DEPLOYING SPLUNK® ENTERPRISE INSIDE VIRTUAL ENVIRONMENTS

Configuring virtual machines to run Splunk software

Background

Splunk® Enterprise is the platform for analyzing machine data. Splunk software can be deployed in physical, virtual, cloud or hybrid environments. When deploying Splunk software in a virtualized environment, it is important to plan appropriately. While Splunk Enterprise fully supports virtual deployment, careful planning and configuration should be followed to minimize effects of virtualization overhead. This tech brief describes performance considerations and guidelines for deploying Splunk core components inside virtual machines. Some of the recommendations are specific to particular environments, such as Linux and VMware. For your specific environment, consider the function of the recommendations, not necessarily the specific configuration.

Splunk Deployment Components

The typical components that make up the core of a Splunk deployment include Splunk forwarders, indexers and search heads. Other, less-common components include deployment servers and cluster masters. Each of these components can be run independently inside different virtual machines. However, the system resources required to run these components vary and should be planned beforehand:

Forwarders collect and forward data; these components are usually lightweight and are not resource intensive. No specific guidelines are required to run a forwarder on a virtual machine.

Indexers are responsible for storing and retrieving the data from disk. Because of this, CPU and disk I/O are the most important considerations.

Search heads search for information across indexers and are usually both CPU and memory intensive.

Budgeting system resources and bandwidth to enable search and index performance depends on the total volume of data being indexed and the number of active concurrent searches (scheduled or other) at any time.

In addition to rapidly writing data to disk, indexers do much of the work involved in running searches: reading data off disk, decompressing it, extracting knowledge and reporting. As a result, when scaling up data volumes, additional indexers should be added. These indexers will help handle larger volumes of data, reduce contention for resources during searches and accelerate search performance.

Performance Within VMware Environments

While Splunk software performs fastest when deployed directly on to bare-metal hardware, using Splunk on virtual equipment can and does deliver. There are several performance factors to consider when deploying Splunk software inside VMware virtual machines. These considerations are disk/storage, CPU and memory resources.

- **Disk/Storage:** Splunk indexers are usually CPU-and disk I/O-intensive, so disk exposed to these indexers within virtual machines should be capable of 1200+ random seeks per second. In virtual environments, with virtual machines moving from one type of storage to another, there is less control or guarantee over the type of disk that Splunk virtual machines can access. Shared storage diminishes the benefit of having multiple indexers since all indexers usually participate in the same Splunk searches. It is strongly encouraged that indexers have their own segregated storage.
- **CPU:** Since Splunk search heads and indexers are CPU intensive, sharing CPU with other virtual machines running on the same host

can result in high wait times, which might negatively impact Splunk performance. Splunk indexer and search head virtual machines should have 100% of the vCPU reserved to ensure good performance.

- **Memory:** Memory is critical for Splunk search heads and indexers. Splunk virtual machines should have reserved memory available to them. VMware hosts running Splunk Enterprise should not be configured with memory overcommit, as overcommitting memory will likely result in poor performance due to ballooning and/or page swapping to the hard disk.

Deployment Best Practices

No matter how well you provision your virtual machine, Splunk performance may still be affected by the quality of the underlying hardware.
All performance guidelines use the following reference server configuration.

Physical Host System Reference

A reference machine should contain similar or better hardware configuration as follows:

- Intel server class x86-64-bit chip architecture
- Minimum 2 CPU, 8 core per CPU, >=2Ghz per core (16 cores total without Hyperthreading[1], leaving at least two cores available for the hypervisor)
- Minimum 16 GB of RAM (ensure you leave some RAM unprovisioned for the hypervisor)
- RAID 1+0 is strongly preferred
- Minimum 1200 random seeks per second disk performance
- Standard 1Gb+ Ethernet NICs. Keep management and VMotion traffic on separate networks

Virtual Machine System Requirements

The reference configuration for a virtual machine is as follows:

- 12 vCPU
- 12 GB of RAM
- Full reservations for vCPU and vRAM (**No** CPU and memory overcommit)
- Raw volumes may provide slight improvements over VMFS
- Minimum 1200 **random seek** operations per second disk performance (sustained)
- Use VMware Tools in the guest VM
- Use VMware Paravirtual SCSI (PVSCSI) controller
- Use VMXNET3 network adapter
- Provision virtual disks as Eager Zero Thick when not on an array that supports the appropriate VAAI primitives ("Write Same" and "ATS")
- NFS and iSCSI disks are not recommended due to higher latency and file locking issues. This is due to wide variances in protocol implementations, network architecture, and disk subsystems.

Splunk Single Server Installations (For Data Volumes up to 250GB of Data Per Day)

If you're indexing less than 250GB of data per day, you can deploy Splunk Enterprise as a standard 12-way virtual machine described above. This is normally sufficient for most 250GB/day installations. However, depending on the number of concurrent users, you can deploy additional Splunk instances, following Splunk Enterprise's distributed architecture to distribute search load across the Splunk infrastructure.

- Search performance: Splunk Enterprise limits the number of concurrent searches by the number of cores available to the search head servers. For a single server installation this would be: concurrent searches = 6 + (1 * #cores); thus with a 12 core search head, maximum concurrent searches = 18.

Installations with a large number of scheduled saved searches and/or a sufficient number of active search or dashboard users may require more concurrent search capability. Splunk recommends utilizing the Splunk Distributed Architecture for these type of installations.

Splunk Distributed Architecture (Greater Than 250GB/day and/or 18+ Concurrent Active Searches)

- The scale-out capabilities of Splunk software include auto load balancing and distributed search to run searches in parallel across indexers.

For greater volumes of data or to accommodate many concurrent searches, split the Splunk search and indexing functions and parallelize indexing. Dedicated indexers with their own dedicated storage can handle a larger indexing load since they are not performing as much of the search load as a search head does. See the performance recommendations in Splunk documentation for general guidelines.

Shared Storage Recommendations

- NAS or virtualized storage is strongly discouraged, as is anything with higher than local disk latencies.
- Automated storage tiering solutions are strongly discouraged, and will likely induce unacceptable I/O latency. Splunk has its own native temporal tiering that can be very advantageous for most search use-cases.
- Ensure any VMDK on VMFS disks are thick provisioned as Eager Zero Thick. Eager Zero Thick provisioning insures that additional I/O is not incurred when Splunk Enterprise writes to disk. Without Eager Zero Thick, every block that has not been previously initialized on disk will first incur a write to zero it, and then the write for the data. This is roughly a 20-30 percent penalty for write I/O until all blocks on the file system have been written to. This doesn't pertain to NFS datastores, but we advise against using them without testing (latency tends to be too great).
- To initiate eagerzerothick on an existing volume without overwriting the contents, run this from the ESXi console on the base VMDK for the volume (http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1011170):

  ```
  vmkfstools -k baseVolumeName.vmdk
  ```

  Note: If using a VAAI array supporting the "Write Same" and "ATS" primitives (can be measured by ESXTOP if they exist), these APIs offload the zeroing and locking operations to the array directly and do not require Eager Zero Thick VMFS volumes.
- Test disk I/O performance (random seeks) before implementing Splunk Enterprise.

VMware vMotion, Storage vMotion, DRS, HA

While not formally tested by Splunk, the resources needed for various vSphere migrations will have a performance impact on Splunk deployments within VMware environments. While they are being migrated, they do get quiesced by the hypervisor, so it is recommended that you do these migrations during off peak hours.

If DRS (Distributed Resource Scheduling) is used, either pin the Splunk VMs to specific hosts or ensure that anti-affinity rules are created so that indexers are not overloaded on a single host, or ensure there are enough resources in the cluster to handle reallocation.

- Note: if there is a hardware failure and HA restarts virtual machines running Splunk Enterprise, during the restart some data may not get collected or indexed (particularly UDP streaming data sources such as syslog).

VMware Snapshots

VMware Snapshots (non-VAAI based) are strongly discouraged on Splunk VMs. When a snapshot is created, all new writes to the file system are written to the snapshot file, in the same way as thin provisioned disks, which requires multiple writes per data write. Also, when removing/consolidating snapshots, they require I/O to move blocks from the snapshot into the main VMDK file, plus the writes for all incoming data. This will dramatically affect I/O performance while the snapshot is in existence. Use of array-based snapshotting is preferred, but should be tested for impact to Splunk performance.

Additional Considerations

- Fiber Channel/FCoE access to a dedicated LUN for each Splunk VM is strongly encouraged
- If using Linux, disable guest VM's I/O Scheduler

    - Add elevator=noop to the grub.conf to turn off I/O scheduler (VMware already has an I/O scheduler, so this reduces latency).
    - http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2011861
- For high availability and redundancy requirements, refer to the relevant section in our documentation for physical hardware (http://docs.splunk.com/Documentation/Splunk/latest/Deploy/Scaleyourdeployment)

Summary

As is expected with most virtualized high I/O applications, you should expect as much as 10 percent less performance when running Splunk Enterprise within virtual environments. However, there are many additional benefits to consider. Virtualization offers better resource sharing and utilization, includes HA capabilities, makes provisioning and management an easier exercise, and may support a corporate virtualization mandate.

For best performance, put full reservations on CPU and memory, provision Eager Zero Thick VMDKs, and turn off snapshotting for virtual machines running Splunk Enterprise. Disk quality is also critical to Splunk performance—make sure you are using the best disk available. And to keep up with increasing data volumes, scale your deployment by adding additional Splunk indexers.

1. Hyperthreading has shown to add marginal benefits to Splunk operations, and therefore should not be considered in core calculations.