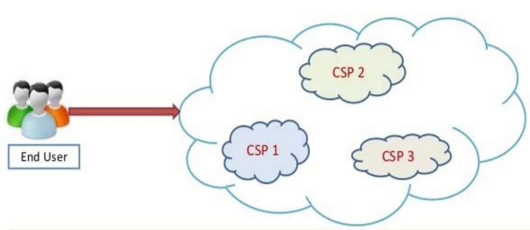
Cloud Federation

Cloud federation: definition

"A federated cloud (also called cloud federation) is the deployment and management of multiple external and internal cloud computing services to match business needs. A federation is the union of several smaller parts that perform a common action."

http://whatis.techtarget.com/definition/federated-cloud-cloud-federation



Cloud Federation is the concept of bringing different services offered by various Providers under a single platform .

- **Capacity utilization**
- **Inter-operability**
- **Catalog of services**
- **Insight about providers and SLA's**

Cloud federation

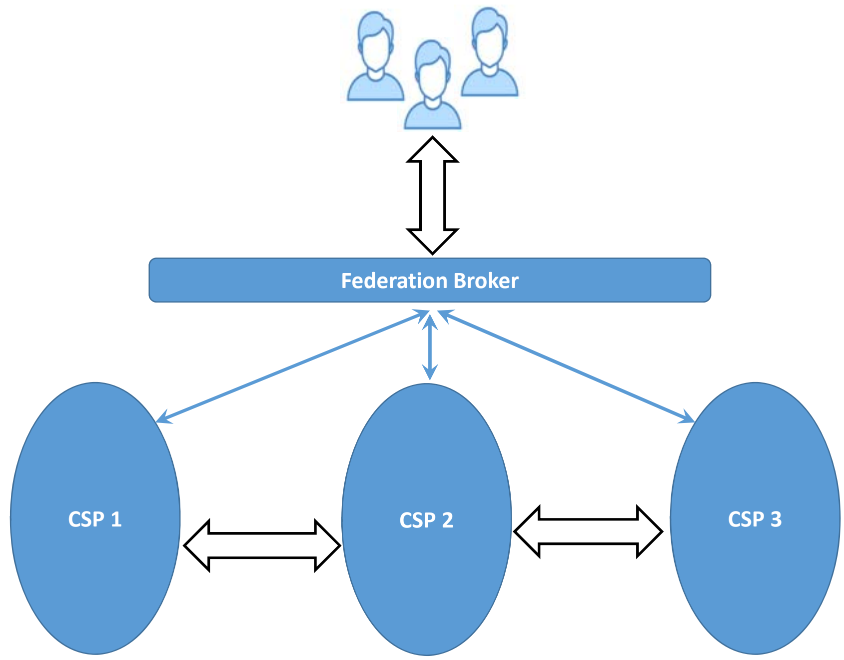
- **Different CSPs join together to form a federation**
- **Benefits:**
  - **Maximize resource utilization**
  - **Minimize power consumption**
  - **Load balancing**
  - **Global utility**
  - **Expand CSP’s global foot prints**

Cloud federation:

- To overcome the current limitations of cloud computing such as service interruptions, lack of interoperability and degradation of services.
- Many inter-cloud organizations have been proposed.
- Cloud federation is an example of an inter-cloud organization.

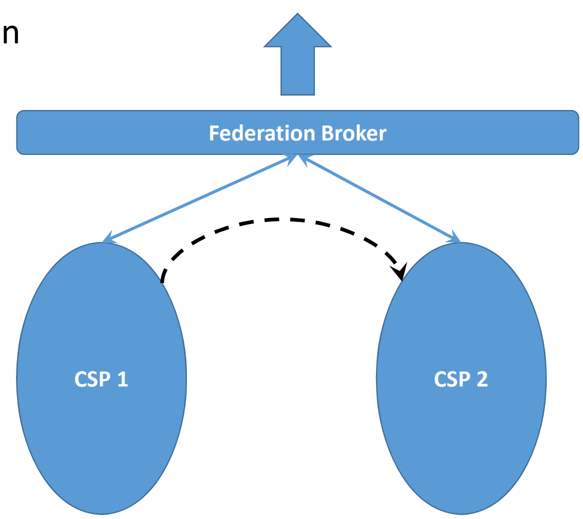
Characteristics

- It is a inter-cloud organization with voluntary characteristics.
- It should have maximum geographical separation.
- Well defined marketing system and regulated federal agreement.
- IT is an environment where multiple SP come together and share their resources.



Cloud federation: Operations

- Federation Placement
- Migration



Migration of VMs

Why Migration??

- **Load Balancing:** For fair distribution of workload among computing resources.
- **Maintenance:** For server maintenance VMs can be migrated transparently from one server to another.
- **Manage Operational Parameters:** To reduce operational parameters like power consumption, VMs can be consolidated on minimal number of servers. Under-utilized servers can be put on a low power mode to reduce power consumption.
- **Quality-of-Service violation:** When the service provider fails to meet the desired quality-of-services (QoS) a user can migrate his VM to another service provider.
- **Fault Tolerance:** In case of failure, VMs can be migrated from one data center to another where they can be executed

Types of Migration

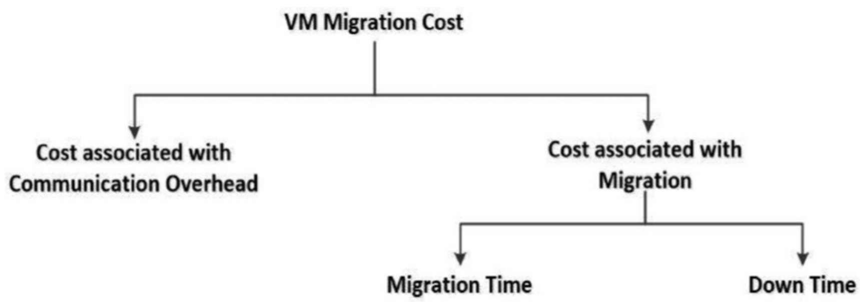
- **Cold or Non-Live Migration:** In case of cold migration the VM executing on the source machine is turned off or suspended during the migration process.
- **Hot or Live Migration:** In case of a hot or live migration the VM executing on the source machine continues to provide service during the migration process. In fact the target VM is not suspended during the migration process.

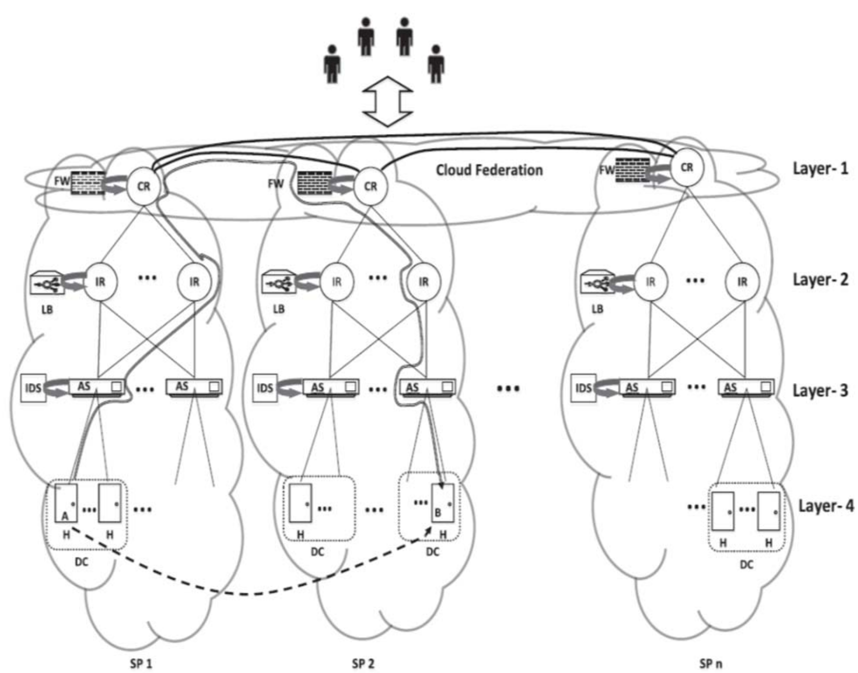
Why to prefer live migration??

- In case of non-live migration the VM providing the services remains suspended during the entire migration process. Hence for large sized VMs the service downtime might be very high.
- For real time applications non live migration can cause severe degradation in service quality which may not be tolerable.

When to Migrate?

- To remove a physical machine from service.
- To relieve load on congested hosts.





Concerns during live migration

- Minimize the **downtime.**
  - **Downtime refers to the total amount of time services remain unavailable to the users.**
- Minimize **total migration time.**
  - **Migration time refers to the total time taken to move a VM from the source host to the destination host. In fact it is the total time taken for the entire migration process.**
- Migration does not unnecessarily disrupt active services through resource contention (e.g., CPU, network bandwidth) with the migrating OS.

What to migrate?

- Memory
- Network
- Disk

Migrating memory

Possible 03 phases

- Push
  - The source VM continues running while certain pages are pushed across the network to the new destination. To ensure consistency, pages modified during this process must be re-sent.
- Stop-and-copy
  - The source VM is stopped, pages are copied across to the destination VM, then the new VM is started.
- Pull
  - The new VM executes and, if it accesses a page that has not yet been copied, this page is faulted in (“pulled”) across the network from the source VM.

Migrating memory

pure stop-and-copy

- This has advantages in terms of simplicity but means that both downtime and total migration time are proportional to the amount of physical memory allocated to the VM.
- This can lead to an unacceptable outage if the VM is running a live service.

Phases of Live Migration

i. **Pre-Copy Phase:** It is carried out over several rounds. The VM continues to execute at the source, while its memory is copied to the destination.

ii. **Pre-copy Termination Phase:** Stopping criteria of Pre-Copy phase takes one of the following thresholds into account: (i) The number of rounds exceeds a threshold. (ii) The total memory transmitted exceeds a threshold. (iii) The number of dirtied pages in the previous round drops below a threshold.

iii. **Stop-and-Copy Phase:** In this phase, execution of the VM to be migrated is suspended at the source. Then, the remaining dirty pages and, state of the CPU is copied to the destination host, where the execution of VM is resumed.

Phases of Iterative Pre Copy Live Memory Migration

- **Pre-copy Phase:** This phase is carried out over several rounds. The VM continues to execute at the source host, while its memory is copied to the destination host. Active pages of the VM to be migrated are copied iteratively in each round. During the copying process some active page might get dirtied at the source host, which are again resent in the subsequent rounds to ensure memory consistency.
- **Pre copy-termination phase**: Stopping criteria of Pre- Copy phase takes one of the following into account:
  - The number of rounds exceeds a threshold.
  - The total memory transmitted exceeds a threshold.
  - The number of dirtied pages in the previous round drops below a threshold.

Phases of Iterative Pre Copy Live Memory Migration

- **Stop-and-Copy Phase:** In this phase, execution of the VM to be migrated is suspended at the source. Then, the remaining dirty pages and, state of the CPU is copied to the destination host, where the execution of VM is resumed.
- **Restarting Phase:** Restart the VM on destination server.

Phases of Post-copy Live Memory Migration

- **Stop Phase**: Stop the source VM and copy the CPU state to the destination VM.
- **Restart Phase**: Restart the destination VM.
- **On-demand Copy:** Copy the VM memory according to the demand.

Note: In the post-copy strategy, when the VM is restarted, the VM memory is empty. If the VM tries to access a memory page that has not yet been copied, this memory page needs to be brought from the source VM. However, most of the time, some memory pages will not be used, so we only need to copy the VM memory according to the demand.

Mathematical Model

- Let T_{mig} be the total migration time.
- Let T_{down} be the total down time.
- For non live migration of a single VM the migration time T_{mig} can be calculated as follows:

$$T_{mig} = V_m / R .$$

where V_m is the size i.e. memory of the VM and R is the transmission rate.

- In fact the down time is same as the migration time in this case because the services of the VM is suspended during the entire migration process.

$$T_{down} = T_{mig}$$

NB: The transmission rate remains fixed for the entire duration of migration.

Mathematical Model

- Let **n** represent the total number of iterations in the pre copy cycle.
- Let $\mathbf{T_{i,j}}$ represents the total time that the $\mathbf{j^{th}}$ iteration transmits the $\mathbf{i^{th}}$ virtual machine's memory.
- V_m be the memory of a VM.
- V_{th} be the threshold for stopping the iterations.
- n_{max} be the maximum number of iterations.
- R is the transmission rate.
- r=(P*D)/R, where P is page size and D is the dirtying rate.
- T_{res} denotes the time taken to restart the VM on the destination server.

Mathematical Model

- In the pre-copy migration mechanism, the VMs memory can be migrated iteratively.
- We can compute the total migration time $T_{i,mig}$ of the i^{th} VM as follows.

$$T_{i,mig} = \sum_{j=0}^{n}(T_{i,j}) = \frac{V_m}{R}\left(\frac{1-r^{n+1}}{1-r}\right)+T_{res}$$

$$T_{i,down} = r^n\left(\frac{V_m}{R}\right)+T_{res}$$

Detailed Explanation

- Round 0 : $t_0 = \frac{V_m}{R}$
- Round 1: $t_1 = \frac{(P*D)}{R} * t_0 = \frac{(P*D)}{R} * \frac{V_m}{R} = r*(\frac{V_m}{R})$
- Round 2: $t_2 = \frac{(P*D)}{R} * t_1 = \frac{(P*D)}{R} * (r*\frac{V_m}{R}) = r^2(\frac{V_m}{R})$
- Round 3: $t_3 = \frac{(P*D)}{R} * t_2 = \frac{(P*D)}{R} * (r^2\frac{V_m}{R}) = r^3(\frac{V_m}{R})$

.

.

.

- Round **n−1**: $t_{n-1} = \frac{(P*D)}{R} * t_{n-2} = \frac{(P*D)}{R} * (r^{n-2} * \frac{V_m}{R}) = r^{n-1} * (\frac{V_m}{R})$
- Round **n** (Stop and Copy): $t_n = \frac{(P*D)}{R} * t_{n-1} = \frac{(P*D)}{R}(r^{n-1} * \frac{V_m}{R})$

  $= r^n(\frac{V_m}{R})$
- $T = t_0+t_1+\ldots+t_{n-1} + t_n$

  $= \frac{V_m}{R}(1+r+r^2 +r^3 + \ldots +r^{n-1} + r^n)$

  $= \frac{V_m}{R}\left(\frac{1-r^{n+1}}{1-r}\right).$

Number of rounds (n)

- Volume of dirty data to be transferred in round j = $r^j.V_m$
- $r^j.V_m < V_{th}$
- => $j = \lceil log r \frac{V_{th}}{V_m} \rceil$
- n= min($\lceil log r \frac{V_{th}}{V_m} \rceil$,$n_{max}$)

Migration of multiple VMs

- Generally multiple VMs are migrated from a source host to the destination host.
- There are different strategies for migration multiple VMs:
  - **Serial Migration.**
  - **Parallel migration.**

Serial Migration

- In case of serial migration of ‘m’ correlated VMs of same type the procedure is as follows:
- The first VM that is selected to be migrated executes its pre-copy cycle and the other (m-1) VMs continue to provide services.
- As soon as the first VM enters into the stop and copy phase the remaining (m-1) VMs are suspended and are copied after the first VM completes its stop and copy phase.
- The reason for stopping the remaining (m-1) VMs is to stop those VMs from dirtying memory.
- The assumption here is that each VM that is copied at full transmission rate (R).
- The downtime for the serial migration includes the stop and copy phase of the first VM, the migration time for the (m-1) VMs and the time to resume the VMs at the destination host.

Serial Migration of VMs.

- Let there are ‘m’ VMs that are to be migrated serially.
- The migration time and downtime for serial migration strategy can be calculated as follows:

$$T^{s}_{mig} = \sum_{i=1}^{m}(T_{i,mig}) = \frac{m.V_m}{R}\left(\frac{1-r^{n+1}}{1-r}\right) + T_{res}$$

$$T^{s}_{down} = \frac{V_m}{R}.\, r^n +$$

$$(m-1)\frac{V_m}{R}\left(\frac{1-r^{n+1}}{1-r}\right) + T_{res}$$

Parallel Migration

- The difference between parallel and serial migration is that all 'm' VMs start their pre-copy cycles simultaneously.
- In fact each VM shares (R/m) of the transmission capacity.
- As the VM sizes are same and transmission rates are same the VMs begin the stop and copy phase as the same time and they end the stop and copy phase also at the same time.
- Since the stop and copy phase is executed in parallel and they consume the same amount of time the downtime is in fact equivalent to the time taken by the stop and copy phase for any VM added to the time taken to resume the VMs at the destination host.

Parallel migration of VMs

- $T^{p}_{mig} = \sum_{i=1}^{m}(T_{i,mig}) = \frac{m.V_m}{R}\left(\frac{1-mr^{n(p)+1}}{1-mr}\right)+T_{res}$
- $T^{p}_{down} = \frac{m.V_m}{R}.\ (m.r)^{n(p)} + T_{res}$

Thank you.