

Artículo

[Ricardo Paiva](#) · Abr 9, 2021 Lectura de 6 min

[Open Exchange](#)

Analizar el rendimiento de entrada/salida de la base de datos en disco

Objetivo

Esta herramienta se usa para generar una Entrada/Salida (E/S) de lectura aleatoria desde dentro de la base de datos. La finalidad de esta herramienta es llevar la mayor cantidad de tareas posibles para conseguir las IOPS objetivo y asegurar que se mantienen tiempos de respuesta de disco aceptables. Los resultados recopilados de las pruebas de E/S variarán de configuración a configuración, de acuerdo con el subsistema de E/S. Antes de ejecutar estas pruebas, asegúrate de que el sistema operativo y la monitorización del nivel de almacenamiento estén configurados para capturar métricas de desempeño de E/S para su posterior análisis.

Metodología

Empieza con una cantidad reducida de procesos y 10 000 iteraciones por proceso. Usa 100 000 iteraciones por proceso para matrices de almacenamiento basadas íntegramente en tecnología flash. A continuación, aumenta el número de procesos, por ejemplo, empieza con 10 trabajos y aumenta en 10, 20, 40, etc. Sigue ejecutando pruebas individuales hasta que el tiempo de respuesta esté por encima de 10 ms de forma constante, o las IOPS calculadas ya no aumenten de forma lineal.

Como guía, los siguientes tiempos de respuesta para lecturas aleatorias de bases de datos de 8KB y 64KB (sin cache) normalmente son aceptables para las matrices íntegramente flash:

- Promedio \leq 2ms
- No debe exceder \leq 5ms

La herramienta necesita una base de datos IRIS.DAT pre-expandida vacía, de un tamaño de al menos el doble del espacio de memoria en el servidor y de al menos cuatro veces el tamaño de la cache del controlador de almacenamiento. La base de datos debe ser mayor que la memoria, para asegurar que las lecturas no se guarden en la cache del sistema de archivos.

La herramienta usa el comando VIEW de ObjectScript, que lee bloques de bases de datos en memoria, así que si no estás logrando los resultados esperados, puede que todos los bloques de la base de datos ya estén en memoria.

Especificaciones y objetivos

Completa la siguiente tabla con tus objetivos y especificaciones de entorno:

Especificación

Almacenamiento

Servidor físico

Especificación

Máquina virtual

Tamaño de la base de datos

Memoria compartida

IOPS objetivo

Tiempo de respuesta objetivo

Instalación

Descarga la herramienta PerfTools.RanRead.xml de GitHub [aquí](#).

Importa PerfTools.RanRead.xml en el namespace USER.

```
USER> do $system.OBJ.Load("/tmp/PerfTools.RanRead.xml","ckf")
```

Ejecuta el método Help para ver todos los puntos de acceso. Todos los comandos se ejecutan en %SYS.

```
USER> do ##class(PerfTools.RanRead).Help()
```

```
InterSystems Random Read IO Performance Tool
```

```
-----  
do ##class(PerfTools.RanRead).Setup(Directory,DatabaseName,SizeGB,LogLevel)  
    - Creates database and namespace with the same name. The log level must be in the  
    range of 0 to 3, where 0 is "none" and 3 is "verbose".  
do ##class(PerfTools.RanRead).Run(Directory,Processes,Iterations)  
    - Run the random read IO test.  
do ##class(PerfTools.RanRead).Stop()  
    - Terminates all background jobs.  
do ##class(PerfTools.RanRead).Reset()  
    - Deletes all random read history stored in ^PerfTools.RanRead*  
do ##class(PerfTools.RanRead).Export(directory)  
    - Exports a summary of all random read test history to tab delimited text file.
```

Configuración

Crea una base de datos vacía (preexpandida) llamada ZRANREAD de aproximadamente el doble de tamaño de la memoria del servidor físico a probar. Asegúrate de que la base de datos vacía tenga al menos cuatro veces el tamaño de la cache del controlador de almacenamiento. Para crear automáticamente un namespace y una base de datos, puedes hacerlo manualmente o usar el siguiente método.

```
USER> do ##class(PerfTools.RanRead).Setup("/usr/iris/db/zranread","ZRANREAD",100,1)

Creating 100GB database in /usr/iris/db/zranread/
Database created in /usr/iris/db/zranread/
Run %Installer Manifest...
2016-05-23 13:33:59 0 PerfTools.RanRead: Installation starting at 2016-05-23 13:33:59
, LogLevel=1
2016-05-23 13:33:59 1 CreateDatabase: Creating database ZRANREAD in /usr/iris/db/zran
read// with resource
2016-05-23 13:33:59 1 CreateNamespace: Creating namespace ZRANREAD using ZRANREAD/ZRA
NREAD
2016-05-23 13:33:59 1 ActivateConfiguration: Activating Configuration
2016-05-23 13:34:00 1 EnableEnsemble: Enabling ZRANREAD
2016-05-23 13:34:00 1 ActivateConfiguration: Activating Configuration
2016-05-23 13:34:00 0 PerfTools.RanRead: Installation succeeded at 2016-05-23 13:34:0
0
2016-05-23 13:34:00 0 %Installer: Elapsed time 1.066633s
Database /usr/iris/db/zranread/ ready for testing.
do ##class(PerfTools.RanRead).Run(directory,processes,iterations) e.g.
do ##class(PerfTools.RanRead).Run("/usr/iris/db/zranread/",1,10000)
```

Ejecución

Ejecuta el método Run aumentando la cantidad de procesos y tomando nota, cada vez, del tiempo de respuesta. Si las pruebas son demasiado rápidas o los resultados no son los esperados, aumenta entonces el número de iteraciones a 10 000.

```
USER> do ##class(PerfTools.RanRead).Run("/usr/iris/db/zranread",20,10000)

InterSystems Random Read IO Performance Tool
-----
Starting 20 jobs in the background.
To terminate jobs run: do ##class(PerfTools.RanRead).Stop()
Waiting for jobs to finish.....
Random read background jobs finished.
20 processes (1000 iterations) average response time = 7.18ms
Calculated IOPS = 2787
```

Resultados

Los resultados para cada ejecución se guardan en USER en la tabla SQL PerfTools.RanRead. Ejecuta la siguiente consulta SQL para ver un resumen de resultados.

```
SELECT RunDate,RunTime,Database,Iterations,Processes,
       {fn ROUND(AVG(ResponseTime),2)} As ResponseTime,
       {fn ROUND(AVG(IOPS),0)} As IOPS
FROM PerfTools.RanRead
GROUP BY Batch
```

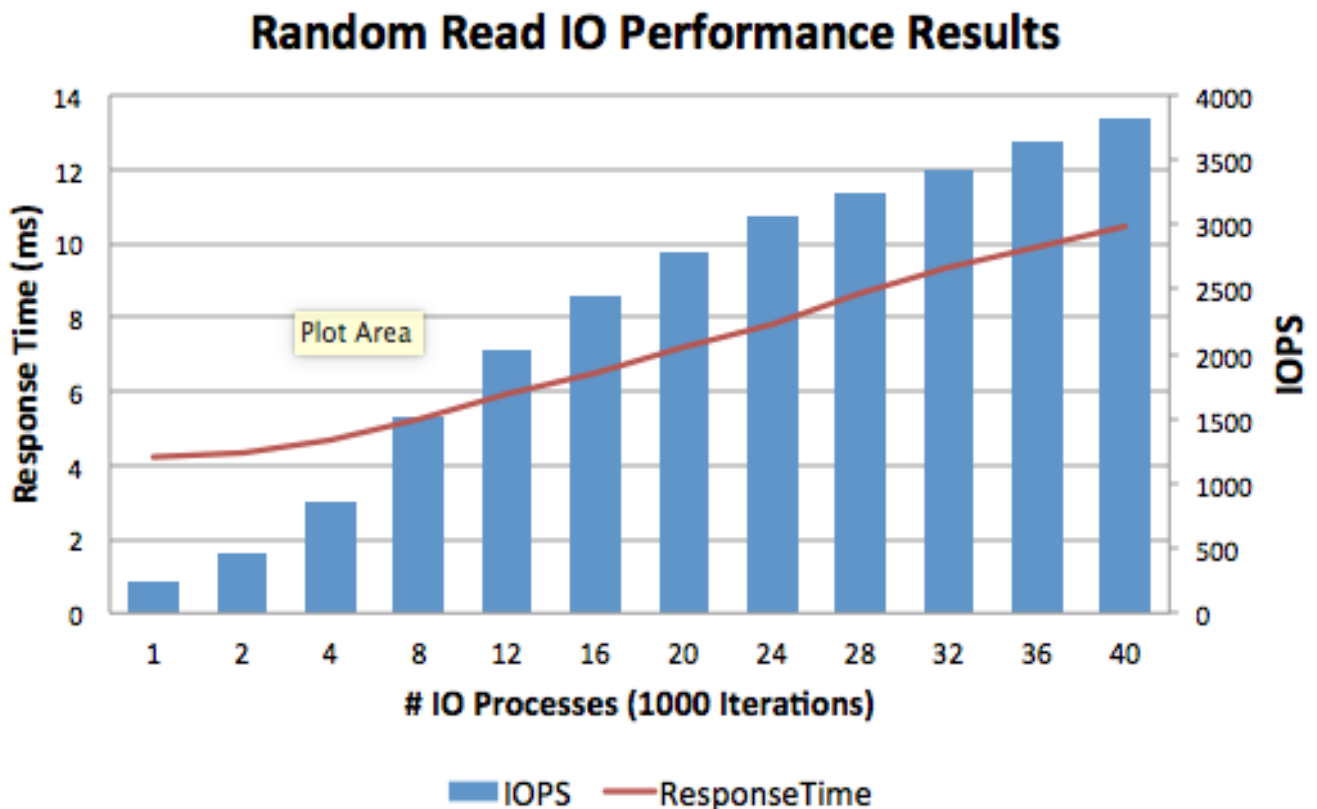
Para exportar el conjunto de resultados a un archivo de texto delimitado por tabulaciones, ejecuta lo siguiente:

```
USER> do ##class(PerfTools.RanRead).Export("/usr/iris/db/zranread/")
```

```
Exporting summary of all random read statistics to /usr/iris/db/zranread/PerfToolsRanRead_20160523-1408.txt
Done.
```

Análisis

Abre en Excel el archivo de texto exportado. Luego, copia y pega los datos en la hoja de cálculo PerfToolsRandomRead_Analysis_Template.xlsx para hacer un gráfico.



La hoja de cálculo de muestra se puede descargar desde GitHub [aquí](#).

Limpieza

Cuando se terminen de ejecutar las pruebas, elimina el historial ejecutando:

```
%SYS> do ##class(PerfTools.RanRead).Reset()
```

[#Bases de datos #Rendimiento #InterSystems IRIS](#)
[Compruebe la aplicación relacionada en InterSystems Open Exchange](#)

de-datos-en-disco