Amirkabir University of Technology
(Tehran Polytechnic)

Cloud Computing

OpenStack Nova Architecture

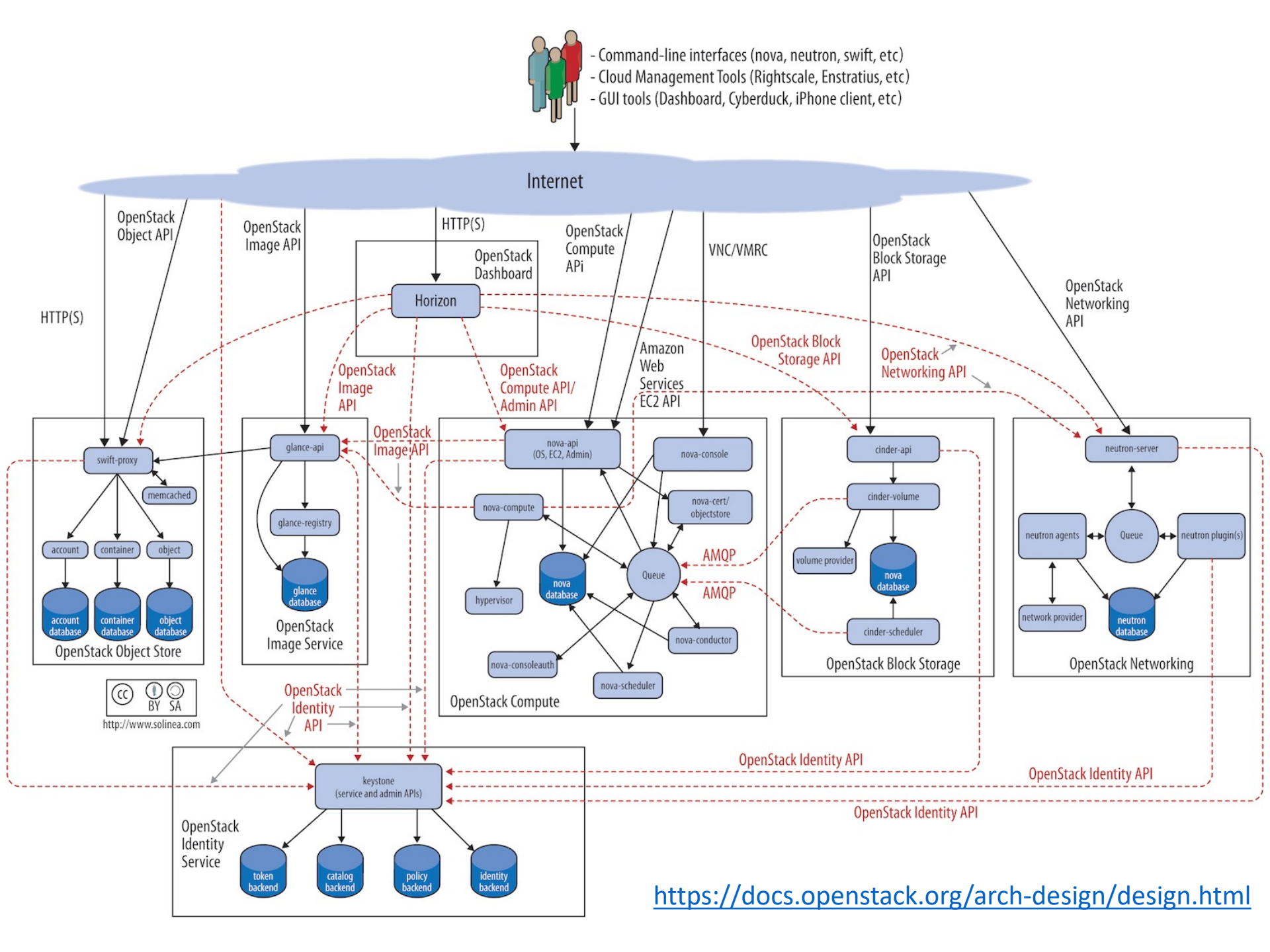
Seyyed Ahmad Javadi
sajavadi@aut.ac.ir

Spring 2024

https://www.slideshare.net/HaimAteya/an-intrudction-to-openstack-2017

https://docs.openstack.org/security-guide/introduction/introduction-to-openstack.html

Overview



https://docs.openstack.org/arch-design/design.html

Controller node components

glance_api			
horizon			
heat_engine	heat_api_cfn	heat_api	
neutron_server			
ovn_northd	ovn_sb_db	ovn_nb_db	
nova_api	nova_scheduler	nova_novncproxy	nova_conductor
cinder_scheduler	cinder_api		
keystone	keystone_fernet	keystone_ssh	
placement_api	etcd	rabbitmq	memcached
keepalived	haproxy	cron	kolla_toolbox
chrony			

No need to memorize the table for the exam

Compute node components

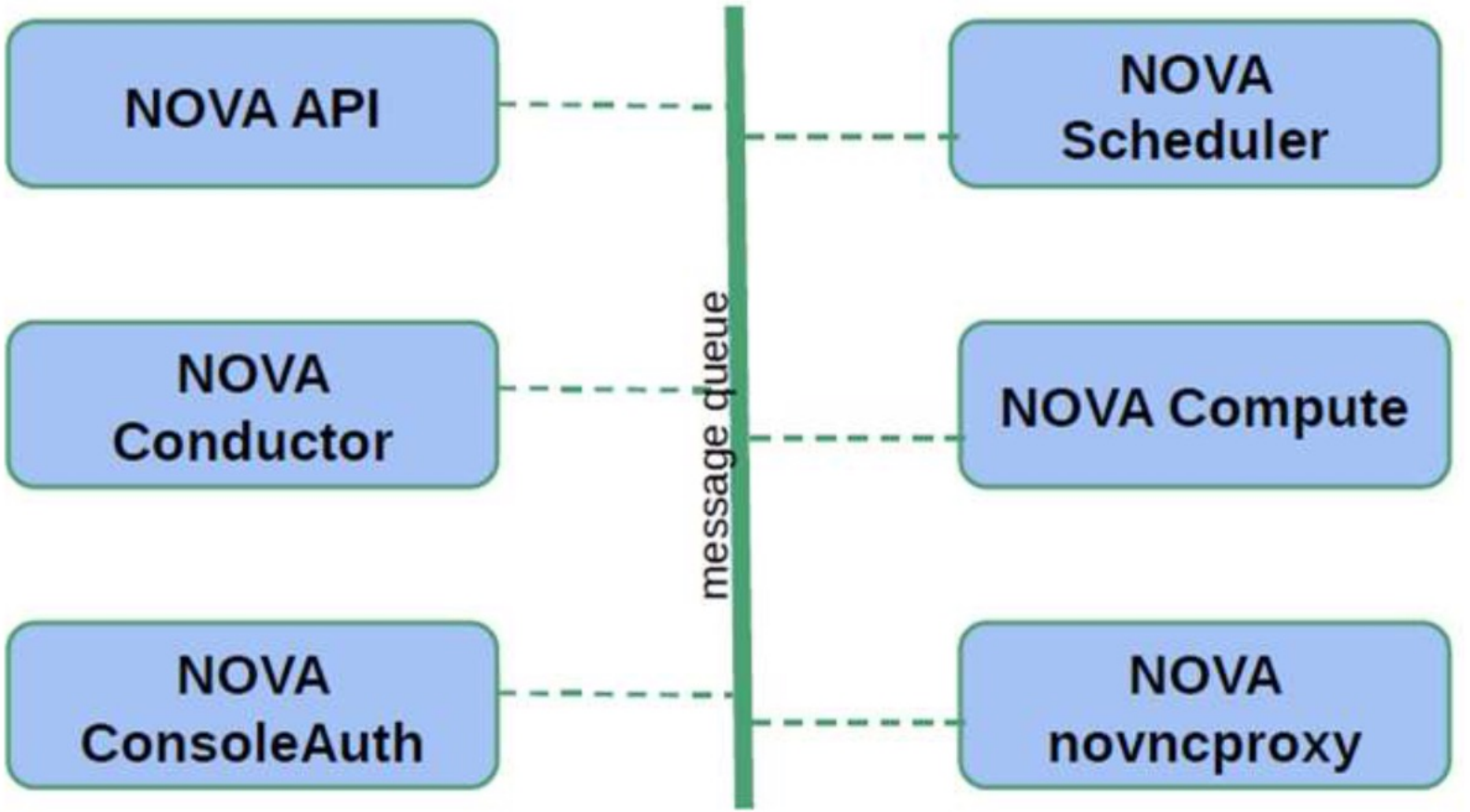
openvswitch_vswitchd	openvswitch_db	
nova_compute	nova_libvirt	nova_ssh
cinder_backup	cinder_volume	
tgtd	iscsid	
neutron_ovn_metadata_agent	ovn_controller	
cron	kolla_toolbox	chrony

No need to memorize the table for the exam

Nova

- Provided compute as service
- The main part of an IaaS system
- It is designed to manage and automate pools of computer resources
- Compute's architecture is designed to scale horizontally

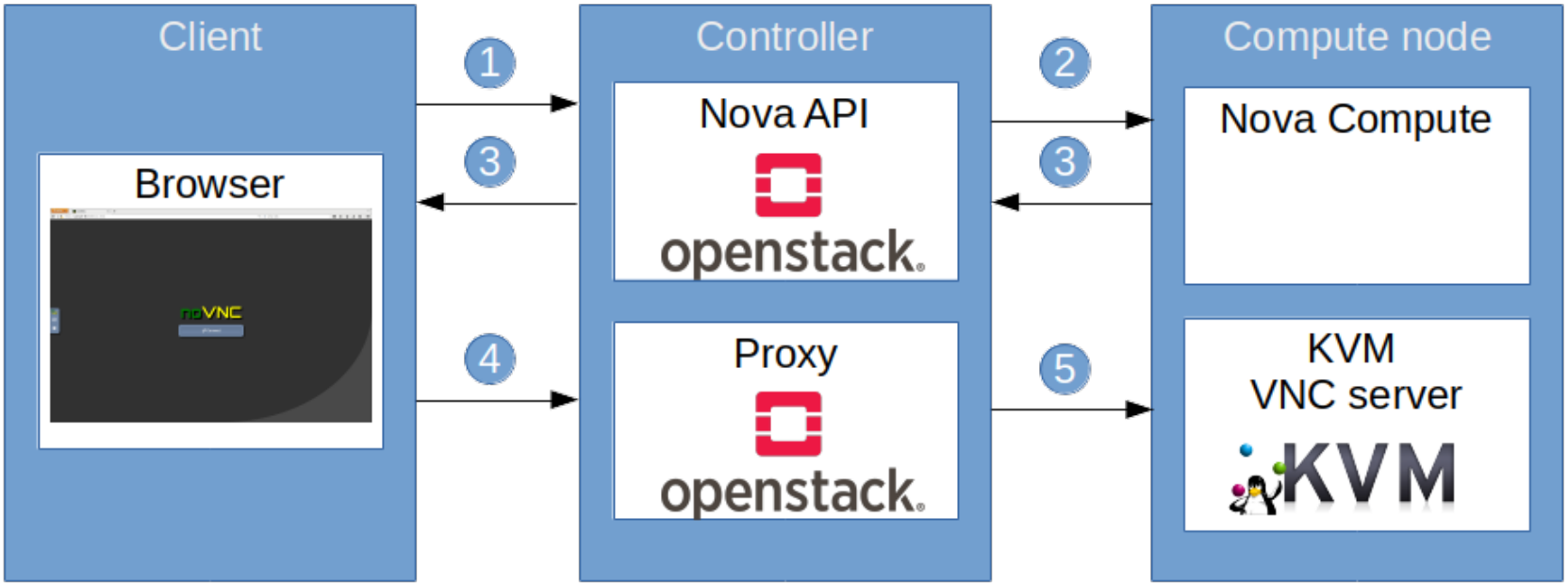
Nova Components



Nova Components

- Nova-conductor
  - Provides database-access support for Compute nodes
- Nova-consoleauth
  - Handles console authentication
- Nova-novncproxy
  - Provides a VNC proxy for browsers

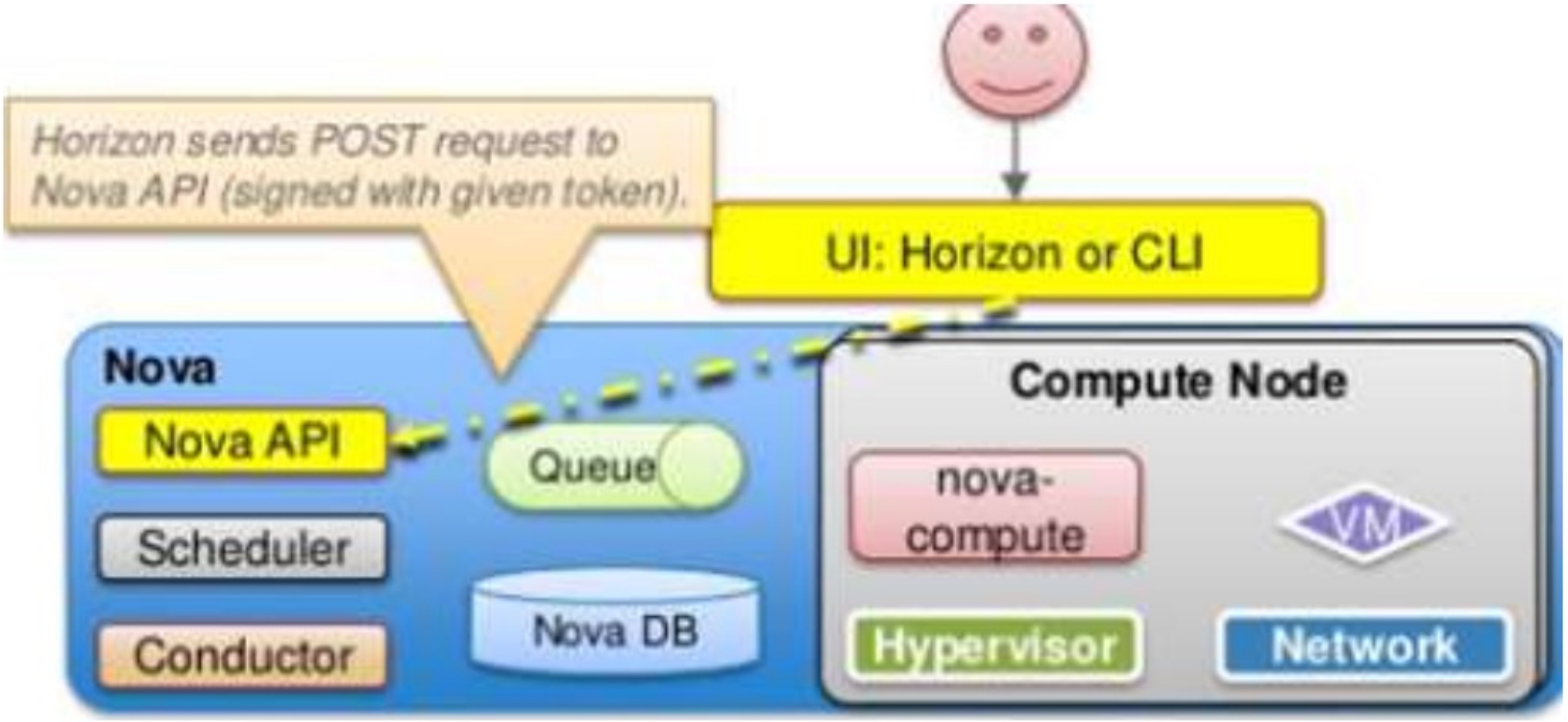
The Nova VNC proxy



https://leftasexercise.com/2020/02/14/openstack-nova-installation-and-overview/

NOVA API

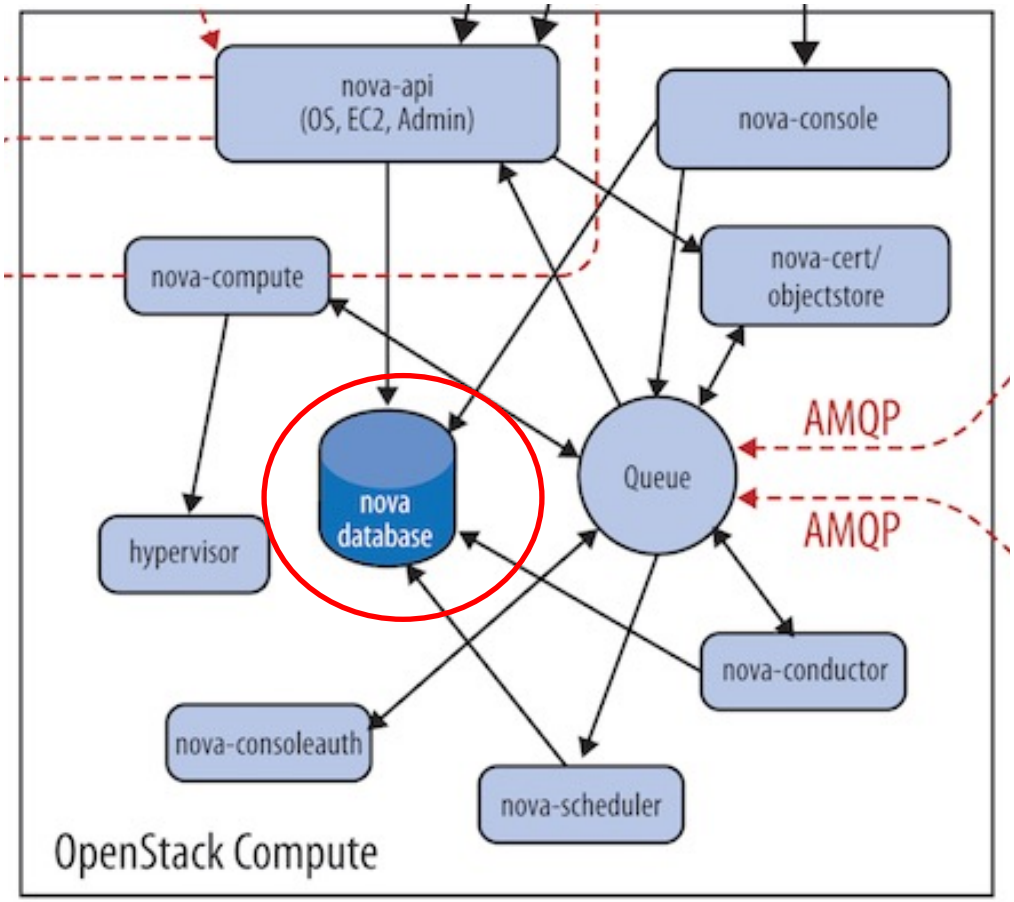
- NOVA-API is responsible to provide an API for users and services to interact with NOVA



On Compute Node

- There is a periodic task (Resource Tracker), which collects host information.

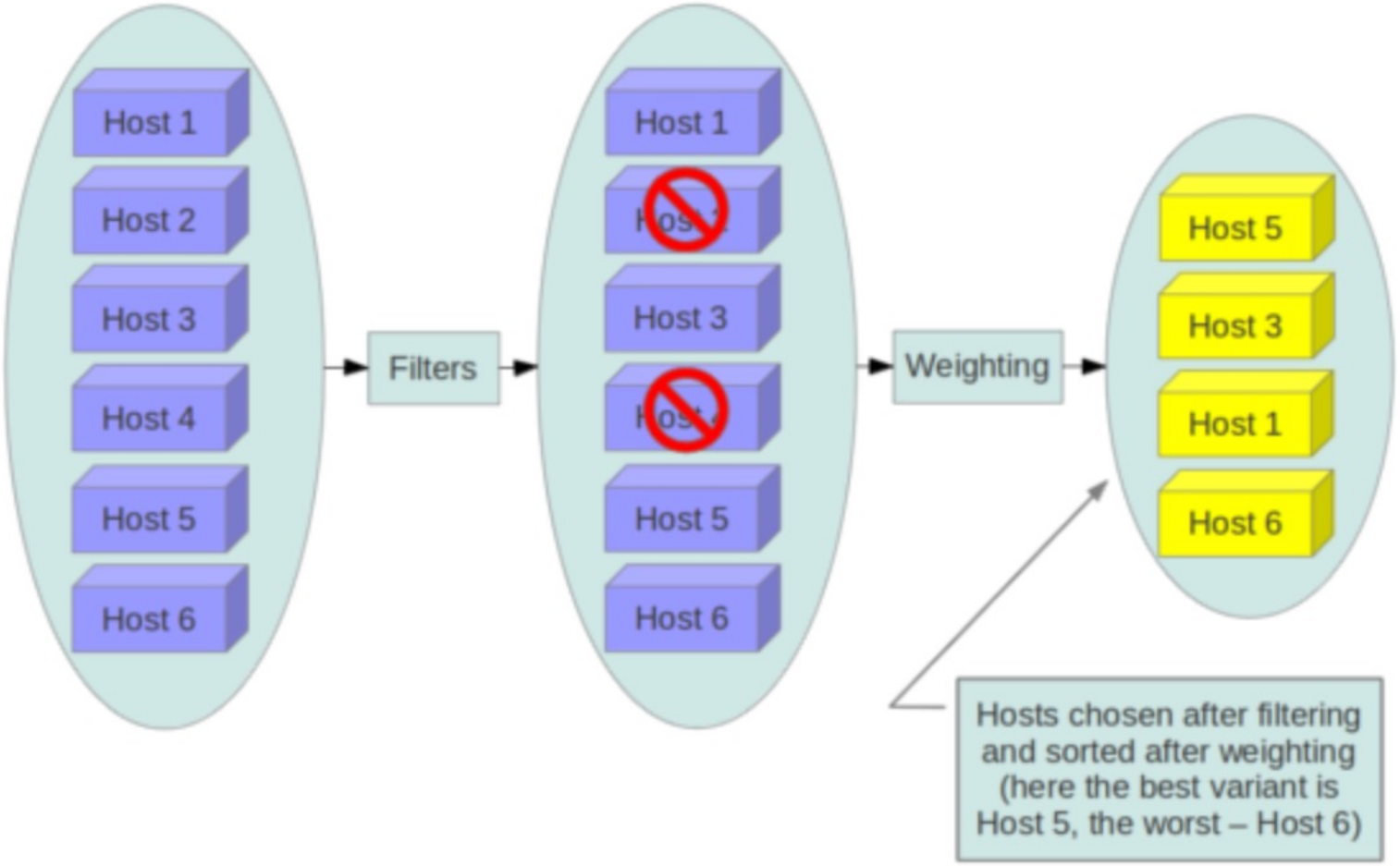
- This information is then stored to ***database***



On Controller Node

- Request from nova API reaches conductor
- Conductor interacts with the scheduler
- Scheduler ***uses filters*** to identify the best node
  - From the information stored in ***database***
- Selected host information is sent back to conductor
- Conductor uses the compute queue and directs it to selected host
- The compute node then launches the instance

Filters and Weights



Some Common Filters

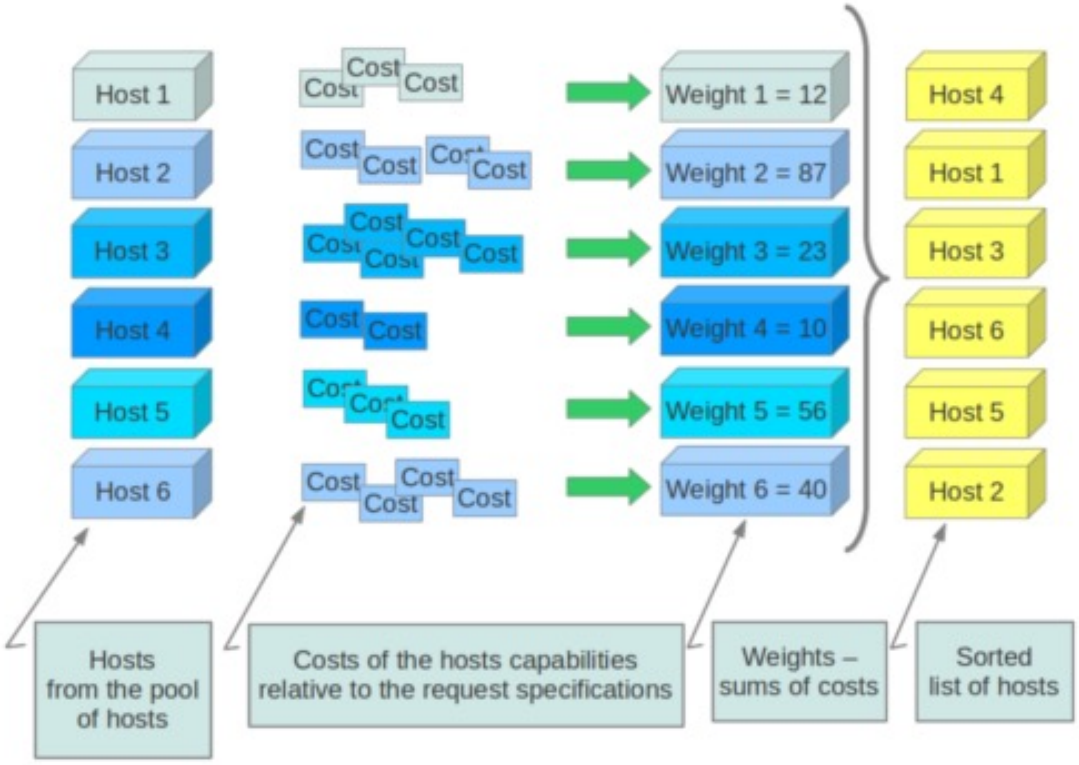
- **AvailabilityZoneFilter**
  - Return hosts where node_availability_zone name is the same as the one requested
- **RamFilter**
  - Return hosts where (free_ram * ram_allocation_ration) is greater than requested ram.
- **ComputerFilter**
  - Return hosts where asked instance_type (with extra_specs) match capabilities

Some Common Filters (cont.)

- **DiskFilter**
  - Returns hosts with sufficient disk space available for root and ephemeral storage.
- **RetryFilter**
  - Filters out hosts that have already been attempted for scheduling purposes.

Weights

- Scheduler applies cost function on each host & calculates the weight.



https://docs.openstack.org/nova/latest/admin/scheduling.html

Some Possible Cost Functions

- Considering free RAM among filtered hosts.
  - Highest free RAM wins.
- Considering least workload (e.g., IO ops) among filtered hosts.
- Can consider any specific metric we want to consider in a similar fashion.
  - Can be enabled from configuration file.

```
weight = w1_multiplier * norm(w1) + w2_multiplier * norm(w2) + ...
```

Weights (cont.)

- **RAMWeigher**
  - Compute weight based on available RAM on the compute node. Sort with the largest weight winning.
- **CPUWeigher**
  - Compute weight based on available vCPUs om the compute node. Sort with the largest weight winning.
- **DiskWeigher**
  - Hosts are weighted and sorted by free disk space with the largest weight wininng.

Weights (cont.)

- **MetricWeigher**
  - This weigher can compute the weight based on the compute node host's various metrics.
  - The to-be weighed metrics and their weighing ration are specified in the configuration file as the followings:

```
[metrics]
weight_setting = name1=1.0, name2=-1.0
```

General Cost Function

```
weight = w1_multiplier * norm(w1) + w2_multiplier * norm(w2) + ...
```

Metric	Range
CPU utilization	(0, 100) usage percentage
Outbound network traffic	(0, 10^9) byte per second

Least Loaded Server with No Normalization

Weight (Load) = 1 * (CPU utilization) + 1* (Outbound network traffic)

	CPU utilization	Outbound network traffic
Host1	95	100000
Host2	10	100090

Least Loaded Server Without Normalization

Weight (Load) = 1 * (CPU utilization) + 1* (Outbound network traffic)

	Weight
Host1	(1 * 95) + (1 * 100000) = 100095 ✓
Host2	(1*10) + (1* 100090) = 100100

Host1 is selected!

Not good ☹

Min-Max Normalization

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

As we discussed in the class, the min and max come from the nature of data

Getting Back to the Previous Example

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

	CPU utilization	Outbound network traffic
Host1	95	100000
Host2	10	100090

	CPU utilization	Outbound network traffic
Host1	(95-0)/(100-0)=0.95	(100000-0)/(10^9-0)=0.0001
Host2	(10-0)/(100-0)=0.1	(100090-0)/(10^9-0)=0.00010009

Least Loaded Server with Normalization

Weight (Load) = 1 * norm(CPU utilization) +

1* norm(Outbound network traffic)

	Weight
Host1	(1 * 0.95) + (1 * 0.0001) = 0.9501
Host2	(1*0.1) + (1* 0.00010009) = 0.10010009✓

Host2 is selected!

Good job :)

نمونه سوال امتحانی

۵) سوال OpenStack (۲۰ نمره)

الف) به انتخاب خودتان، پنج مولفه اصلی openstack را نام ببرید و وظیفه هر کدام را کوتاه بنویسید. (۱۰ نمره)

ب) فرض کنید openstack رو طوری پیکربندی کردید که برای انتخاب میزبان یک ماشین مجازی از فیلتر RamFilter استفاده نموده و برای وزن‌دهی، از ترکیب دو شاخص درصد استفاده از CPU و مقدار ترافیک خروجی شبکه میزبان (برحسب کیلو بایت) استفاده می‌کند. در این پیکربندی ram_allocation_ration برابر با **۱.۵** قرار داده شده و برای شاخص‌های درصد استفاده از CPU و مقدار ترافیک خروجی وزنی برابر با **۰.۵** در نظر گرفته شده است.

Host name	Zone	Free Memory (GB)	CPU usage (%)	Outbound network traffics (KB)
Server1	A	32	98	1000
Server2	A	8	30	5000
Server3	A	12.5	40	1200
Server4	B	10	80	1030
Server5	B	8	20	500
Server6	B	16	10	1100

درخواستی برای ایجاد ماشین مجازی با 16GB حافظه اصلی و ۴ هسته CPU در Zone B اماده است. مشخص کنید از بین میزبان‌های جدول بالا کدام میزبان‌ها از فیلتر عبور می‌کنند و وزن هر کدام چقدر خواهد بود و کدام یک برای ایجاد ماشین مجازی انتخاب می‌شود. (۱۰ نمره)

نکته مهم در مورد پاسخ سوال قبلی

- همانطور که در کلاس دیدیم، برای نرمال‌سازی min-max و تعیین مینیمم و ماکزیمم، بایستی به **بازه ممکن** برای هر ستون رجوع کرد و مثلا استفاده از ۵۰۰ به عنوان مینیمم ستون ترافیک خروجی شبکه و ۵۰۰۰ به عنوان ماکزیمم این ستون، به پاسخ صحیحی منتج **نمی‌شود**. در اینجا بایستی منیمم و ماکزیمم برای این ستون به شکل بازه داده می‌شد. به عنوان مثال باید بازه [0, 125000KB] داده می‌شد و شما مینیمم این ستون را 0 و ماکزیمم ان را 125000 در نظر می‌گرفتید. از طرفی برای درصد استفاده از پردازنده، بایستی بازه [0, 100] داده می‌شد. از طرفی Zone B در این سوال، نکته انحرافی است. این تیپ سوال رو جدی بگیرید :)