ready. set. go.

vmware PARTNER EXCHANGE

vSphere API Best Practices

TEXIBP1001

Balaji Parimi, VMware Inc.

FEBRUARY 8 – 11, 2010 | MANDALAY BAY | LAS VEGAS, NEVADA

Disclaimer

This session may contain product features that are currently under development.

This session/overview of the new technology represents no commitment from VMware to deliver these features in any generally available product.

Features are subject to change, and must not be included in contracts, purchase orders, or sales agreements of any kind.

Technical feasibility and market demand will affect final delivery.

Pricing and packaging for any new technologies or features discussed or presented have not been determined.

"These features are representative of feature areas under development. Feature commitments are subject to change, and must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery."

Follow Us

Join the conversation
Get the latest scoop
Be a part of the network

Facebook
Search for
VMware Partner Network

Twitter
Search for
VMware_Partners

Linkedin
Search for
VMware Partner Network

Subscribe to RSS

http://blogs.vmware.com/developer

Follow this event LIVE!
#VMwarePEX2010

Help us help you! Please fill out the survey at the end of the course.

Introduction

This session assumes that the audience are familiar with the vSphere Web Services SDK.

The various use case scenarios outlined in this session are based on our experiences in working with various partners.

About the Speaker

Balaji Parimi, Sr. MTS, VMware Inc.

- Current role and responsibilities
  - VMware engineering consultant to the partners, focusing on creating solutions based on VMware's vSphere platform.
  - Train partner engineering teams on using the vSphere APIs.
  - Review partner's solution architecture.
- Background
  - Have a master's degree in Computer Science, and a master's degree in Software Engineering.
  - Have been working in the industry since 1998.

Agenda

vSphere API Usage Best Practices

- Client Session Management
- Views
- Managed Object Properties
- Performance Statistics
- Virtual Hardware Provisioning Tidbits
- VMotion, Storage VMotion Tidbits
- Changing vCenter Configuration parameters using the SDK
- Debugging
- Resources

Q & A

Client Session Management

Use Case: The client session times out after 30 minutes of inactivity. What are the options to maintain session validity with least overhead?

- Re-establish the session whenever it expires.
- Caveats:
  - Requires exception handlers for all SDK API calls to re-establish session when session not authenticated exception occurs.
  - Session specific objects need to be re-initialized whenever a new session is established.
- Maintain session validity by making sure that the interval between invoking two API calls is less than 30 minutes.
- Caveats:
  - If invoking an API call, just to maintain session validity, make sure to invoke a method that generates the least amount of traffic. E.g.: Retrieve CurrentTime from the server.

Views

Use Views to monitor changes, retrieve properties of managed objects or managed object references

- InventoryView covers the entire inventory.
- ListView covers the collection of specified managed objects.
  - The onus is on the client application to provide the collection of managed object references required for creating the view.
- ContainerView covers the managed objects in a container.

InventoryView

To monitor for changes to inventory:

- Create InventoryView
- Open the inventory view folders to monitor for changes
- Monitor for changes to ‘view’ property, as it contains the MoRefs of all the managed objects in the inventory

```
vimPort.openInventoryViewFolder(
          inventoryViewMoRef,
          new ManagedObjectReference[]{}//MoRefs of
                        //all the containers to expand
          );
```

ListView

To monitor for changes to the properties of a group of objects:

- Create ListView with the MoRefs of all the managed objects
- Create the appropriate TraversalSpec to start from the created ListView
- Monitor for changes to the properties of the objects using PropertyCollector

```
vimPort.createListView (
          serviceContent.getViewManager(),
          new ManagedObjectReference[]{}//MoRefs of
                      //all the VMs
          );
TraversalSpec tSpec = new TraversalSpec();
tSpec.setType(“ListView”);
tSpec.setPath(“view”);
```

ListView

```
PropertySpec pSpec = new PropertySpec();
pSpec.setType(“VirtualMachine”);
pSpec.setPathSet(propertyArr);//Props String array

ObjectSpec oSpec = new ObjectSpec();
oSpec.setObj(viewObj);//View object
oSpec.setSelectSet(new SelectionSpec[]{tSpec});
```

ContainerView

Retrieve the managed object references of entities in a container using ContainerView:

- Obtain the managed object reference of the container
- Create the ContainerView
- ContainerView has a property with the name ‘view’ and it is an array of type ManagedObjectReference. This contains the MoRefs of all the types specified during the ContainerView creation.

```
vimPort.createContainerView(
          serviceContent.getViewMgr,
          containerMoRef,
          new String[]{“VirtualMachine”},//MO type
          true //recursively look in the container
          );
```

Managed Object Properties

Use Case: I want to retrieve the names of all virtual machines, and ESX hosts in my inventory. How can I achieve this using one API call? Is it possible to retrieve both the MOREF and some properties of a managed object?

- Use View objects (InventoryView, ContainerView and ListView).
  - Have to create the View object first
  - Use PropertyCollector to retrieve the desired properties.
    - TraversalSpec is different as the starting object is View object.
- Alternative approach:
  - Create appropriate PropertyFilterSpec objects for retrieving the properties of different managed objects and use property collector to retrieve those properties in one call.
  - The return value (ObjectContent) contains the MOREF (in the obj property), and the requested properties (in the propSet property).

Managed Object Properties

Use Case: What is the best caching strategy (caching the managed object properties or their references)?

- Caching the managed object properties makes sense when dealing with properties that seldom change (e.g.: data centre name, vm name etc.)
- Caching the managed object references rather than their properties makes sense for properties that change very often (e.g.: VM power state). Since, the managed object references do not change until the managed object is removed from the inventory, it is easier to maintain the cache.
- Caveats:
  - Make sure to update the cache with the relevant MOREF whenever there is a change to the inventory.

Managed Object Properties

Use Case: I want to retrieve the managed object references of all the hosts or virtual machines in a container. I want to retrieve the properties of these objects periodically.

- Create ContainerView with the Datacenter managed object with the type property set to HostSystem or VirtualMachine or both.
- It contains managed object references to all the managed objects.
- It always represents the current configuration of the inventory and reflects any subsequent changes.
- Can be used to track changes to the inventory in a specific container (like a folder, data center, resource pool, compute resource etc.).
- Can be used to monitor for changes to the properties of VirtualMachine or HostSystem objects in a container.
- Caveats:
  - The View objects are session specific.

Managed Object Properties

Use Case: What are the different ways to monitor for changes to the properties of several different managed objects?

- Use View objects for getting updates on a specific set of objects (this can avoid creating / destroying PropertyFilter objects).
  - This can allow you to add / remove objects to the View without having to create a new PropertyFilter object.
- Alternatively, create appropriate PropertyFilter objects for monitoring the properties of managed objects.
  - Use this approach only when you are monitoring properties of very few specific objects.

Managed Object Properties

Some tidbits for monitoring applications

- Use summary property whenever specific objects are being monitored. Since summary property encapsulates other useful properties of the object as well (e.g.: VM summary property contains VM runtime info, basic configuration, guest info etc.).
- All checkForUpdates and waitForUpdates return as soon as any change occurs to any of the properties in any of the PropertyFilter objects.
- Use checkForUpdates, if you want to find if there is a change since the last version you know.
- Use waitForUpdates, if you want to monitor the changes.
- Caveats:
  - There is no notion of associating a specific PropertyFilter object to a specific check/waitForUpdates API method.
  - The subsequent invocations of check/waitForUpdates API method requires using the version string returned from the previous invocation for correct results.
  - The PropertyFilter objects and ListView, ContainerView, and InventoryView are session specific.

Performance Statistics

Use case: Describe the best practices in retrieving the performance statistics?

- Retrieve the performance counters information once per vCenter / ESX server and reuse them.
- The counters information is the same for all ESX hosts. If your application is retrieving the statistics directly from the ESX host, you don't have to retrieve the counters information for every host.
- Performance provider summary is not for a specific entity, but for an entity type. E.g.: The performance provider summary is the same for all virtual machine objects.
- Retrieve the available performance metrics information once per managed entity and reuse them.
  - Need to retrieve these again if there is any change to the hardware (e.g.: a NIC is added / removed, a disk is added / removed etc.).
- Use wild card for performance metrics information instead of specific metric IDs.

Performance Statistics

- Create performance query specification based on the performance counters and metrics information already obtained.
- Reuse the performance query specification, if the performance statistics are being queried repeatedly for a managed entity.
- Use multiple query specifications instead of one query specification for every queryPerf API invocation.
- Use CSV format for stats query output, especially if your query retrieves more data.
- Caveats:
  - Make sure the VC server stats level is set to the appropriate value for the counter in question.

Performance Statistics

Use case: Explain the intricacies in retrieving the real-time performance statistics?

- Check the performance provider summary for the entity to confirm that it supports real-time stats.
- Real-time statistics can be retrieved for any stats level irrespective of the vCenter Server stats level.
- Query the real-time stats at a slower rate than the supported refresh rate.
- Caveats:
  - Do not compare the information with the results of “esxtop”. The ultimate source of information is the same, but the information metric and the refresh rate are different.

Events

Use case: How can my application receive event notifications for the events of interest?

- Create EventHistoryCollector with the appropriate EventFilterSpec, to get only the events of interest.
- Every event that passes the EventFilterSpec is appended to the ‘latestPage’ property of the EventHistoryCollector.
- Use waitForUpdates API method to keep track of changes to the ‘latestPage’ property of the EventHistoryCollector.
- Process the event and send notifications.
- Caveats:
  - The application should keep track of the events it has processed.
  - There is a risk of losing events if the events are generated faster than they could be processed.

Events

Use case: My application requires triggering a work flow based on a specific event or property change. How can I achieve this?

- Using the EventHistoryCollector, your application can get event notification. The event processing part can trigger the work flow or any other action.
- Using PropertyFilter and waitForUpdates monitor the ‘latestPage’ property of the EventHistoryCollector for any changes. The client can trigger the work flow or any action as changes occur.
- Caveats:
  - Resort to this approach only if the supported alarms do not serve the purpose.

Tasks

Use case: My application requires keeping track of the status of all the tasks. My application runs into exceptions occasionally with the message, “managed object reference not found”. What should I do to avoid this?

- Any API that ends with “_Task” returns a Task object so that the client does not need to wait for the call to complete.
- The Task objects are very short lived, and hence need to be used as soon as possible.
- If the returned Task object cannot be used immediately, resort to using TaskHistoryCollector.
- TaskHistoryCollector, can provide information about the completed tasks.
- Caveats:
  - The latestPage of TaskHistoryCollector contains TaskInfo of the completed tasks.
  - The oldest item is removed to make room for the new one.

Virtual Hardware Provisioning Tidbits

Virtual hardware is added to the VM using the reconfigVM_Task API.

Every hardware device (NIC, disk, or any peripheral device) requires a unique key to distinguish itself from the other devices in the virtual machine. This is a positive number assigned by vSphere after the device is successfully added to the virtual machine.

While adding a device you have to use a number that has not been used by the existing device. You can use a negative number like "-100".

While updating or removing the device, retrieve the device object for the virtual machine, make the necessary changes, set the VirtualDeviceConfigSpec operation to the appropriate value and then invoke the reconfigVM_Task API to effect the changes.

VMotion / Storage VMotion Tidbits

Check whether the VMotion / Storage VMotion can succeed prior to initiating it.

Use checkMigrate_Task to check if VMotion can succeed for a virtual machine.

Use checkRelocate_Task to check if Storage VMotion can succeed for a virtual machine.

Use queryVMotionCompatibilityEx_Task to check for VMotion compatibility of multiple virtual machines to multiple hosts.

Always specify the target resource pool if the virtual machine is in a specific resource pool at the source, otherwise VMotion / Storage VMotion fails. If the target resource pool is not specified, the vCenter server tries to VMotion the virtual machine to the same resource pool at the destination host, hence the failure.

Changing vCenter Configuration Parameters using SDK.

Use the “queryOptions” API to get all the available configuration options.

The key string for option name has to match with the name of that option in the vCenter server.

Use the “updateOptions” to update the necessary options.

Debugging

Look at the actual SOAP messages that are exchanged between the client and the server. Simplest way to do that is to enable HTTP protocol (look at page#25 in the Developer's set up guide) and use tcpmon. You can use some other tool that can capture with HTTPS.

Change the log level to 'trivia', and examine the log files.

Host agent (hostd) on the ESX host carries the operations invoked via the vCenter Server using the SDK.

The log files are rotated for vCenter server, and hostd.

For vCenter server logs, vpxd-index file (C:\Documents and Settings\All Users\Application Data\VMware\VMware VirtualCenter\Logs) contains the number indicating the current log file.

For hostd logs, hostd-index file (/var/log/vmware) contains the number indicating the current log file.

Additional Resources

Developer's Setup Guide:

http://www.vmware.com/support/developer/vc-sdk/visdk400pubs/sdk40setupguide.pdf

Programming Guide:

http://www.vmware.com/support/developer/vc-sdk/visdk400pubs/sdk40programmingguide.pdf

API Reference:

http://www.vmware.com/support/developer/vc-sdk/visdk400pubs/ReferenceGuide/index.html

Developer Community:

http://developer.vmware.com

Code Samples:

http://communities.vmware.com/community/developer/codecentral

Other Useful Sessions

TEXIBP 1004 – All About Events, Alarms and Tasks (11/02/10 @ 1:30PM – Lagoon B)

TEXIBP 1005 - Developing Performance Monitoring Tools Using the vSphere APIs: Simple Troubleshooting Techniques and Programming Best Practices (11/02/10 @ 2:45PM – Lagoon C).

Any Questions?

Backup Slides

Backup slides

Managed Object Properties

Use case: I would like to monitor the changes to the inventory. What is the simplest way to achieve this with out burdening the VC server?

- Use InventoryView, appropriate PropertyFilter and waitForUpdates.
- Keeps track of changes to only the open folders.
- The onus is on the client application to open the appropriate folders.
- More useful for UI clients that display the inventory.
- Caveats:
  - The InventoryView is session specific.