elevada cantidad de lecturas a la base de datos. Este era el origen de las ralentizaciones aparentemente aleatorias. Como estos informes que abarcan mucho tiempo leen datos hacia los búferes globales, el resultado es que los datos de usuario interactivos se recuperan del almacenamiento físico en vez de hacerse desde la memoria. Esto ejerce una gran carga sobre el almacenamiento para atender las lecturas.

Supervisar *PhyRds* y *Rdratio* te dará una idea del estado de funcionamiento de tus sistemas y quizás te permita encontrar informes o consultas mal realizadas. Podría existir una razón válida para tener un *PhyRds* alto -- quizás sea necesario ejecutar un informe a cierta hora. Con los servidores y sistemas operativos modernos de 64 bits y su gran capacidad de memoria física, deberías poder minimizar el *PhyRds* en tus sistemas de producción.

> Si observas un *PhyRds* alto en tu sistema, hay varias estrategias que puedes evaluar:
> - Aumentar la cantidad de búferes (globales) de base de datos (y la memoria del sistema) para mejorar el desempeño.
> - Mover fuera del horario de oficina la creación de informes o extractos que requieran un largo tiempo.
> - Los informes de solo lectura que requieran mucho tiempo, las tareas por lotes y las extracciones de datos pueden ejecutarse en un servidor de shadowing independiente o mirror asíncrono, lo que minimizará el impacto sobre los usuarios interactivos y quitará parte de la carga sobre los recursos del sistema, como CPU e IOPS.

Un *PhyRds* que normalmente sea bajo es algo bueno, y debe ser nuestro objetivo al dimensionar sistemas. Sin embargo, si tienes un *PhyRds* bajo y los usuarios se quejan del rendimiento, hay otras cosas que se pueden revisar para asegurarse de que el almacenamiento no sea un cuello de botella - la cantidad de lecturas podría ser baja debido a que el sistema ya no puede dar más servicio. Analizaremos el almacenamiento con mayor profundidad en un artículo posterior.

Resumen

En esta publicación hemos visto que mostrar con gráficos las métricas recogidas en pButtons puede darnos una idea del estado del sistema de un vistazo. En publicaciones futuras, analizaré en mayor profundidad la relación entre el sistema y las métricas de Caché, y cómo se pueden usar para planificar para el futuro.

#Administración del sistema #Rendimiento #Caché #InterSystems IRIS #InterSystems IRIS for Health

URL de fuente:https://es.community.intersystems.com/post/plataformas-de-datos-de-intersystems-y-su-rendimiento-parte-2