



Log Performance Timer VMware vRealize Automation

for

VMware Professional Services

Prepared by

Chris Dombrowski

VMware Professional Services

cdombrowski@vmware.com

Version History

Date	Ver.	Author	Description	Reviewers
07/02/18	0.1	Chris Dombrowski	Initial Release	

© 2018 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. This product is covered by one or more patents listed at <http://www.vmware.com/download/patents.html>.

VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

VMware, Inc.
3401 Hillview Ave
Palo Alto, CA 94304
www.vmware.com

Contents

Version History.....	2
Contents.....	3
1. General Requirements	4
2. Solution Design	4
2.1 Standard Time Measurement Action	4
2.2 Log Insight parsing	5
3. Operation	8
3.1 Measuring workflow runtime.....	8

1. General Requirements

Identifying and understanding potential performance issues with vRO code has always been a difficult endeavor. In most cases, debug code is added to suspect code segments to determine where potential bottlenecks exist and perhaps why the bottleneck is occurring. This type of debug code is reactionary, limited in scope and effect, and typically removed or forgotten once the immediate bottleneck is defined.

What is needed is a standardized method of adding code performance measurements inline for any vRO code segment (action, workflow, scriptable task, external call, etc.) that can be consolidated and presented in a normalized way.

2. Solution Design

The basic time measurement in vRO (and any other code engine) is done by calculating the difference between two time variables, typically `startTime` and `endTime`. Once the time measurement is available, it can be as simple as printing the value to a log to make it available.

The issues faced within vRO code with regard to time measurements are:

- How to save the start time of a measurement when traversing multiple vRO code elements
- How to have multiple time measurements in flight
- How to view all time measurements and consolidate equivalent measurements graphical data.

The solution has fixed on a single vRO action to “mark” any start time and provide the time measurement when called a second time. To make multiple in flight measurements possible each time variable is identified either by the current workflow token id or by a UUID provided in the vRO action input. When the time measurement is complete a standard log entry is made that can be parsed by vRealize Log Insight to provide consolidated data views.

2.1 Standard Time Measurement Action

A vRO action named “*logPerformanceTime*” in the module *com.vmware.pso.util* has been created to provide code execution time measurements. The basic functions of this action are:

- This action writes a log message (for log insight or other logging) that details the start time, end time and duration of a workflow.
- The start time of the primary workflow is given by the executing workflowToken object.
- The start time of any embedded child workflows in the primary workflow is set by calling this action with an input of “mark” (case insensitive). This marks the current position as time zero for any calls not in the primary workflow.
- All calls of this action in child workflows will reset the time zero mark. The action must be called again to mark a time zero.
- This action can also be used in actions used for vRA UI form inputs. The msg input format is as follows:
 - To “mark” a time zero -- `mark::<msg>::UUID`
 - To log the time -- `end::msg::UUID`
 - The keyword mark is required. Any other keyword can be used to log the time. The double colon demarcation is required.
 - The UUID can be created in the measured action using `System.nextUUID()`. Be sure to send the same UUID when marking and logging time of an act

Individual start times are saved in a configuration element in the *PSO /Performance* folder, in the configuration element *workflowTokens*. Workflow tokens and UUIDs are automatically removed from this configuration element when the time measurement completes, and when the “clean” message option is used.

The logPerformanceTime action writes its output to the System.log, which places messages in both the vRO interactive-scripting log and the Server log. When parsing the messages on Log Insight, ensure only one log is used by using the filter function to specify the log, if necessary.

2.2 Log Insight parsing

Log Insight is used to parse and consolidate the logPerformanceTime action’s output. The basic method of parsing is to use the *[perfruntime=* message header to identify the time measurement, and the *[perfmsg=* header to identify the different workflows, actions, or whatever standard code indicators you need to use.

To create a log insight chart for inclusion into a Log Insight dashboard detailing a particular performance measurement, perform the following:

1. Ensure vRO is attached and providing logs to Log Insight
2. Add the logPerformanceTime action to the workflows and actions that will be used for the performance measurement (see the *Operation* section for examples).
3. Execute the workflow to add the logPerformanceTime messages to the vRO logs. It is best to execute the workflow multiple times to produce multiple messages.
4. Access Log Insight Interactive analysis
 - a. Filter the output to find the logPerformanceTime messages. This can be done by adding the filter “text” “contains” “perfmsg”. Set the timeframe to show the data.
 - b. The following fields need only be created during the first chart creation. These fields will be available for use for subsequent charts
 - i. Create a new Field by highlighting the message content you have set immediately after *[perfmsg=* and immediately prior to the end bracket “]”, and select “Extract Field”. Name the field created “**perfmsg**”. Ensure the “Available for” dropdown is set to **All Users** if the dashboards will be shared between users or exported.

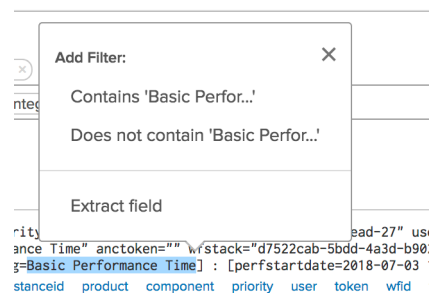


Figure 1- Select the performance message

The 'Fields' configuration window shows the following details:

- Field name:** perfmsg
- Available for:** All Users (dropdown)
- Extracted value (regex):** Custom regex: .*
- Pre and post context (regexes):**
 - Pre: perfmsg=
 - Post: \]: \[perfstartdate
- Buttons:** + ADDITIONAL CONTEXT (recommended), DELETE, DUPLICATE, UPDATE

Figure 2- Name the field and set availability

- ii. Create a new Field by highlighting the time measurement (an integer given in milliseconds) immediately after `[perfruntime=` and immediately prior to the end bracket `]`, and select “Extract Field”. Name the field created “**perfruntime**”. Ensure the “Available for” dropdown is set to **All Users** if the dashboards will be shared between users or exported.

The 'Add Filter' dialog box shows the following options:

- Contains '4294'
- Does not contain '4294'
- Extract field

The log entry below shows a field named `perfruntime` with the value `4294` highlighted.

Figure 3- Select the perfruntime value

The 'Fields' configuration window shows the following details:

- Field name:** perfruntime
- Available for:** All Users (dropdown)
- Extracted value (regex):** Integer: -?d+
- Pre and post context (regexes):**
 - Pre: perfruntime=
 - Post: \]
- Buttons:** + ADDITIONAL CONTEXT (recommended)

Figure 4- Name the field and set availability

- c. Remove the “text” filter and add a filter “perfmsg” “contains” with a value of the performance message used in the workflow/action. You can also use “matches regex” instead of “contains” on this filter for an exact match. Ensure the filepath filter is valid if required.



Figure 5- Set the chart filter

- d. Set the “Chart Type” from “Count Of Events” (the default) to “Numeric Function for” and select “perfruntime” in the dropdown. Select Average, Min, and Max, then select “Apply”. The Log Insight chart should change to a line chart with values of the perfruntime shown over time. If the chart is not a line chart, you can change the chart from Automatic to Line in the right hand dropdown.

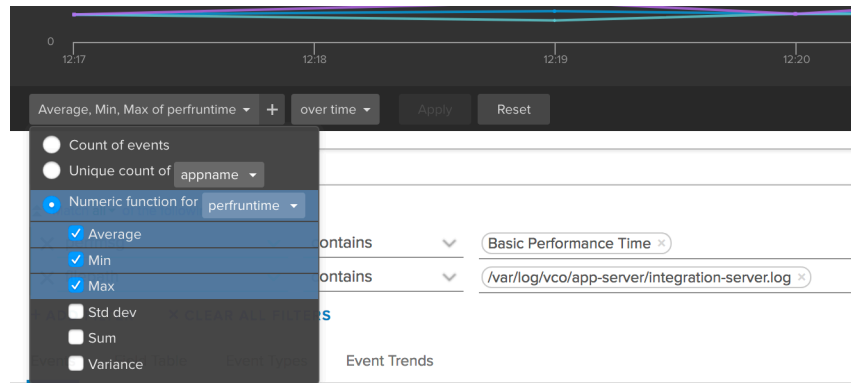


Figure 6- Set the chart type to Numeric perfruntime

- e. The chart can now be added to a Log Insight dashboard. Depending on your Log Insight access the dashboard can be made available to other users or exported for use in another Log Insight.

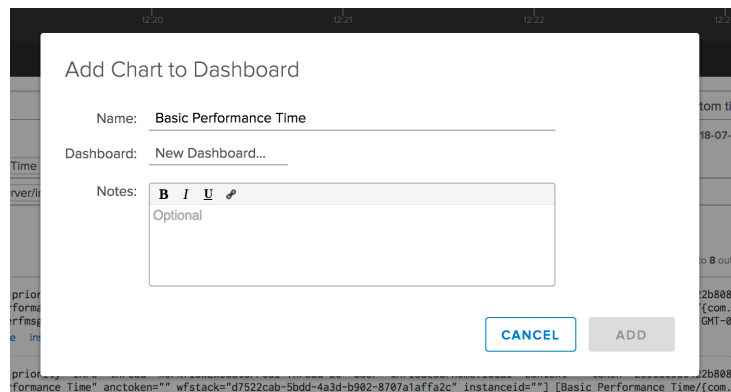


Figure 7- Add chart to dashboard

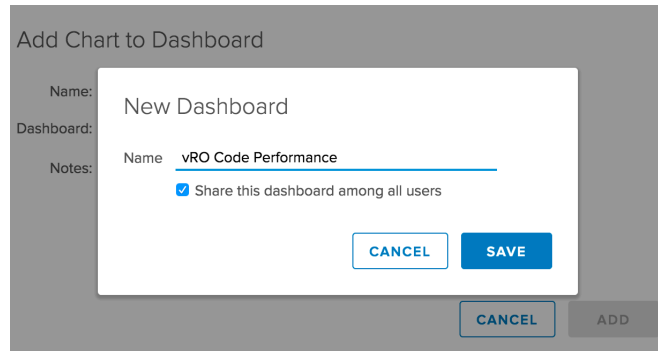


Figure 8- Create new dashboard

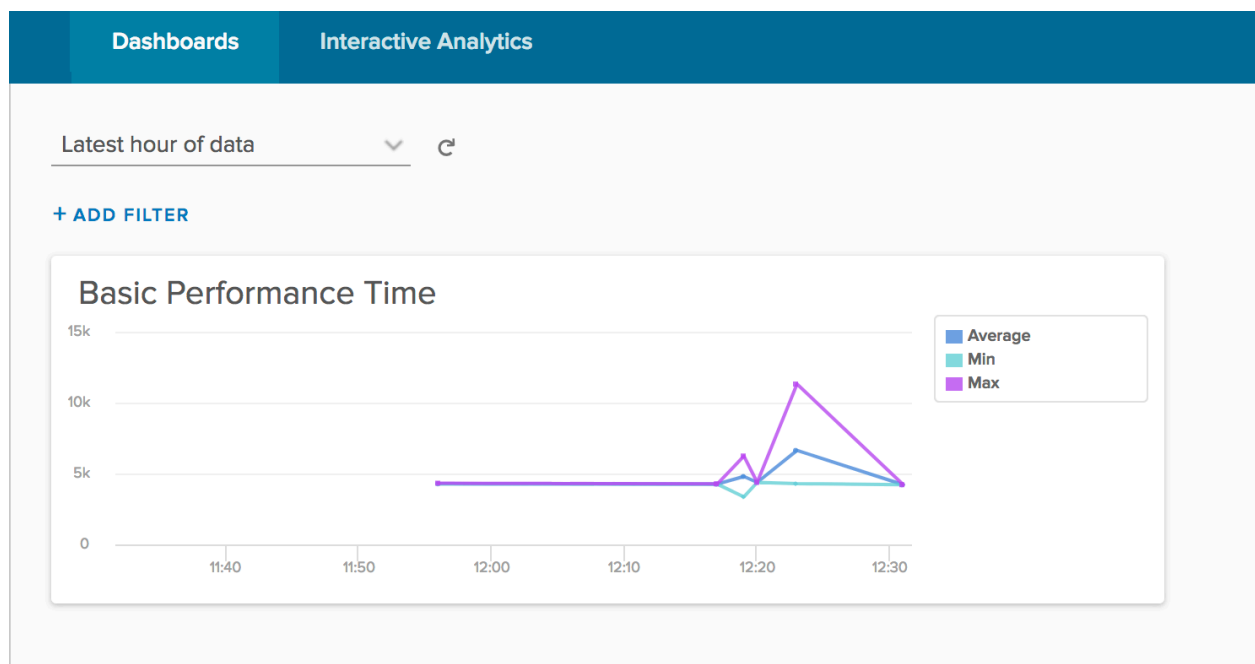


Figure 9 - Completed performance chart on the dashboard

Additional dashboards can be created simply by accessing the Log Insight Interactive Analysis and modifying the filter for “perfmsg” to show the different performance messages made available within the vRO code base.

3. Operation

3.1 Measuring workflow runtime

A particular workflow runtime can be measured by placing the logPerformanceTime action at the beginning and at every exit of the workflow. The input message of the first action call is “mark”, and the input of the exit calls can be anything that provides meaning for log insight consolidation; typically the current workflow name is used as the exit message. It is possible, but not necessary, to have separate messages for error paths in order to identify traversal of those paths in Log Insight.

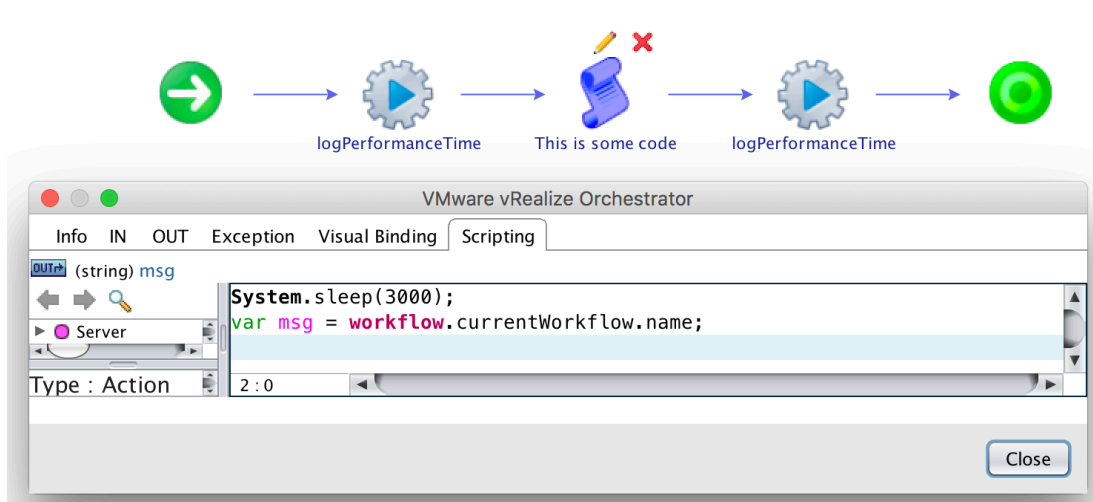


Figure 10- Basic Performance Time workflow showing “mark” action and message action

Child workflows that include calls to the logPerformanceTime action do not affect any calls to the action made by the parent workflow. Parent workflow performance times, however, will include all child workflow times since the codetiming is linear from start of the parent workflow to the end of the parent workflow

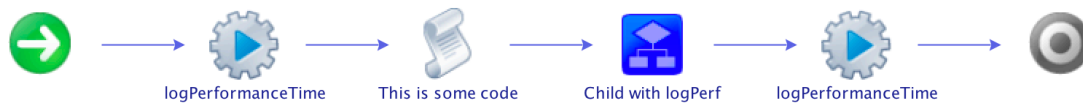


Figure 11- Child workflow with logPerformanceTime calls

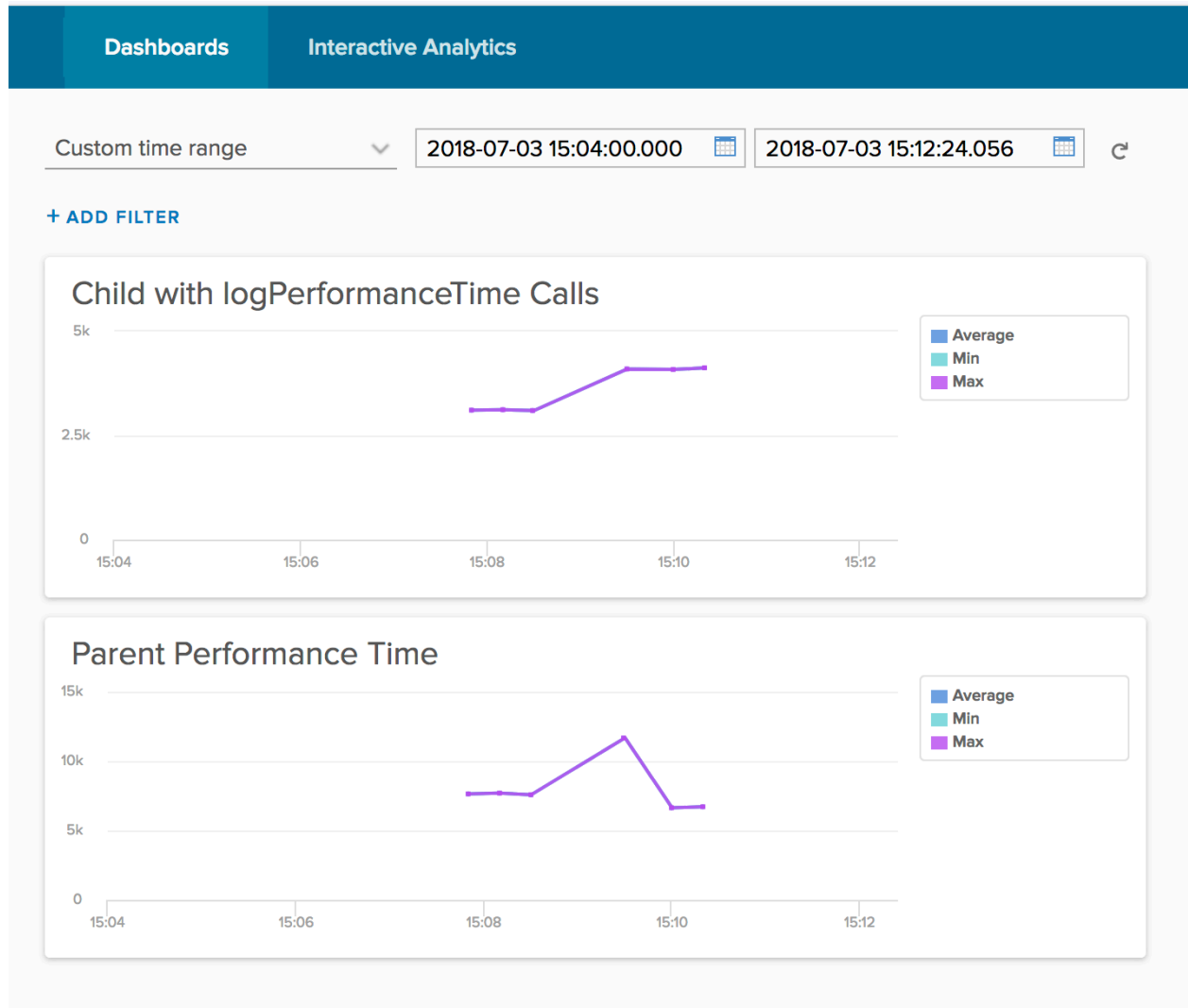


Figure 12- Parent and nested child workflow performance. Note the parent includes all child time.