Axibase Warehouse Designer

Warehouse Collection Agent

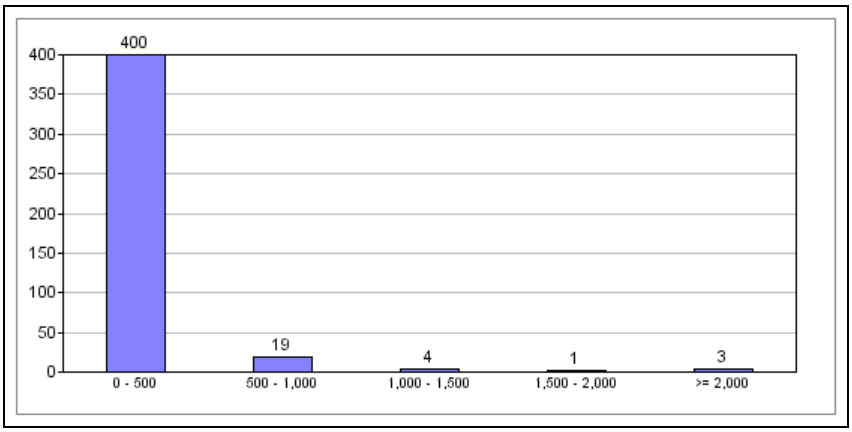
Date: August 6, 2008

Axibase Corporation
19925 Stevens Creek Boulevard
Cupertino, CA 95014
408.973.7897 phone
support@axibase.com
www.axibase.com

Overview

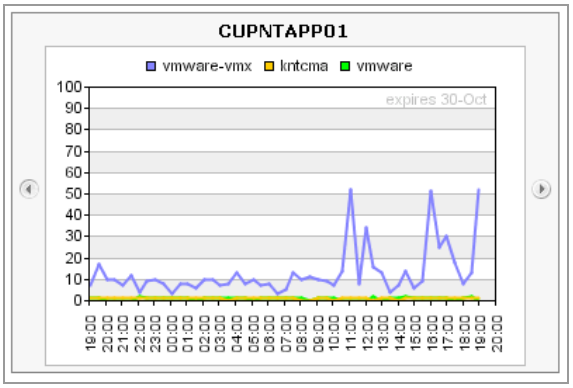
Performance analysis at the process level such as the ability to view historical CPU and memory utilization by process/program/user as well as to identify top-N resource consumers on a given server is a useful diagnostic technique. However, the sheer volume of raw data that needs to be collected from OS agents to generate process reports presents a serious challenge even for well-designed ITM installations. In fact, at 10 Mb of daily data throughput per agent (exceeding 100 Mb in some cases), very few ITM environments are capable of handling increased workloads without degrading the stability and performance of ITM's core functions such as resource monitoring and event processing. The data collection challenge is exacerbated by the fact that some servers have thousands of running processes as shown below.

Distribution of process counts in a medium size UNIX environment (~400 servers)

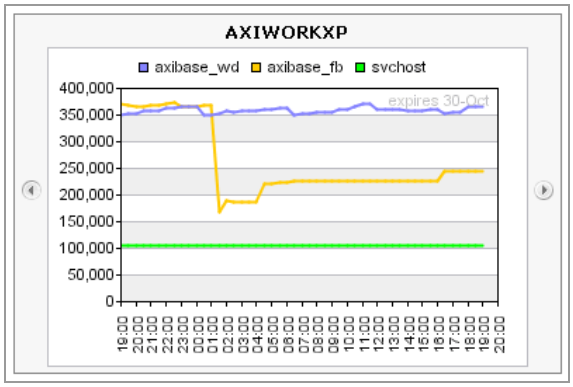


As a result, the Process attribute group, one of the most frequently requested by system owners and performance analysts, often remains disabled for historical collection due to technical constraints.

Top 3 Processes by Daily CPU Utilization



Top 3 Processes by Daily Memory Usage



Solution

Axibase Warehouse Designer v201 introduced a Warehouse Collection Agent (WCA) which is designed to make Process data collection affordable for all ITM customers.

WCA is a Java application that executes scheduled SOAP requests for real-time data from ITM agents and stores the data in the Tivoli Data Warehouse. TDW tables created by WCA adhere to TDW 2.1 schema conventions and can be queried by existing database clients including the Tivoli Enterprise Portal itself. Unlike ITM's native Warehouse Proxy Agent which collects complete data samples from all ITM agents, WCA can be targeted at particular agents or agent groups while the data samples can be minimized to include only user-specified attributes. This combination of agent targeting and custom samples substantially reduces network throughput and disk space requirements as illustrated below:

UNIX Agent: "Process"	**WPA**	**WCA**
Collection interval, minutes	15	15
Process instances per sample	150	150
Samples per day	96	96
Rows per agent per day	14,400	14,400
Avg. row size in TDW, bytes	**644**	**128**
Disk space per agent per day, Mb	9	2
Agents	**1,000**	**100**
Disk space per day, Mb	8,844	176
Disk space per week, Mb	**61,908**	**1,230**

Linux Agent: "Linux_Process"	**WPA**	**WCA**
Collection interval, minutes	15	15
Process instances per sample	100	100
Samples per day	96	96
Rows per agent per day	9,600	9,600
Avg. row size in TDW, bytes	**515**	**91**
Disk space per agent per day, Mb	5	1
Agents	**1,000**	**100**
Disk space per day, Mb	4,715	83
Disk space per week, Mb	**33,005**	**583**

Windows Agent: "NT_Process"	**WPA**	**WCA**
Collection interval, minutes	15	15
Process instances per sample	60	60
Samples per day	96	96
Rows per agent per day	5,760	5,760
Avg. row size in TDW, bytes	**246**	**100**
Disk space per agent per day, Mb	1	1
Agents	**1,000**	**100**
Disk space per day, Mb	1,351	55
Disk space per week, Mb	**9,459**	**385**

Requirements

- IBM Tivoli Monitoring, version 6.1+.
- Axibase Warehouse Designer, Enterprise Edition, version 201+.
- The following DDL command privileges are required for the warehouse database user (ITMUser): CREATE TABLE, DROP TABLE, CREATE INDEX, DROP INDEX, ALTER TABLE, ALTER INDEX.

Implementation Notes

- Even though the Warehouse Collection Agent was originally designed for pulling Process information from base OS agents, it is capable of collecting data from other ITM agent types, including custom and universal agents.
- Tables created by WCA in the TDW database are named based on the underlying attribute group by adding a "_Poll" suffix. For example, a collection table for "Linux_Process" attribute group is called "Linux_Process_Poll".
- A typical WCA collection table schema includes a subset of columns from the underlying table. The differences are illustrated in Figure 1 using UNIX Process table as an example.
- WCA consists of multiple independent collectors, each responsible for pulling data from a particular attribute group. To avoid conflict, only one collector for each underlying attribute group is supported. For example, collecting process CPU utilization at 5 minutes and process Memory usage at 15 minutes using multiple collectors is not supported. Instead, a single collector for both CPU and Memory usage must be configured for the Process attribute group with a meaningful interval.
- WCA does not require direct connectivity to the agents. Instead, the application utilizes the SOAP interface on Hub TEMS for sampling data from remote agents.
- WCA itself is not an ITM agent and does not appear in the managed system list or the Physical View navigator in TEP.
- WCA does not have a dependency on the status of historical data collection for the underlying attribute group, however for most practical purposes WCA should not be collecting data for attribute group that are already enabled for collection in ITM.
- WCA provides an option to prune data from collection tables. The retention interval is configured for each collector separately. The default pruning interval is 1 week.
- WCA does not provide summarization capabilities.
- WCA does not create Timestamp (sample time) and TMZDIFF (time zone offset) columns since all samples are time-stamped by the server.
- WCA is a multi-threaded program and as such allows several collector threads to execute at the same time. AWD administrators can control the maximum number of concurrent collectors by changing `server.collector.concurrent.limit` property in `server.properties` file. The default limit is 4 threads.
- WCA can only collect data from online agents as reported by the Hub TEMS. If the agent or its managing TEMS server (RTEMS) is offline, the collector does not issue a request for this agent.
- If AWD is configured to collect data from multiple Hub TEMS servers which use different TDW databases or multiple user schemas in the same database, the collector creates *_Poll table in each database/schema to insert agent data based on their TDW affinity.
- Unlike WPA, WCA does not force agents to store short-term historical data on a local file system or TEMS. The data is discarded by the agent upon completion of the request.
- WCA aggregates requests for multiple agents into a single SOAP query to minimize incremental workload on the TEMS servers. Agents are grouped by (R)TEMS into batches which are limited to 16 agents by default. The maximum batch size can be modified for each collector instance in the collector editor. The optimal batch size depends on the average number of instances returned by a particular agent for the same request. It is advisable that the total number of instances not to exceed 1,000 for each batch.

- Each collector provides a way to automatically create database indexes on collection tables in order to speed up execution of SELECT statements by database users and client programs. Since indexes require additional disk space and are likely to slow down INSERT statements, it is recommended that collector performance is regularly reviewed by AWD administrators using [Execution Log] statistics available in the collector editor.
- The minimum collection interval is 5 minutes. Because some ITM agent types (e.g. WMQ agent, universal and custom agent) implement caching and allow users to customizing the caching interval, it recommended that the WCA collection interval always exceeds agent caching interval to avoid unnecessary polling.
- Whenever the set of collected attributes is modified in the collector editor, WCA alters the collection table schema by adding or dropping columns based on new configuration. If the underlying database server does not support COLUMN DROP command (such as IBM DB2 v8-), the table will retain the column which will be populated with NULL values.
- WCA collectors can fail for a variety of reasons such as network faults, database login errors, malformed SOAP responses, or version incompatibilities between agents and TEMS servers. If the failure is non-recoverable, the collector is disabled. Otherwise, the collector re-tries execution at the next scheduled interval. In both cases, an alert is dispatched to AWD administrators' email addresses for further investigation.
- SOAP execution and database insert statistics are tracked on a per-collector basis and are available under [Execution Log] tab in the collector editor.
- WCA collection tables are available for ad-hoc, scheduled, and real-time reporting similar to the underlying tables, including top-N template reports (Figures 4 - 5).

Configuration

- **Creating a new collector:**
  - Login as Administrator
  - Click on [Admin] tab in the top menu
  - Click on 'Collectors' link
  - Click on [Add] link (Figure 2)
  - On the [Query] tab, select a product (agent type), such as Windows NT (Figure 3)
  - Select a table (attribute group), e.g. Process
  - Select one or several columns, e.g. % Processor Time
  - Select at least one key column if the attribute group returns multiple instances per table. For example, select Binary Path, ID Process, Process Name for the NT Process group.
  - Click [Validate] to check if configuration is valid.
  - Click on Properties tab and modify default poll (sampling) interval, pruning (retention) interval, and database index properties.
  - Click [Save]
  - Click on [Overview] tab and change Status to 'Enabled'
  - Click on [Targets] tab and drag-and-drop agent group, host or agent icons from the right pane (available targets) to the left pane (collector targets).
  - Click [Save]
  - Once the collector has at least one target and validates without errors, it will be added to the WCA execution queue.

- **Modifying an existing collector:**
  - Login as Administrator
  - Click on [Admin] tab in the top menu
  - Click on 'Collectors' link
  - Click on the collector name in the Collector Status table (Figure 2)
  - Modify collector query and properties as described above.
  - If the underlying table is redefined for an existing collector, the WCA will not automatically delete any data from the old collection table. To delete the data or to drop the table manually, use [Truncate] and [Drop] buttons prior to reconfiguring the table.
  - To force WCA to collect data ahead of scheduled execution for a given collector, click on [Execute] button.

- **Deleting a collector:**
  - Login as Administrator
  - Click on [Admin] tab in the top menu
  - Click on 'Collectors' link
  - Click on the collector name in the Collector Status table (Figure 2)
  - The WCA will not automatically delete any data from the old collection table when the collector is deleted. To delete the data or to drop the table, use [Truncate] and [Drop] buttons prior to deleting the collector.

Tables and Figures

Figure 1. WPA/WCA Table Schema Differences

UNIX “Process” table: 63 columns

...	Name	Data type	Length	...
	TMZDIFF	INTEGER	4	...
	WRITETIME	CHARACTER	16	...
	"System_Name"	VARCHAR	64	...
	"Timestamp"	CHARACTER	16	...
	"Process_ID"	INTEGER	4	...
	"Flag"	CHARACTER	8	...
	"Execution_State"	CHARACTER	4	...
	"User_ID"	INTEGER	4	...
	"Parent_Process_ID"	INTEGER	4	...
	"CPU_Utilization"	INTEGER	4	...
	"Priority"	INTEGER	4	...
	"Nice_Value"	INTEGER	4	...
	"Entry_Address"	CHARACTER	8	...
	"Size"	INTEGER	4	...
	"Event_Waited_On"	CHARACTER	8	...
	"Terminal_Device"	CHARACTER	8	...
	"Time"	CHARACTER	8	...
	"Command"	VARCHAR	32	...
	"Process_Command"	VARCHAR	100	...
	"Reptype"	CHARACTER	4	...
	"Real_Group_ID"	INTEGER	4	...
	"Effective_User_ID"	INTEGER	4	...
	"Effective_Group_ID"	INTEGER	4	...
	"Process_Group_Leader_ID"	INTEGER	4	...
	"Session_ID"	INTEGER	4	...
	"Scheduling_Class"	CHARACTER	8	...
	CPU_ID	INTEGER	4	...
	"User_Name"	VARCHAR	32	...
	"StartTime"	CHARACTER	16	...
	"Elapsed_Time"	CHARACTER	12	...
	"Virtual_Size"	INTEGER	4	...
	"Mem_Pct"	DECIMAL	31	...
	"CPU_Pct"	DECIMAL	31	...
	"Total_CPU_Percent"	DECIMAL	31	...
	"Sample_CPU_Pct"	DECIMAL	31	...
	"Heap_Size"	INTEGER	4	...
	"Stack_Size"	INTEGER	4	...
	"Major_Fault"	INTEGER	4	...
	"Minor_Fault"	INTEGER	4	...
	"Context_Switch"	INTEGER	4	...
	"Involuntary_Context_Switch"	INTEGER	4	...
	"User_CPU_Time"	CHARACTER	12	...
	"System_CPU_Time"	CHARACTER	12	...
	"Total_CPU_Time"	CHARACTER	12	...

+ 19 more columns

UNIX “Process_Poll” table: 7 columns

...	Name	Data type	Length	...
	"System_Name"	VARCHAR	64	...
	WRITETIME	VARCHAR	16	...
	"Command"	VARCHAR	32	...
	"Process_Command"	VARCHAR	100	...
	"Process_ID"	BIGINT	8	...
	"User_Name"	VARCHAR	32	...
	"CPU_Pct"	DOUBLE	8	...

Figure 2. Collector Status Table

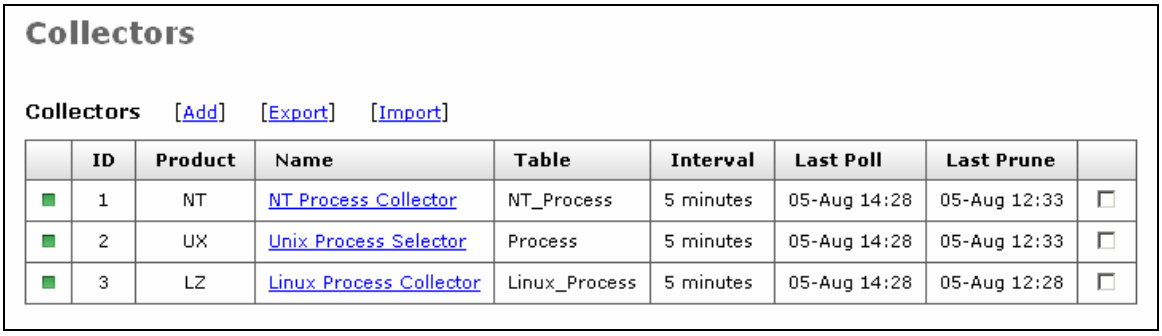


Figure 3. Collector Editor

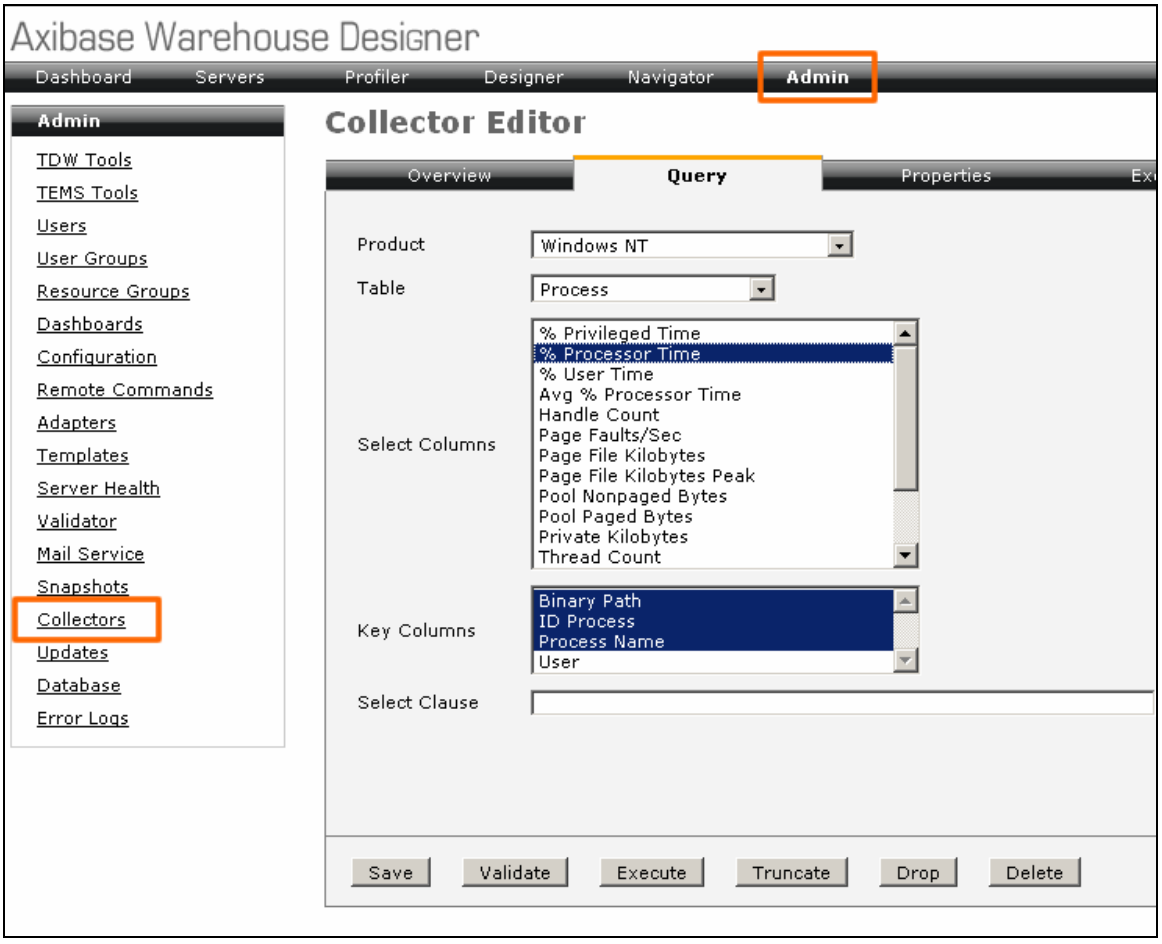


Figure 4. Windows NT_Process_Poll table in Designer

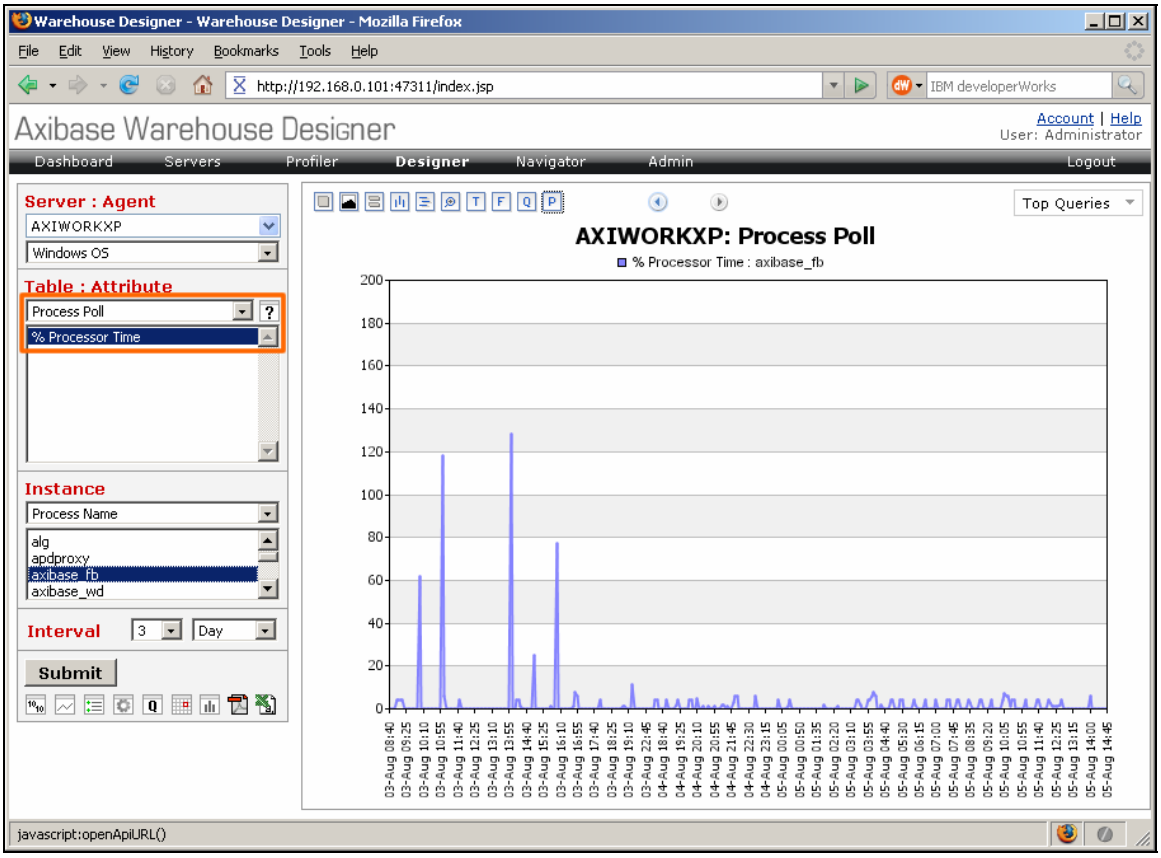


Figure 5. Top-3 Windows processes by CPU template in Profiler