CERN openlab

HUAWEI OPENSTACK CLOUD

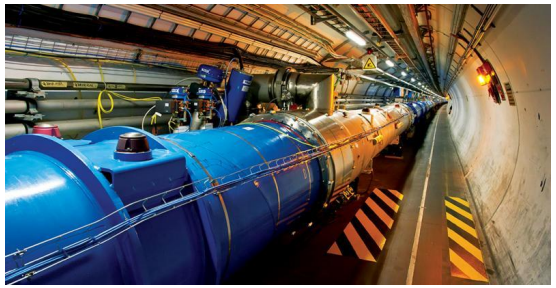
THEODOROS TSIOUTSIAS
theodoros.tsioutsias@cern.ch
SURYA SEETHARAMAN
surya.seetharaman@cern.ch

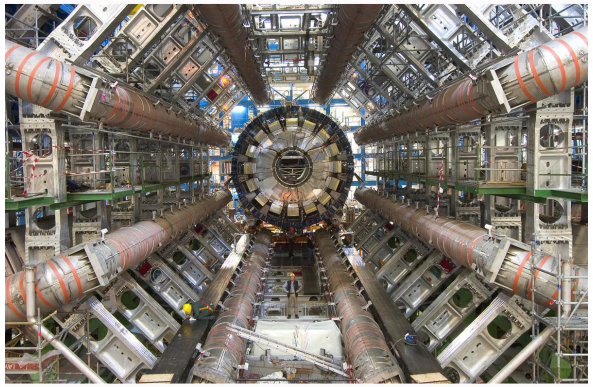
24/01/2019

PROJECTS

Currently there are two projects

- *Migration from Cells_v1 to Cells_v2*
  - Surya Seetharaman
- *Preemptible Instances*
  - Theodoros Tsioutsias





Migration from CellsV1 to CellsV2

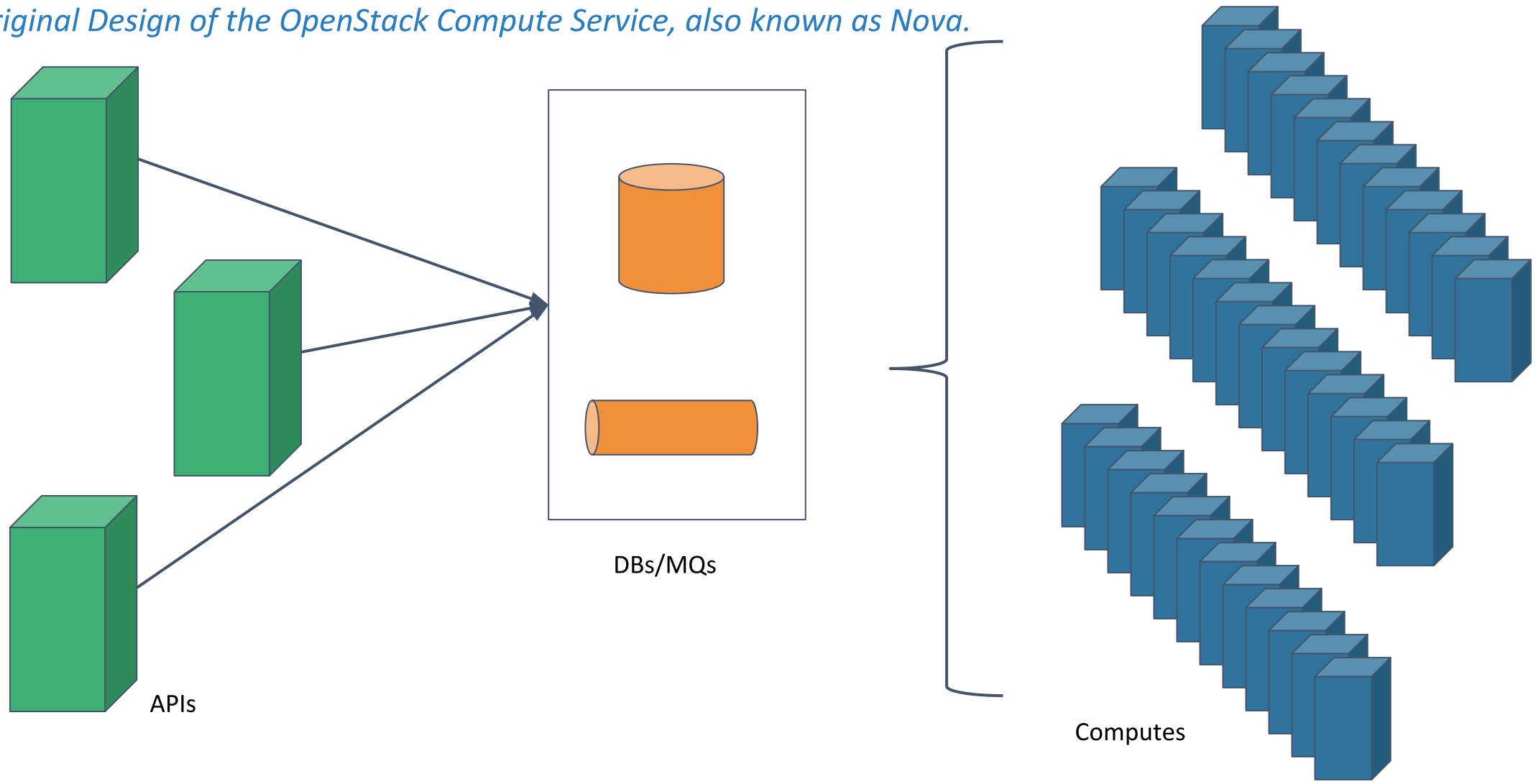
Surya Seetharaman

Openstack at CERN

Compute Statistics from Scaling Point of View

Nova Without Scaling

Original Design of the OpenStack Compute Service, also known as Nova.



Inspired from https://www.openstack.org/videos/boston-2017/scaling-nova-how-cellsv2-affects-your-deployment

How do we Scale... ?

Compute Service is scaled using Nova Cells

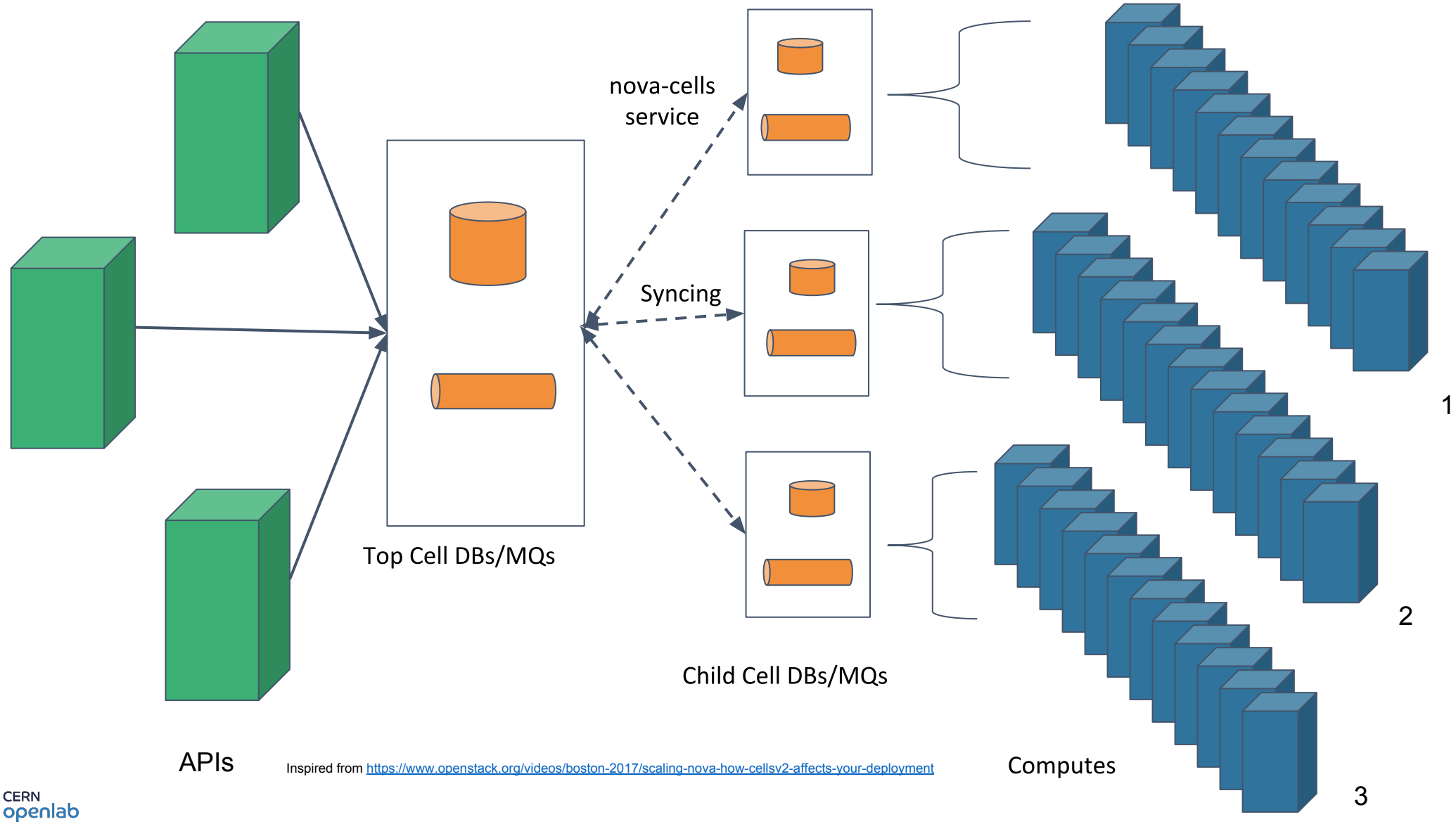
➔ What are Nova Cells ?

1. A functionality to scale the OpenStack compute cloud.
2. Hosts in the compute cloud are divided into groups i.e cells.
3. Each cell has its own DB and MQ.
4. Cells are in a tree-like structure having a top ‘api’ cell and child cells.

➔ Why do we use Cells ?

1. A strategy for scaling.
2. Failure Isolation.
3. Provides Elasticity/Flexibility.

Nova with Cells-v1



Inspired from https://www.openstack.org/videos/boston-2017/scaling-nova-how-cellsv2-affects-your-deployment

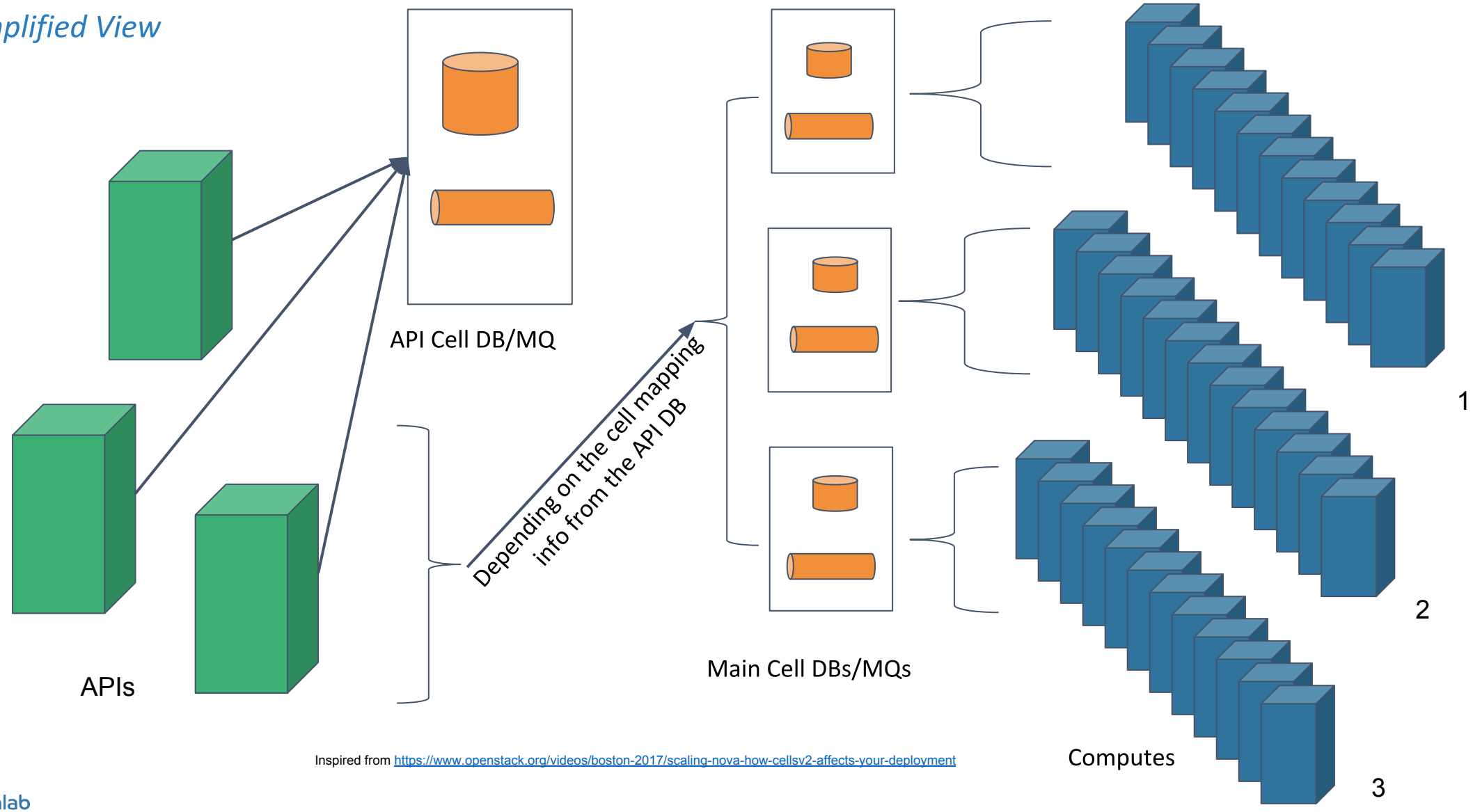
Limitations of Cells-v1

Upstream development has stopped for v1.

- ➔ Syncing/Python Replication is annoying.
- ➔ Top level cell scheduler has limitations.
  - ◆ It’s the story of grass on the opposite side seeming more green.
- ➔ Race Conditions since its not built-in in a core manner.
- ➔ Two different code paths for nova and nova with cells.
- ➔ No path for upgrade from no cells to cells-v1.
- ➔ Data duplication in the parent cell.

Nova with Cells-v2

Simplified View



Inspired from https://www.openstack.org/videos/boston-2017/scaling-nova-how-cellsv2-affects-your-deployment

Migration to Cells-v2

Completely integrated with the whole nova code - it's in the main code path

- ➔ Basically overcomes all the issues with Cells-v1.
- ➔ Global data moves to the API DB (all data in one place, no duplication).
- ➔ No need of syncing, no more upcalls.
- ➔ Single scheduler - knows about all nodes - pre-claims resources.
- ➔ DB/MQ info of each cell in API DB - no more "nova-cells" service.
- ➔ Most importantly, this is the maintained cells version upstream.
- ➔ All deployments are single cell-v2 or multi-cell.

On 25th April 2018 CERN moved to Cells-v2

One top level API cell and ~74 child cells; each having ~200 computes.

CERN Cells - Current Status

- ➔ CERN has been successfully running CellsV2 for the past nine months.
- ➔ Project involved integration with upstream nova team
- ➔ PTG, Summits, contributions
- ➔ involvement with development and pushing features needed for CERN
- ➔ preparing for the upgrade
- ➔ Dealing with post-upgrade issues:
  - ◆ Improving performance, flexibility and resilience of cellsv2

- ➔ The Project Goals have been achieved.

References:

1. http://stackalytics.com/?company=cern&user_id=tssurya&release=all&metric=commits

Preemptible Instances

THEODOROS TSIOUTSIAS

Introduction

Quotas

Cloud Computing gives the illusion of infinite capacity

- Quota System:

1. Sets limits on resources
2. Ensures everyone makes use of their fair share of the resources

- Operators use **quotas** per project to:

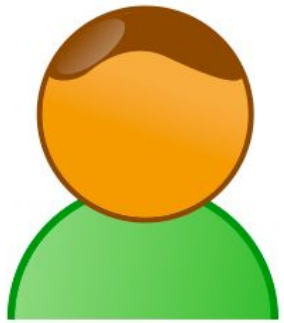
1. Prevent system capabilities from being exhausted without notification
2. Manage the resource allocations
3. Avoid “Over-committing” resources
4. Reserving Resources for operations with higher priority

Scenario 1/3

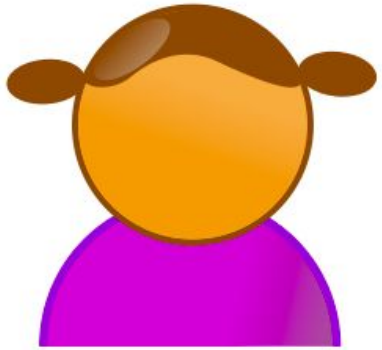
Micro view

- Alice has available resources
- Bob's quota is exhausted and he needs more computing power

Bob could tell Alice: "Please let me use the your idle resources."



Bob



Alice

Scenario 2/3

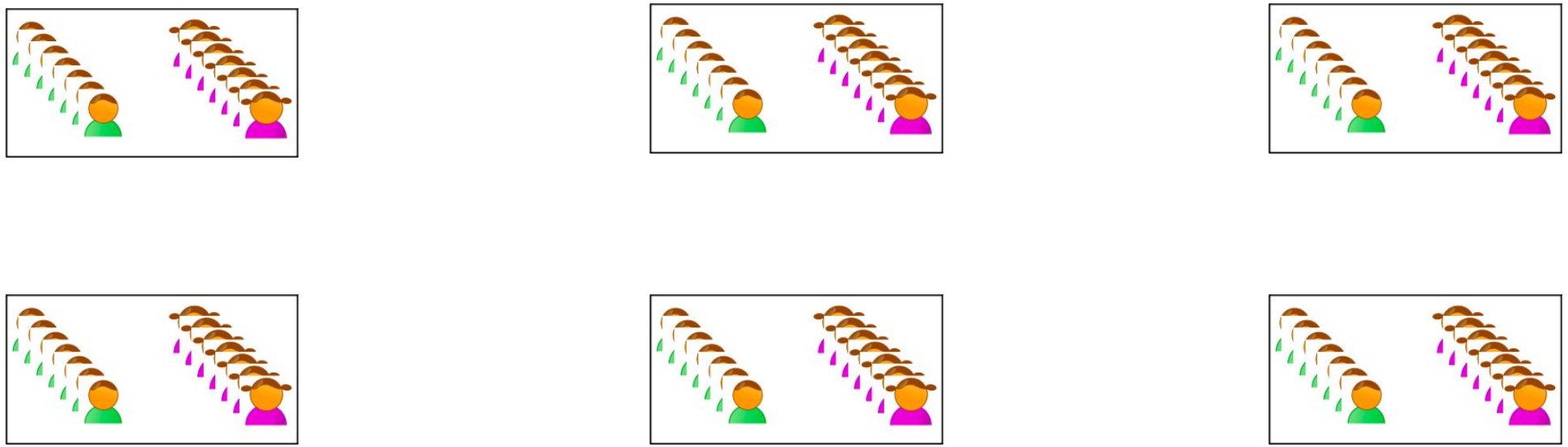
But what happens in a department?

- Imagine a now department in a company with tens of employees having the same issue

Scenario 3/3

Or even worse, in a big organization?

- The same race exists between different departments in an organization



Problem

Keeping track of the available resources

- As the organization grows, the amount of unused/idle resources may increase as well.
- Quotas are hard limits:

> Even if there are free resources, they cannot be allocated to a project whose quota is exceeded

- This leads to a reduction in cloud utilization:

> There are resources in idle state!

Solution

- Introducing the concept of Preemptible Instances:
  - created even after the quota for a project is exceeded
  - use idle resources
  - terminated as soon as the resources are needed for higher priority tasks

- The result of this:
  - handling the demand for extra resources
  - increasing the cloud utilization
  - maintaining the fair sharing of the infrastructure

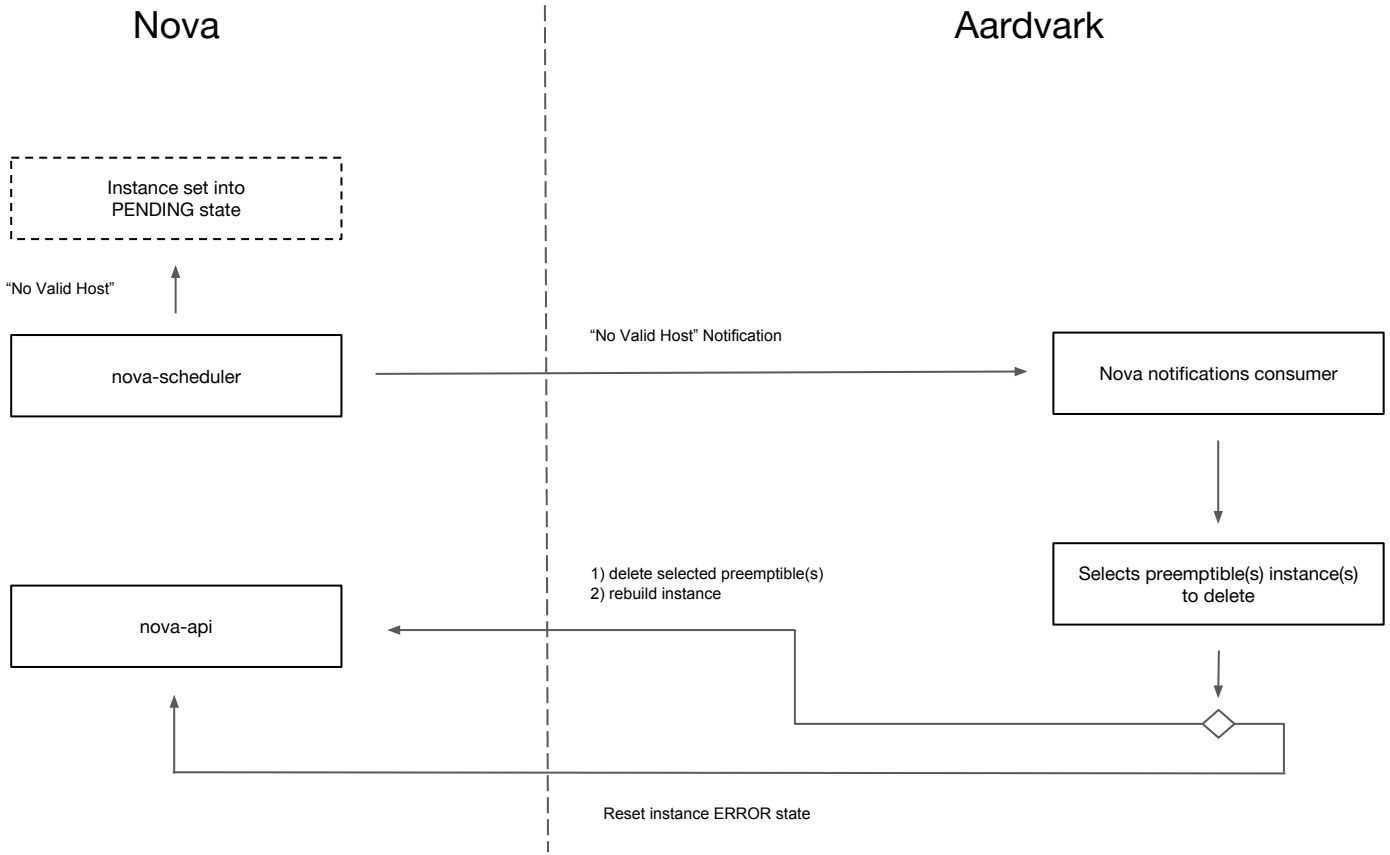
Openstack Preemptible Instances

Keeping things simple

- Used dedicated projects:
  - These projects have unlimited quota
  - Instances in these projects are preemptible

- Introduced a Reaper service:
  - Orchestrator for the preemptible instances
  - Applies strategies to free up the resources
  - Aardvark!

Workflow


Nova
Aardvark
Instance set into PENDING state
"No Valid Host"
nova-scheduler
"No Valid Host" Notification
Nova notifications consumer
Selects preemptible(s) instance(s) to delete
1) delete selected preemptible(s)
2) rebuild instance
nova-api
Reset instance ERROR state

Current Status

Upstream changes and CERN deployment

- Prototype Implementation was presented during the OpenStack Summit in Berlin
- We are planning to deploy Aardvark in CERN Cloud later this year

- Aardvark repo:
  - https://gitlab.cern.ch/ttsiouts/aardvark

QUESTIONS?