Concept Caching In Clustered Env

Your Rating: ☆☆☆☆☆ Results: ★★★☆☆ 78 rates

Summary of findings and ideas for cache configuration for JR clustered installations.

Considering options for advaning cache implementation to behave correctly in clustered env.

Introduction

Probably enterprise only feature.

This concept is based on issue MGNLADVCACHE-10 - Getting issue details... STATUS found while testing clustering and subsequent discussion.

Use case

- page commenting
- any other non activation based content updates

EHCache

EHcache supports clustering.
Considerations for using EHcache clustering support:

- public instances would need to know each other (IP), making public instance configuration more complicated ... the author will have only one subscriber, but public that will be such subscriber would need to know about all other public instances ... what would happen if such public goes down ... would need to figure out which of the cache slaves will become master and change subscriber on the author to that master
- the cache configuration would need to be different based on cluster configuration - full vs. partial clustering vs. partial clustering (specially if author itself participates in the cluster)
- solves the issue only for EHcache, making initial barrier for implementing other underlying cache for Magnolia even more difficult.

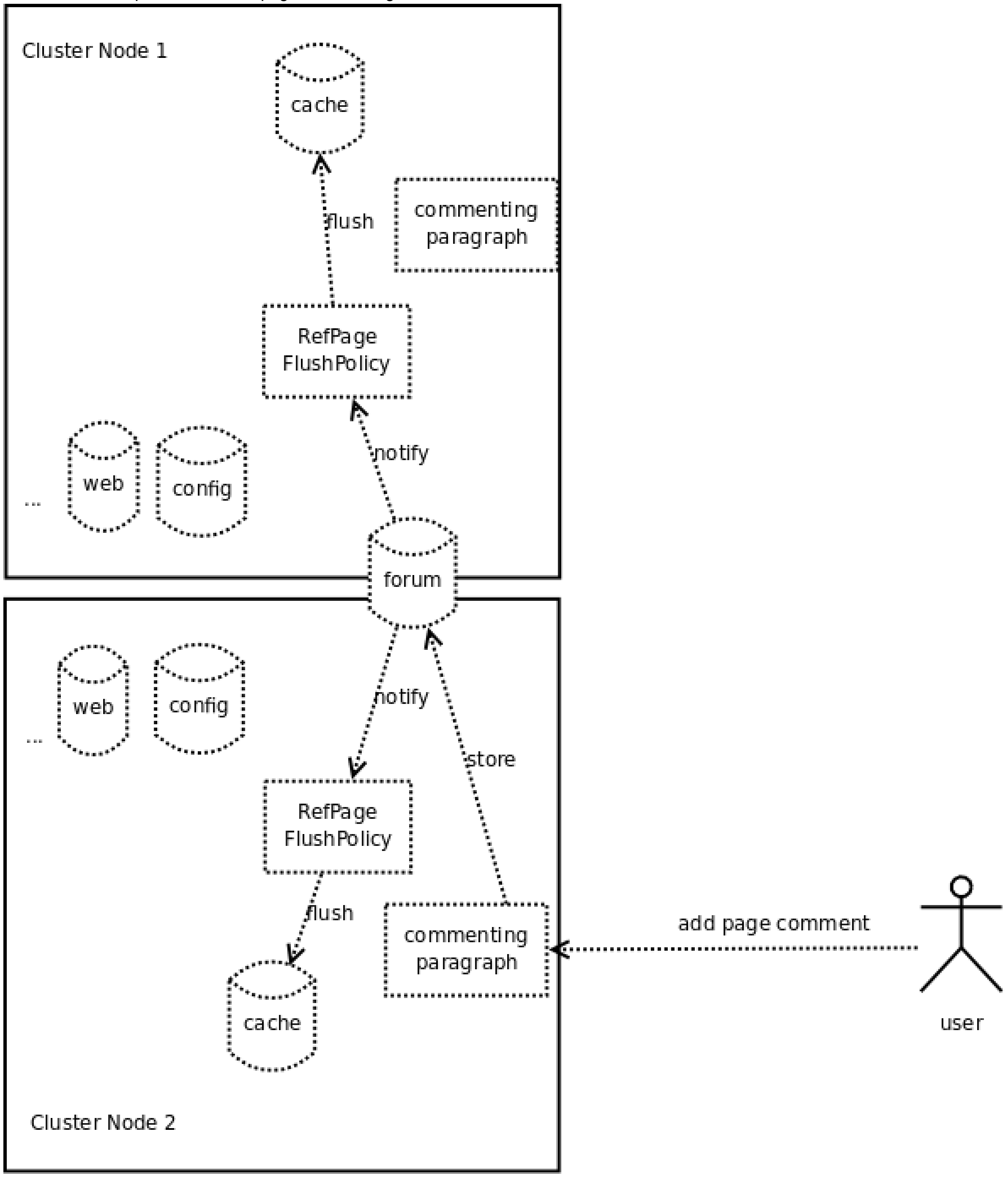
Magnolia cache

Considerations for advancing current cache implementation:

- instances participating in the cluster are already bound together by JR journalling and only events related to clustered workspaces are propagated across cluster, no extra config necessary
- there is mechanism in place for instructing the cache (Flush policy) which works well in clustered environment
- the Flush Policy is already implemented for flushing all the content (and used for activation)
- new flush policy for flushing single only items would have to be implemented
- The advanced caching techniques manipulating cache items directly would need to be rewritten to use Observation for all cache manipulation to ensure events are propagated across cluster
- New method (since 4.1) for flushing single page from cache would need to be changed to use single page flush policy instead of retrieving cache instance and removing content directly

Implementation details

Cache updates should be realized only via observation to ensure distribution over the cluster.
Schema of the implementation for page commenting based on observation



First code changes have been committed.

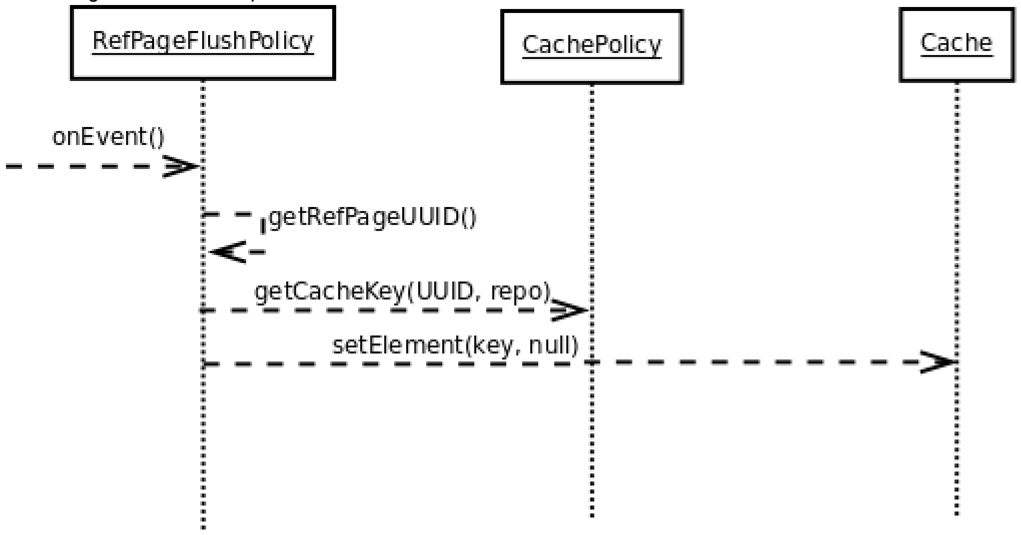
In cache module MAGNOLIA-2616 - Getting issue details... STATUS :

- New methods in CachePolicy API to allow exposing conversion of uuid to cache key, should the cache policy support this.
- Implementation of the above for `Default` cache policy.

In commenting module MGNLCMNT-2 - Getting issue details... STATUS :

- Implementation of the flush policy that is able to detect page that needs to be flushed based on new comments being added.
- Registration of the above policy on the startup of the module

Class diagram of current implementation:



Pending

- peer review
- what might still change
    - is some method naming, and
    - probably final location for the code related to flushing the page in the policy. Since this is rather generic, it might be moved to either cache or advanced cache module to expose functionality to the others who might be interested in using it.
    - initialization of the flush policy ... still undecided whether it is really business of the commenting module to register this or whether cache module should take care.
    - configuration - is it enough that there is only this one policy that can be used with commenting, or is there a real chance that we or someone else might ever want to replace it with something else in some case.