

MidPoint Community Meetup 2025

Cloud Resources with Rate Limitations



Join at slido.com #3042 573



Pavol Mederly, May 2025 Senior Software Developer

Agenda

- Motivation
- How midPoint helps
- Demo
- Configuration details
- Tips and tricks
- Limitations
- Future work

Evolveum

Conclusion + Q&A

Join at **slido.com #3042 573**











Community Meetup

Default operation



Reading when needed

- Bulk reading of data on the source system being synchronized (import, reconciliation)
- Ad-hoc reading of data from other systems

Updating when needed

 Executing modifications as soon as they are detected

Provides good data consistency, but not suitable for slow and/or limited resources!



Improving the efficiency

Evolveum

• **Shadow caching:** after reading from the resource, data are stored in midPoint repository.



• **Operation grouping:** instead of writing to the resource, changes are stored in midPoint repository







- When midPoint needs a shadow or a shadow item and it is in the repository, we won't retrieve it from the
 resource
- This eliminates resource reads for:
 - **outbound** mappings
 - strong, weak, or with a range
 - **inbound** mappings
 - during **recomputation** task (with default settings)
 - during import and reconciliation tasks (if "reconcileAll" option is set)



How to enable

<internals>

<shadowCaching>

<defaultPolicy>

<cachingStrategy>passive</cachingStrategy>

<defaultCacheUse>useCachedOrFresh</defaultCacheUse>

</defaultPolicy>

</shadowCaching>



Demo

- Four systems: 2 source, 2 target ones
- 10 accounts on each
- Basic mappings, all strong



Configuration

- What is cached
 - attributes: all / <u>defined</u> / mapped / none (with overriding)
 - associations: <u>all</u> / none (with overriding)
 - activation: <u>all</u> / none
 - credentials: <u>all</u> / none
 - auxiliary object class information: all / none
- For how long (time to live)
 - default is 1 day
- Overridable

<caching> <cachingStrategy>passive</cachingStrategy> <scope> <attributes>all</attributes> <associations>all</associations> <activation>all</activation> </scope> <timeToLive>P1D</timeToLive> </caching>

• Systemwide \rightarrow resource \rightarrow object class \rightarrow object type \rightarrow attribute/association



Configuration

<caching>

<cachingStrategy>passive</cachingStrategy>
<scope>

...

</scope>

<defaultCacheUse>useCachedOrFresh</defaultCacheUse>

- </caching>
- Cached data use

Evolveum

• Random access (recomputation or other operations)

Л

- use fresh data (as in 4.8)
- use cached or fresh
- use cached or ignore
- use cached or fail
- Synchronization task (import, reconciliation)
 - from resource or from cache (fetch or no fetch)

<activity> </work> </objects> </objects> </objects> </objects> </objects> </cochectedShadowsUse>useFresh</cachedShadowsUse> </recomputation> </recomputation>

</work>

</activity>





Typical use

• Naive

- Just turn it on, [set TTL,] and see what happens
- Organized, e.g.
 - Set TTL to 24 hours
 - Set up a nightly task that will fetch all accounts
 - (Either with full synchronization, or as simple refresh.)
 - Maybe with using live sync to update the cached data in regular intervals



Things to watch for

- Volatile attributes
- Performance



Limitations

- Not all operations go through the cache
 - Iteration: midpoint.isUniqueAccountValue
 - Low-level MODIFY related operations: avoiding duplicate values, volatile attributes handling, READ+REPLACE mode
- Time to live
 - respected for ad-hoc (random) reads
 - but ignored for search operations (import and reconciliation with noFetch mode)
- Minor limitations for generic repo



Future work

- Overcoming known limitations
- Monitoring and visibility
- Enforcing operation or rate limits via task/activity policies
 - Stopping the task
 - Slowing down the task



Conclusion

- Shadow caching and operation grouping can significantly help with slow or costly resources
- Sometimes it may require little bit of planning and finetuning
- Please, try it and give us a feedback







Thank you for your attention

Feel free to ask your questions now!

Join at **slido.com #3042 573**



Operations grouping + provisioning propagation

- Resource is set to postpone (group) operations
- A propagation task executes these operations
 - but only if specified time interval (e.g. 1 hour) elapsed

<resource oid="f34328fa-f229-11e7-8cc9-e3f77b478e97">

```
...
<consistency>
<coperationGroupingInterval>PT1H</operationGroupingInterval>
</consistency>
...
</resource>
<task>
...
```

```
<activity>

<activity>
<propagation>

<resourceRef oid="f34328fa-f229-11e7-8cc9-e3f77b478e97"/></propagation>
</work></activity></task>
```



