Article
David Loveluck · Nov 20, 2017 5m read

APM - Using the IRIS or Caché History Monitor

APM normally focuses on the activity of the application but gathering information about system usage gives you important background information that helps understand and manage the performance of your application so I am including the IRIS History Monitor in this series.

In this article I will briefly describe how you start the IRIS or Caché History Monitor to build a record of the system level activity to go with the application activity and performance information you gather. I will also give examples of SQL to access the information.

What is the IRIS or Caché History Monitor?

The History Monitor is available on IRIS or on earlier Caché and Ensemble platforms. It is an extension of the IRIS or Caché System Monitor. It keeps a persistent record of the metrics relating to the database activity (e.g. global reads and updates) and system usage (e.g. CPU usage).

There are several tools to collect these statistics, some of which go into enormous detail but they can hard to understand. The History monitor is designed to be simple to use, to run continuously on live systems and require less effort and expertise to understand the output.

It stores information in a small number of hourly and daily tables that are easily queried with SQL. So if you start it today the historic record is there when you need it. And of course you can still supplement the historic record with more detailed data collection when you have a problem to investigate.

What are the costs and benefits?

The history Monitor is very lightweight and won't add a significant load to the running system. The disk space used is also very small with storage adding up to about 130 MB per year even if you choose to extend the lifetime of hourly statistics as I have recommended in this article.

It is easy to configure and the output needs no further analysis to make it usable.

The day you turn the History Monitor on, you will see little or no advantage over other tools that can give more immediate detail. The benefit comes weeks or months later, when you are working on a capacity plan or investigating a performance issue.

It provides a historic record of many important metrics including

- CPU usage
- Size of database files and journals
- Global references and updates
- Physical reads and writes
- License usage

It also records a large number of more detailed technical metrics that could be helpful when investigating changes in performance of an application.

How do I access data stored by the History Monitor?

Tables

The information is stored in the %SYS namespace. It is readily accessible using SQL and can therefore be analyzed with any popular reporting package. There are four main daily tables SYSHistory.DailyDB, SYSHistory.DailySys, SYSHistory.DailyPerf and SYSHistory.DailyWD holding the daily summaries. There are equivalent tables holding the hourly summaries.

Hourly and Daily fields

The daily and hourly tables have a daily or hourly field respectively of the form ' 64491¦¦14400' where the two parts are the $h date and time values when the background job ran to generate the data. The time piece doesn't have much meaning in the daily tables.

Elementkey field

Some tables include the average and maximum values observed in each time period and the standard deviation. The type of entry is shown by the value of the elementkey field

Therefore a typical query to see the growth in average daily CPU usage would be:

SELECT Substr(DS.Daily,1,5) as DateH, (100-DS.SysCPUIdle) as AvgCPUBusy

FROM SYSHistory.DailySYS   DS

WHERE elementkey='Avg'

ORDER BY DateH

How do I configure and start the History Monitor?

Open a Caché terminal session and change to the %SYS namespace. Then run the command

Do ^%SYSMONMGR

You will be offered a number of character menus. Enter the numbers to make the following selections:

    5. Manage Application Monitor

2) Manage Monitor Classes

Then select choose activate option twice and specify the class names

 1)Activate/Deactivate Monitor Class

 %Monitor.System.HistoryPerf

 Yes

1) Activate/Deactivate Monitor Class

  %Monitor.System.HistorySys

  Yes

There are a number of of other classes but don't be tempted to activate them without testing first. Some use PERFMON and are not suitable for running on a live system.

Then use the exit option until you get back to the first menu. To activate the changes stop and then start the system monitor from the first menu.

1) Start/Stop System Monitor

1) Start/Stop System Monitor

The history monitor will run continuously even if the system is restarted.

You may want to keep the hourly statistics for longer than the default of 60 days. To do this use the method SetPurge(). E.g. to keep hourly statistics for a year:

%SYS>do ##class(SYS.History.Hourly).SetPurge(365)

More Complex SQL Example

For a more complicated example, if you also want the daily average CPU and the average CPU usage between 9am and 12 noon and you want information about global references and updates:

SELECT Substr(DS.Daily,1,5) Day,

      (100-DS.SysCPUIdle) as DailyAvgCPU,

      Round(AVG(100-H1.SysCPUIdle),2) MorningAvgCPU ,

       DP.PerfGloRef, DP.PerfGloUpdate

FROM SYSHistory.DailyPerf DP,

     SYSHistory.DailySYS DS,

     SYSHistory.HourlySys H1

WHERE DP.Daily=DS.Daily and DP.elementkey='Avg' and DS.elementkey='Avg'

and H1.elementkey='Avg'and substr(DS.Daily,1,5)=Substr(H1.Hourly,1,5)

and Substr(H1.Hourly,8,12) in (32400,36000,39600)

GROUP BY DS.daily

Which on my test system gives ...

Day	DailyAvgCPU	MorningAvgCPU	PerfGloRef	PerfGloUpdate
64514	13.64	2.25	99.03	7.73

64515	19.94	14.67	91.95	6.32
64516	8.79	12.14	102.21	6.91
64517	14.09	3.36	39729.06	5393.97
64518	20.26	25.11	15036.53	60.63
64519	9.27	15.5	3898.47	153.68
64520	5.54	1.78	87.94	5.65
64521	6.08	1.89	117.49	6.73
64524	17.8	16.81	70.8	5.14

Documentation

The history monitor is described in full in the documentation

#Monitoring #Performance #Caché

Source URL:https://community.intersystems.com/post/apm-using-iris-or-cach%C3%A9-history-monitor