

What's new in Nova CellsV2?

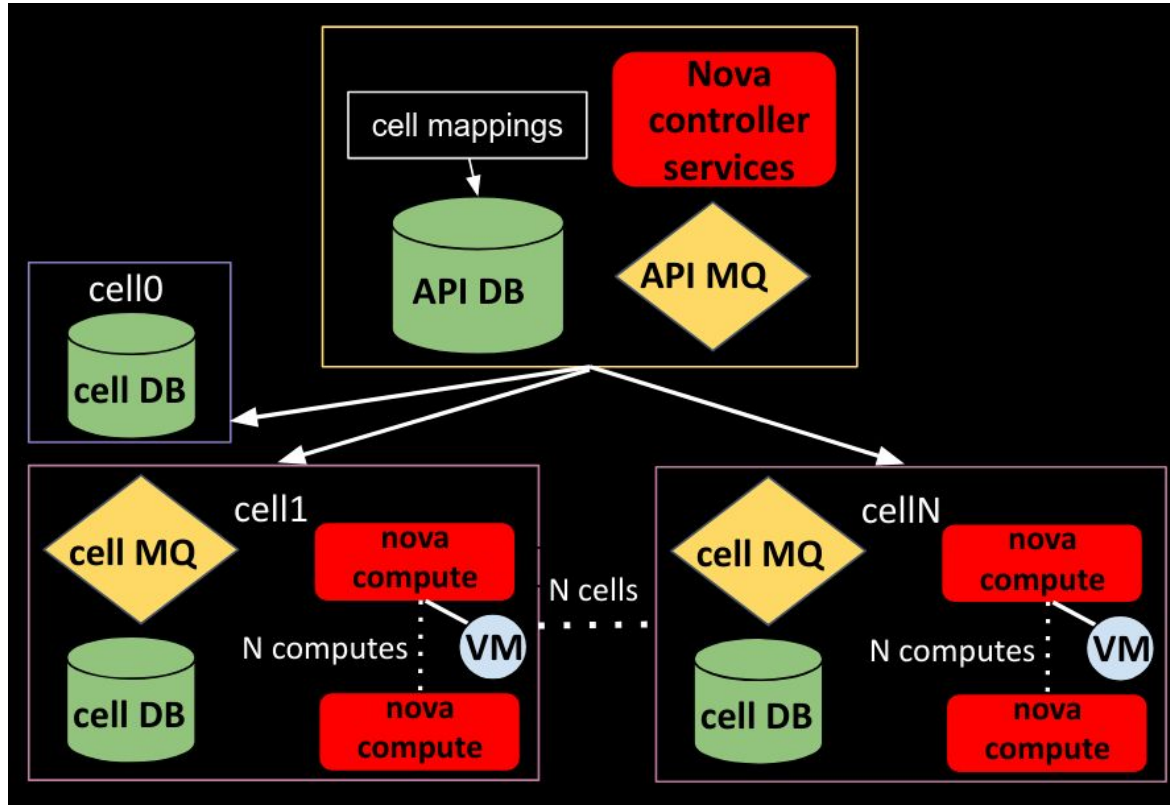
Matt Riedemann (mriedem on IRC) - Huawei
Surya Seetharaman (tssurya on IRC) - CERN



Overview

1. Introduction to Nova Multi-Cells
2. What's new in Cells?
 - a. Handling Down Cells
 - i. Making listing operations more resilient
 - ii. A new mechanism for calculating Quotas
 - iii. Operator and user highlights
 - iv. Known issues and limitations
 - b. Cross-cell Resize
 - i. Use cases
 - ii. Design specifics and implementation workflow
 - iii. Known issues and limitations

Nova Cells (multi-cells-v2)

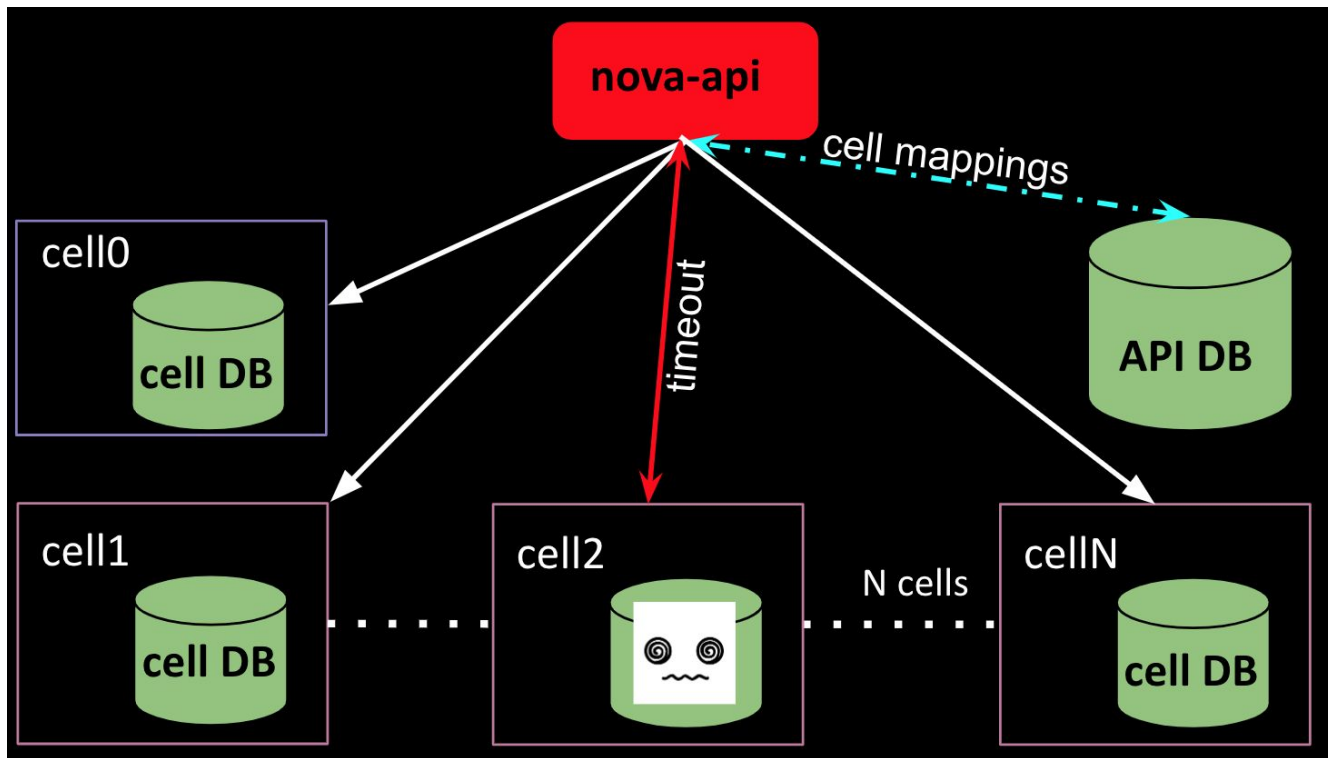


See [nova cells](#) for a more detailed view.

Handling Down Cells

- A step towards making cells more resilient.
- Available from the **Stein** release.

Problem Statement





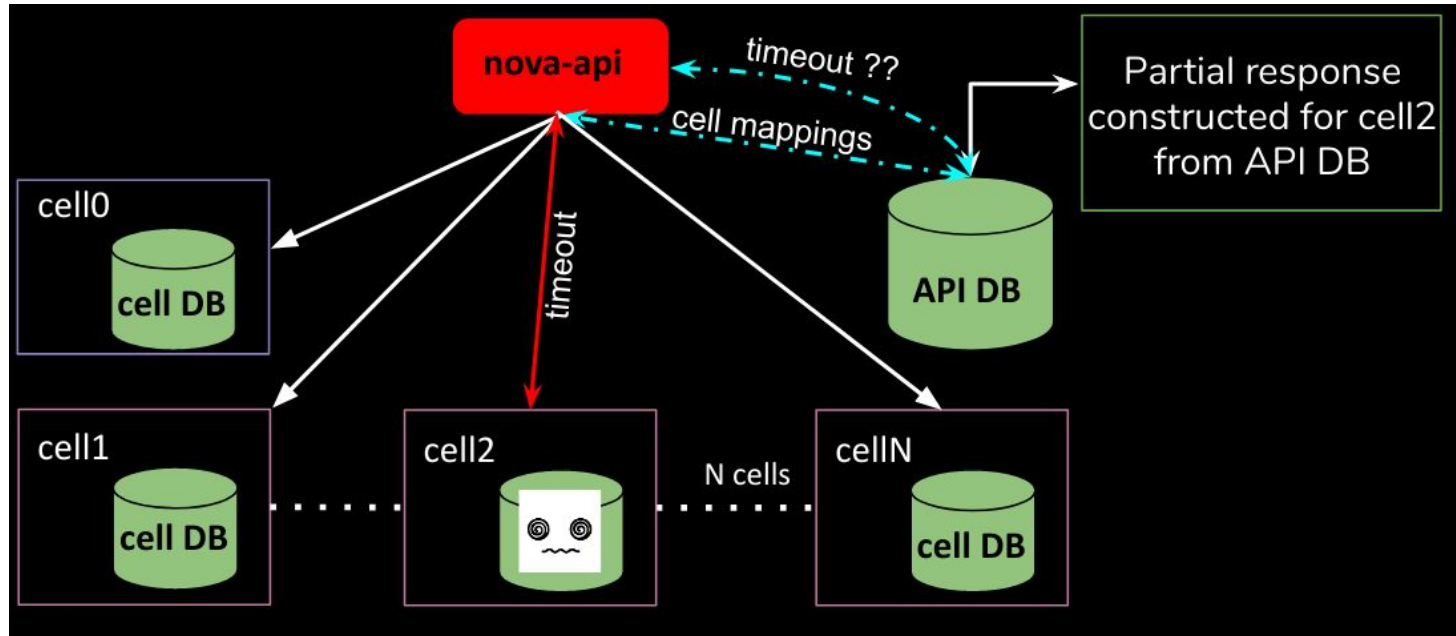
Problem Statement

- When a cell goes down basic operations like `GET /servers` and `GET /os-services` **stop working** for the whole infrastructure.
- However one cell going down should not affect the users and operators from listing resources from the API.

A single cell going down **should not** impact the whole infrastructure

Implemented Solution

Return **partial information** for the down cells from the **API database**





Scoped Use Cases

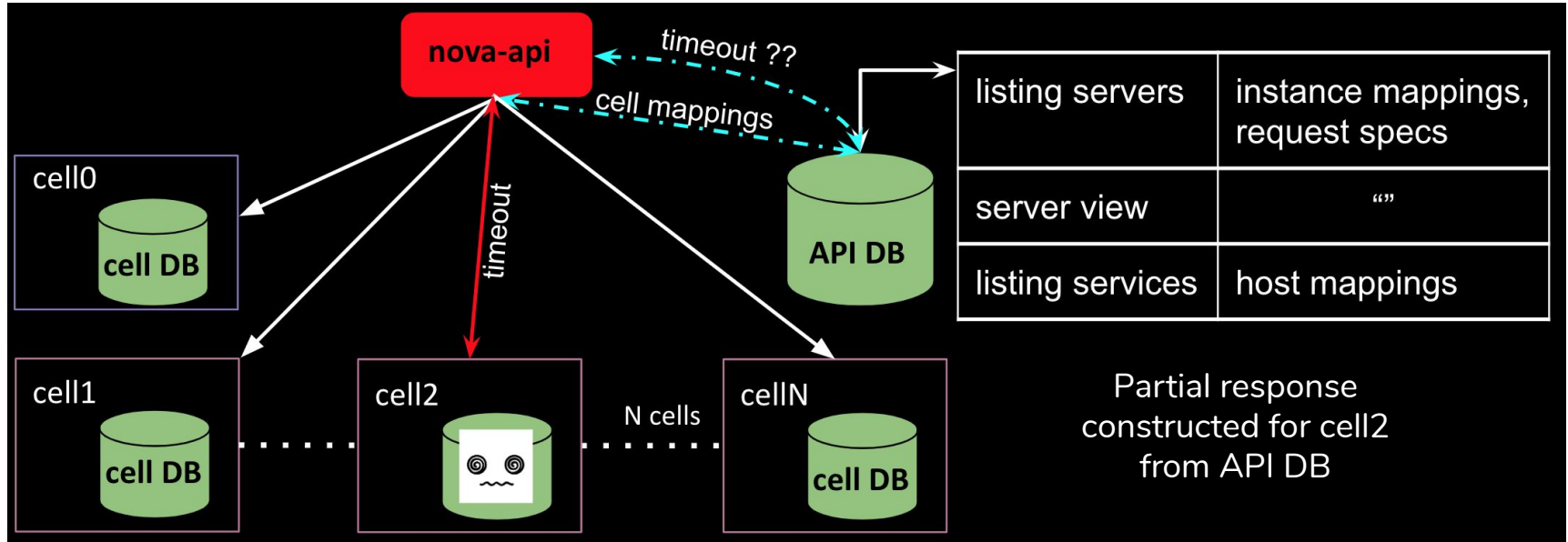
The specific use cases that have been addressed using the aforementioned solution are:

1. Listing Servers
2. Viewing a Server
3. Listing Compute Services
 - 3.1. Note that this is limited to the “nova-compute” services per cell.

See [handling down cells](#) for more information.

Implemented Solution

Return **partial information** for the down cells from the **API database**





Example Scenario

We have three cells which are all up:

Name	UUID
cell0	00000000-0000-0000-0000-000000000000
cell1	808bfe0b-fcb5-429f-a387-604496d3549d
cell2	f61a032f-014b-4193-a74e-fc21db3191d7

We force cell2 to go down:

```
Apr 15 08:37:12 surya001 devstack@n-api.service[31379]: ERROR nova.context [None req-c14193de-5ea3-40c3-ab13-b07f24d80cd6 admin admin] Error gathering result from cell f61a032f-014b-4193-a74e-fc21db3191d7: DB ConnectionError: (pymysql.err.OperationalError) (2003, "Can't connect to MySQL server on u'137.138.157.150' ([Errno 111] ECONNREFUSED)") (Background on this error at: http://sqlalche.me/e/e3q8)
```



Listing Servers

Response when cell0 and cell1 are up but cell2 is down:

```
Apr 15 08:37:12 surya001 devstack@n-api.service[31379]: WARNING nova.compute.api [None req-c14193de-5ea3-40c3-ab13-b07f24d80cd6 admin admin] Cell f61a032f-014b-4193-a74e-fc21db3191d7 is not responding and hence only partial results are available from this cell.
```

ID	Name	Status	Task State	Power State	Networks
28d36855-8d78-4598-bf16		UNKNOWN	N/A	N/A	
313f19fd-21c6-4c1c-87f1		UNKNOWN	N/A	N/A	
9dfe0a8e-c4b8-41d3-a1cd		UNKNOWN	N/A	N/A	
a36a95a3-fd47-4a5f-8f4e		UNKNOWN	N/A	N/A	
c5124e7c-4f43-4e7d-a964		UNKNOWN	N/A	N/A	
f434d938-33cf-47b8-b0e9		UNKNOWN	N/A	N/A	
bcb67da5-46fa-41b6-b9b1	admintest3	ACTIVE	-	Running	public=2001:db8::155
639135ab-c557-4381-855f	admintest4	ACTIVE	-	Running	public=2001:db8::1ec
0b96767a-a2c6-4bad-88e7	admintest5	ACTIVE	-	Running	public=2001:db8::18c
178bb359-1392-44a2-a155	admintest78	ACTIVE	-	Running	public=2001:db8::35b



Viewing a Server

From a down cell

```
ubuntu@surya001:~/devstack$ nova show 313f19fd-21c6-4c1c-87f1-8469925b6bc6
```

Property	Value
OS-EXT-AZ:availability_zone	UNKNOWN
OS-EXT-STS:power_state	0
created	2019-01-25T12:50:07Z
flavor:disk	1
flavor:ephemeral	0
flavor:extra_specs	{}
flavor:original_name	m1.nano
flavor:ram	64
flavor:swap	0
flavor:vcpus	1
id	313f19fd-21c6-4c1c-87f1-8469925b6bc6
image	cirros-0.4.0-x86_64-disk (f36d26a0-c5f4-498e-960f-c6cdc6cad126)
status	UNKNOWN
tenant_id	3b6beaf083204623b3f54fdbded16916
user_id	6104fcfea7c746f9b0380e6bc64fc906



Listing Services

Normal response when all cells are up:

```
ubuntu@surya001:~/devstack$ nova service-list
```

Id	Binary	Host	Zone	Status	State	Updated_at	Disabled Reason	Forced down
4b45fd51-89fe-4450-bdfe-2e2f3be8dcde	nova-scheduler	surya001	internal	enabled	up	2019-04-15T08:37:38.000000	-	False
10e2d352-3ee2-4be9-a488-26f6c1377d49	nova-consoleauth	surya001	internal	enabled	up	2019-04-15T08:37:38.000000	-	False
f8b91f2a-c333-4876-ae59-bbe046178494	nova-conductor	surya001	internal	enabled	up	2019-04-15T08:37:38.000000	-	False
4977129d-fe28-4656-94f1-2a8bac0613e1	nova-conductor	surya002	internal	enabled	up	2019-04-15T08:37:39.000000	-	False
c7019841-0b68-4f6a-9812-31dab57ec554	nova-compute	surya002	nova	enabled	down	2019-04-15T08:31:34.000000	-	False
407be032-5046-47c4-b64c-194543a8ce9a	nova-conductor	surya001	internal	enabled	up	2019-04-15T08:37:40.000000	-	False
e394202a-482e-41c8-8de3-842b36a916b3	nova-compute	surya001	nova	enabled	down	2019-04-11T14:17:07.000000	-	False

Response when cell0 and cell1 are up but cell2 is down:

```
ubuntu@surya001:~/devstack$ nova service-list
```

Id	Binary	Host	Zone	Status	State	Updated_at	Disabled Reason	Forced down
4b45fd51-89fe-4450-bdfe-2e2f3be8dcde	nova-scheduler	surya001	internal	enabled	up	2019-04-15T08:37:08.000000	-	False
10e2d352-3ee2-4be9-a488-26f6c1377d49	nova-consoleauth	surya001	internal	enabled	up	2019-04-15T08:37:08.000000	-	False
f8b91f2a-c333-4876-ae59-bbe046178494	nova-conductor	surya001	internal	enabled	up	2019-04-15T08:37:08.000000	-	False
407be032-5046-47c4-b64c-194543a8ce9a	nova-conductor	surya001	internal	enabled	up	2019-04-15T08:37:10.000000	-	False
e394202a-482e-41c8-8de3-842b36a916b3	nova-compute	surya001	nova	enabled	down	2019-04-11T14:17:07.000000	-	False
	nova-compute	surya002		UNKNOWN				



User highlights

- From **microversion 2.69** partial results will be available from the down cells.
- **Prior to 2.69**, depending on [list_records_by_skipping_down_cells](#) user will either get :
 - A response where **results are skipped from the down cells** when the config option is set to True (default).
 - A **500** error response when the config option is set to False.

All the **edge cases** that are **not supported for minimal constructs** would give responses based on the operator's **configuration** of the deployment, **either skipping those results or returning an error.**



Edge Cases

- **Filtering**: partial constructs are **not supported** with filters since it is not possible to validate the matches from the down cells.
 - “all-tenants/all-projects” and “minimal” are supported.
- **Marker**: if the marker specified is an **instance from a down cell** the request will fail with a **500** error code.
- **Sorting**: partial constructs are **not supported** like for the filters.
- **Paging**: partial constructs are **not supported** like for sorting and filtering.



Operator highlights

- Configuration considerations for a cell timeout
 - [database.max_retries](#): by default 10 times before nova declares the cell is unreachable.
 - [database.retry_interval](#): by default 10 seconds
 - [CELL_TIMEOUT](#): hardcoded to 60 seconds after which nova-api gives up and returns partial constructs.
- **Disabling** down cells:
 - removed from being a scheduling candidate.

See [cellsv2_management](#) for more information.



Known Issues

- **nova-api service hangs** on startup.
 - if at least one cell is down and `upgrade_levels.compute = auto`
 - It needs to connect to all the cells to gather the compute service's RPC API version to determine the version cap.
 - See [bug 1815697](#) for more details.
 - workaround is to pin [upgrade_levels.compute](#) to a specific release.
- Performance **degradation**.
 - with regards to operations that need to hit all cells.
- Needless to say that down cell targeted operations like **server creation or deletion will not work**.

Quota Calculation

- Introducing a new quota calculation system that is independent of cells!





Problem Statement

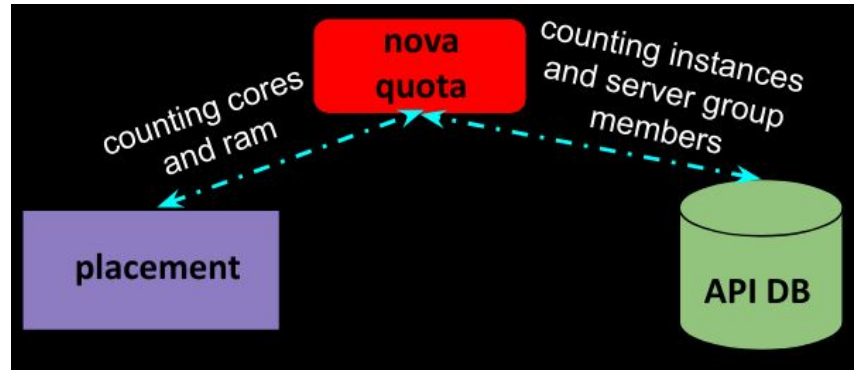
- Cores, RAM and instances are counted by reading all the cell databases and aggregating the results.
 - We use the [scatter-gather utility](#) to loop through cells in parallel.
- Quota calculation mechanism **skips counting resources** from the **unreachable cells**.
 - Hence if the user had instances in the down cell these would **not have been accounted** for when they request a new server creation.
 - However **when the cell comes up** this will have implications since now **the user would be using more resources than allowed**.

A cell going down **should not** impact the quota calculation

Implemented Solution

Counting Resources from **Placement** and **API database**

- Instead of looping over all the cell databases we simply
 - count instances from the API database
 - count RAM and cores from placement



Implementation credit: Melanie Witt (melwitt on IRC) - RedHat



Operator Highlights

- You have to opt-into the new way of counting by setting `[quota]count_usage_from_placement` to True.
 - By **default** nova will still use the **legacy way** of counting quotas from the cell databases.
- Run online data migrations before using the new system
 - else the mechanism **will fallback to the legacy** way of counting resources.

See [count_quota_usage_from_placement](#) for more details



Operator Highlights (continued)

- Behavior changes from legacy **counting for cores and ram**:
 - ERROR instances in cell0 will not be counted
 - During resize quota counting is doubled
 - counts allocations against source and destination
- Limitation:
 - Deployments using **multiple nova's** and a single placement must not use placement to count quotas.

Cross-cell Resize





Use case

- Cloud uses cells to shard by hardware generation and wants to migrate servers from old cells to new cells
- Users can naturally aid in the cell migration by resizing their servers and retain volumes/ports/UUID



Design overview

- Tries to follow traditional resize flow but with entirely new code
 - Server state transitions will be the same
- Enables cold migrating to a target host in another cell
- Full orchestration from (super)conductor using RPC calls
 - RPC timeout controlled with [long_rpc_timeout](#) option
- Target host is validated for volume and port connections

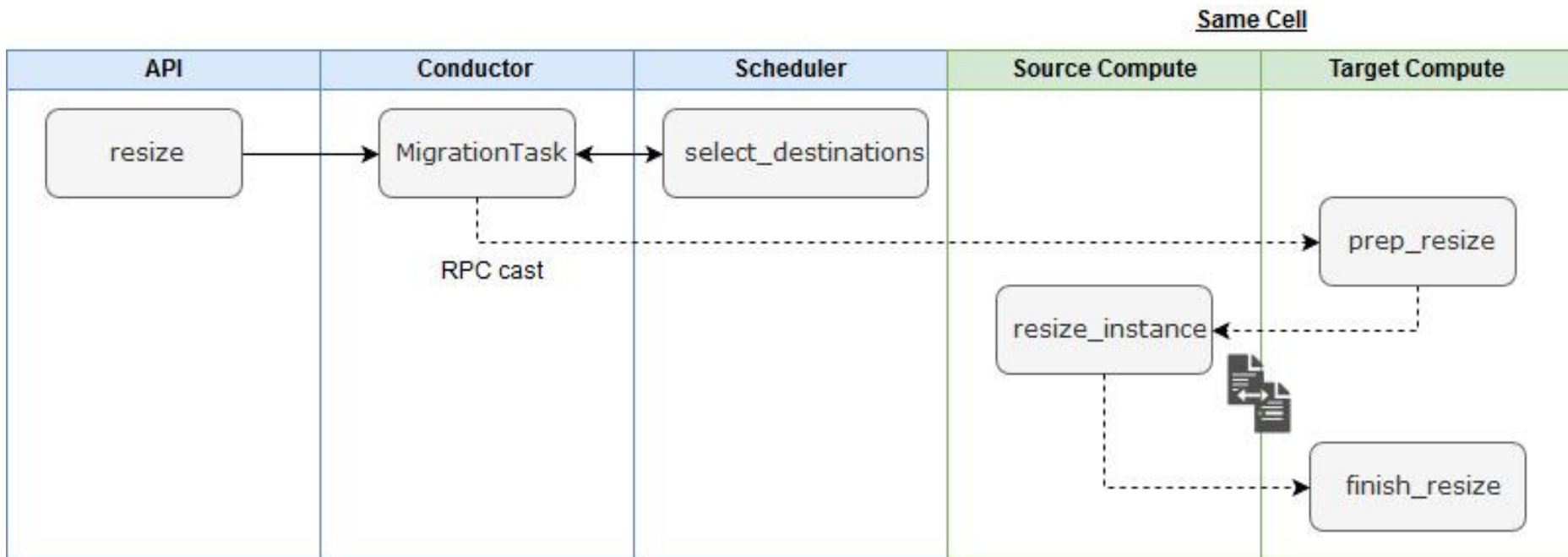


Design overview (continued)

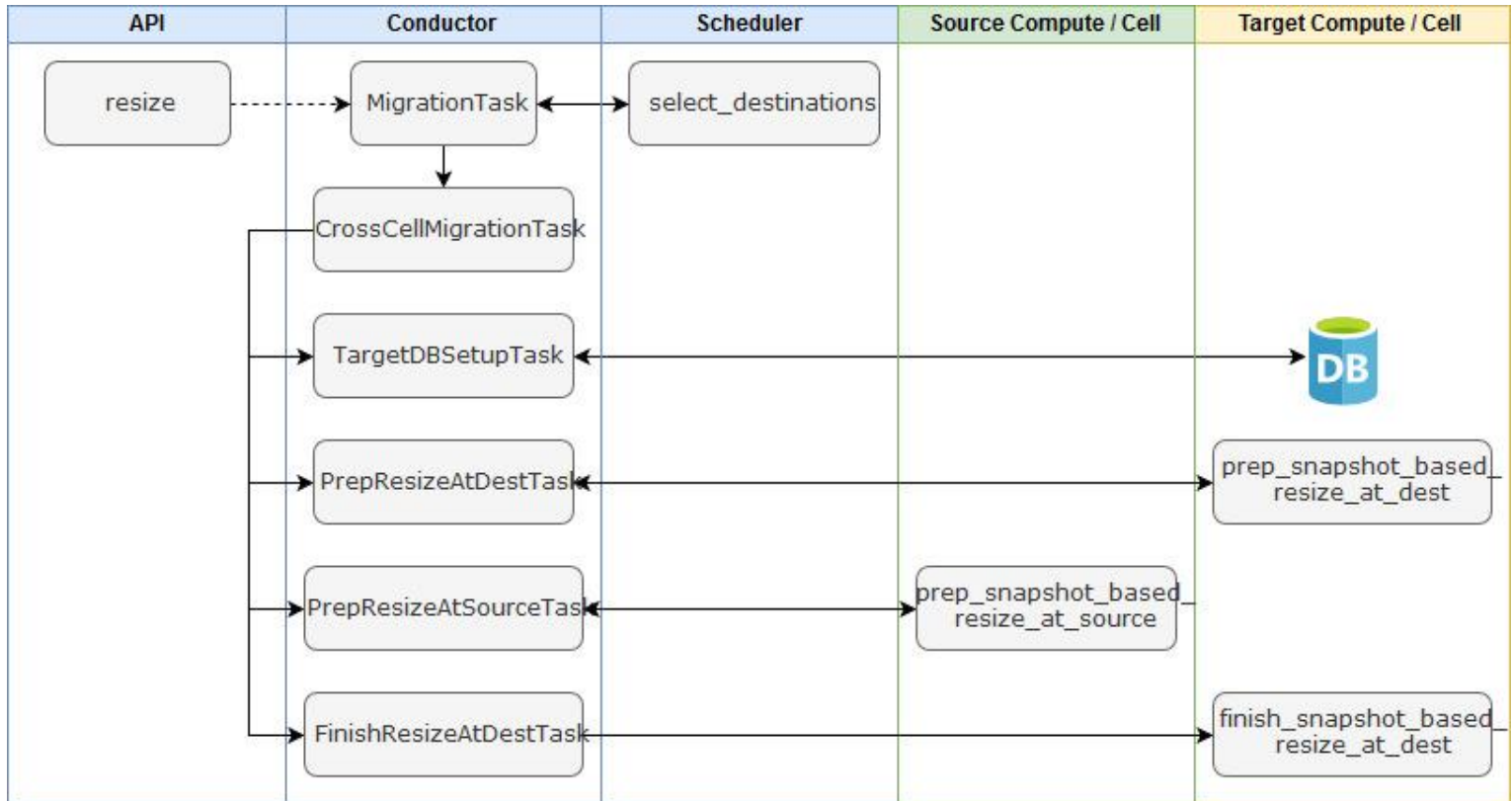
- Instance.hidden field added
- Temporary glance snapshot created for non-volume-backed servers (like shelve)
- New policy rule: `compute:servers:resize:cross_cell`
 - Disabled by default for **all** users
- CrossCellWeigher added



Traditional resize flow



Cross-cell resize flow





Comparison summary

	Traditional	Cross cell
Blocking API	Until prep_resize on dest	Until cast to conductor
Orchestration	Computes RPC to each other	Conductor orchestrates between cells and computes at the top
Root disk file transfer	Direct copy between hosts	Temp snapshot in glance
Database	Single, no duplication	Duplicate records created in the target cell DB



Limitations and known issues

- Personality files are not retained
- Config drive will be rebuilt in the target cell
- `_poll_unconfirmed_resizes` periodic task may not work
- Some instance action **events** will be different from traditional resize
- Notification source may change ([global vs per-cell notification queue](#))



Help wanted

- Reviews
 - <https://review.opendev.org/#/q/status:open+topic:bp/cross-cell-resize>
- Testing
 - Manual
 - CI: [nova-multi-cell job](#)

Thanks for listening!
Questions??

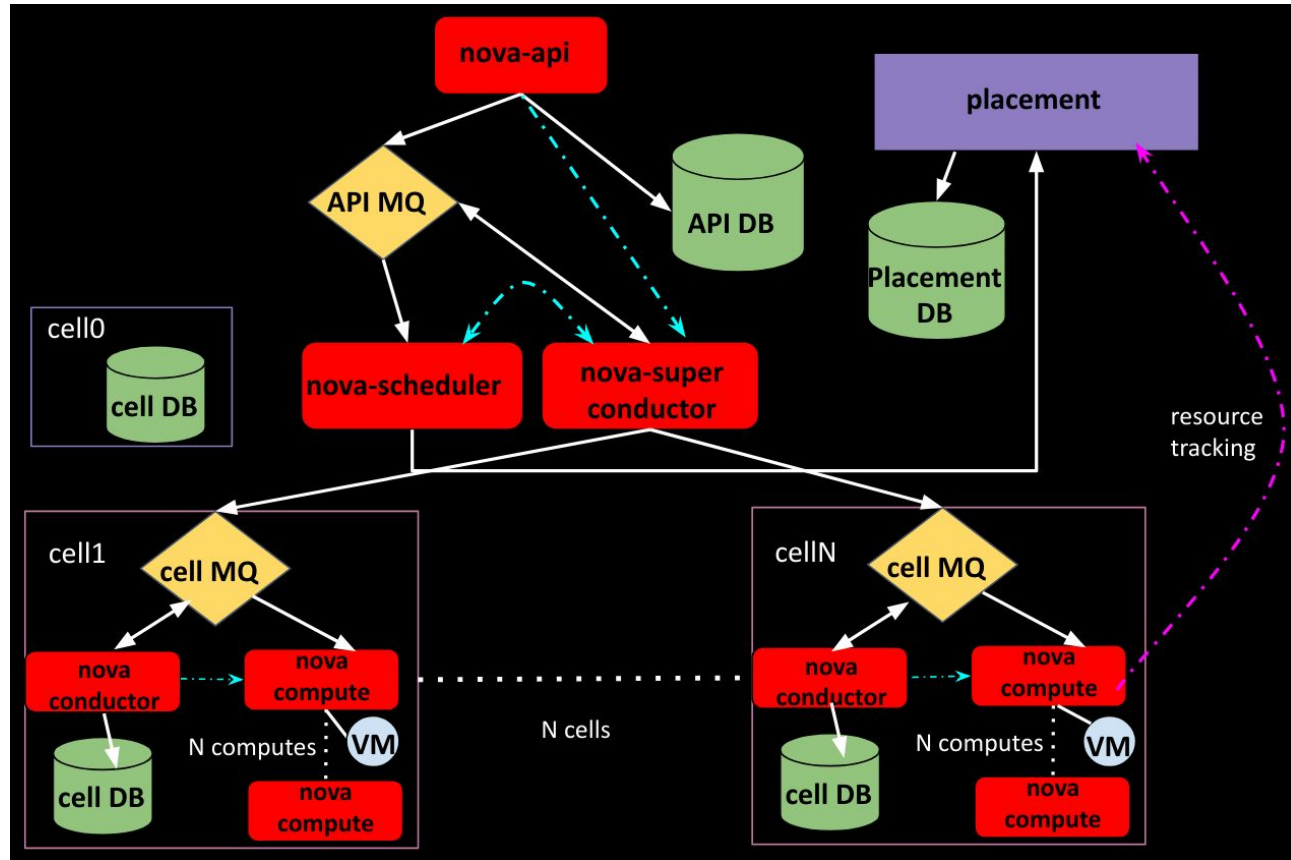
Backup



Discussed Potential Solutions

- Using **searchlight to backfill** when there are down cells. Check out [listing instances using Searchlight](#) for more details.
- Adding **backup DBs** for each cell database which would act as read-only copies of the original DB in times of crisis.
 - however this would need massive syncing and may fetch stale results.

Reality... :)





Implemented Solution

Return **partial information** for the down cells from the **API database**

- Gather all the responses for the records from the up cells like normal and when we find down cells,
 - Go to the **API database** and fill in the available information for those records from the down cells.
 - As a result the response will have **missing information** for the records from the down cells.
 - The status of such records will be “**UNKNOWN**” for the users to realize the transient down time.