AEM DISPATCHER TIPS & TRICKS

Practical Scenarios, Issues, Tips and Tricks in configuring AEM dispatcher

Farms

- Split dispatcher configuration into multiple farms
- Have as many farms as the different caching behavior needed
  - E.g., DAM & Pages almost always need different caching behavior. DAM assets are independent while Pages has dependencies amongst each other
- All configuration parameters are at farm level and can be configured differently for each farm
- Avoid having single long configuration file
  - Decompose farm configuration elements into independent files and include them into a farm configuration file
  - Include all farm configuration file into the dispatcher.any file

Apart from bringing in clarity, this would help avoid duplicating configuration entries. For example, renderers & allowed clients for all farms might be same, while cache invalidation behavior would be different. Have the configuration files split as appropriate for your scenario

Cache Folder

- Its possible to configure different cache folder for each dispatcher farm
- Can have cache folder of one farm nested within the cache folder of the other farm
- We can use this to have one farm touch the stat file at a different level on other farm
- For example, we can have one farm have the cache root as htdocs and another farm have cache root as htdocs/content. This creates the folder and corresponding .stat file entries as
  - htdocs (.stat)
    - content (.stat)
      - content (.stat)

Cache Invalidation

- When having multiple farms, which farm configuration gets used for cache invalidation requests?
- Cache invalidation requests does not use the configuration of the farm that matches the resource that gets invalidated
- Cache invalidation requests for all resources is to the url /dispatcher/invalidate.cache and matches the farm that handles this url
- The resource to invalidate is set in the http header CQ-Handle
- Use url rewrite in Apache to append CQ-Handle to /dispatcher/invalidate.cache to match the farm of the resource invalidated

Vanity URL's

- AEM simplifies managing vanity url's for a page, allowing setting multiple vanity url's for a page
- Vanity url can be any valid url string, irrespective of the level or depth of the source page
- When accessing a page through its vanity url, dispatcher caches that page at the path which is the same as that of the vanity url
- But AEM does not generate the cache flush request for the vanity path, cache flush request is generated only for the full path of the page
- Only easy option to handle this is to set the statfileslevel to 0 for the farm which handles vanity url's

Dispatcher Cache Flushing

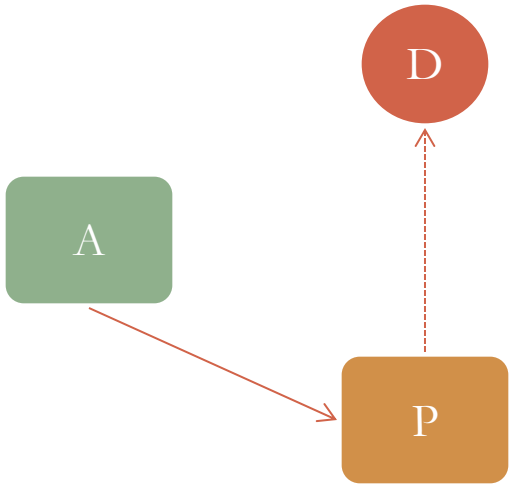
- Cache flushing from Author is easy, always possible, but might lead to race condition
  - Do not use this if your site has significant customer load and content publishing might happen when there is customer activity on the site
- Cache flushing from Publish is tricky unless you have just one publisher and one dispatcher

Remember you cannot do the configuration on the publisher directly, all configuration should be done on the author and activated to be applied on the publish instance (at least when you are doing it straight)

- Gets even more trickier if dispatcher is used for load balancing

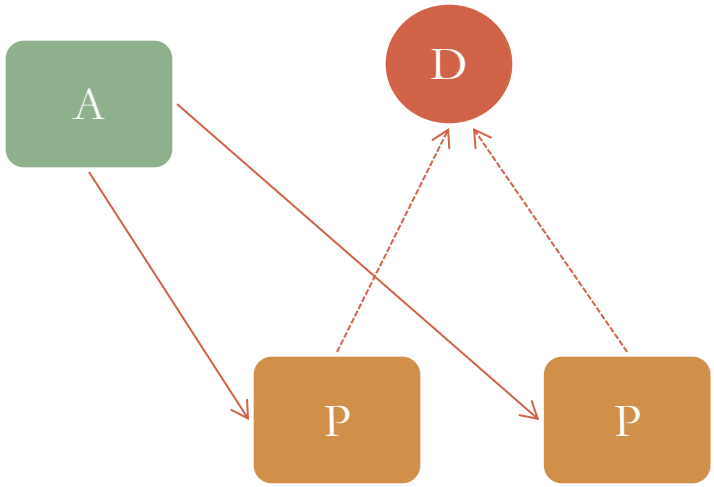
Some Dispatcher Cache Flushing Scenarios

- One publisher and one dispatcher, the easiest but hardly used configuration
- Configure the dispatcher flush agent for publish on the author and activate it
- Single publisher flushes the activated resources on the dispatcher



Some Dispatcher Cache Flushing Scenarios

- Multiple publishers and one dispatcher
- It might be acceptable if each publisher flushes the cache if the no. of publishers is 2. This would not be acceptable when no. of publishers increases
- Designate one or two publishers as flushing publisher and configure dispatcher cache flush agent on those publishers



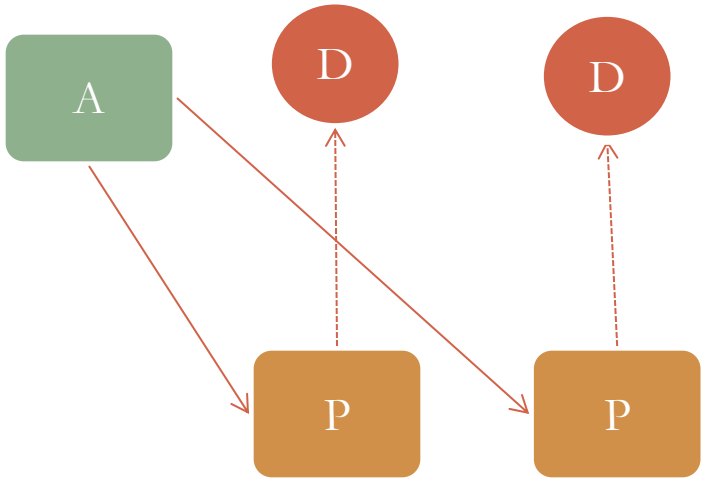
Race condition might occur when the flushing publisher clears the cache and the dispatcher updates the cache from the other publisher where replication has not been completed yet. But the probability of such race condition is less compared to cache flushing from Author

Dispatcher Cache Flush Agent configuration directly on publish instances

- Use curl scripts on publish to create the dispatcher cache flush agent
- Configure the dispatcher cache flushing agents on the Author, create packages of the cache flushing agents as needed by selecting appropriate nodes and install it on the publishers as needed
- Login to the publisher to do the configuration (AEM doesn't directly support this, but allows access to the login page and after login allows access to tools page for doing this configuration)

Some Dispatcher Cache Flushing Scenarios

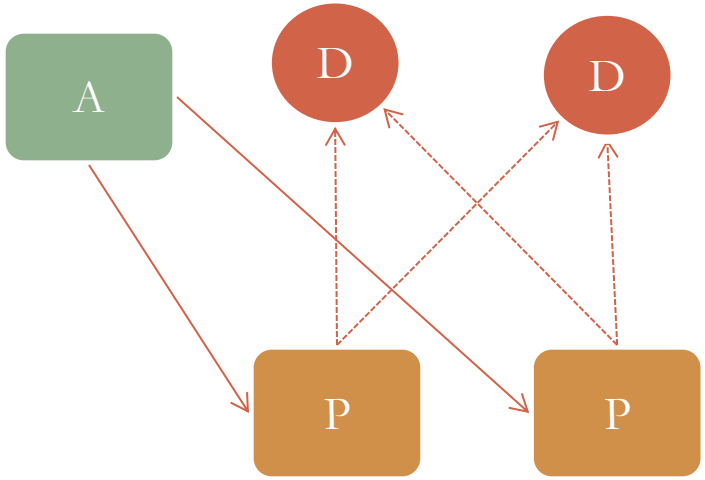
- Multiple publishers, multiple dispatchers, but each publisher is connected to single dispatcher and vice versa
- Looks straight forward, but there is no way to handle dispatcher cache flushing through Author.
- Use one of the option mentioned in the slide 'Dispatcher Cache Flush Agent configuration directly on publish instances' to configure



No race condition in this configuration as each publisher flushes the cache on its corresponding dispatcher. For this reason, its preferable to have one to one match between publisher and dispatcher

Some Dispatcher Cache Flushing Scenarios

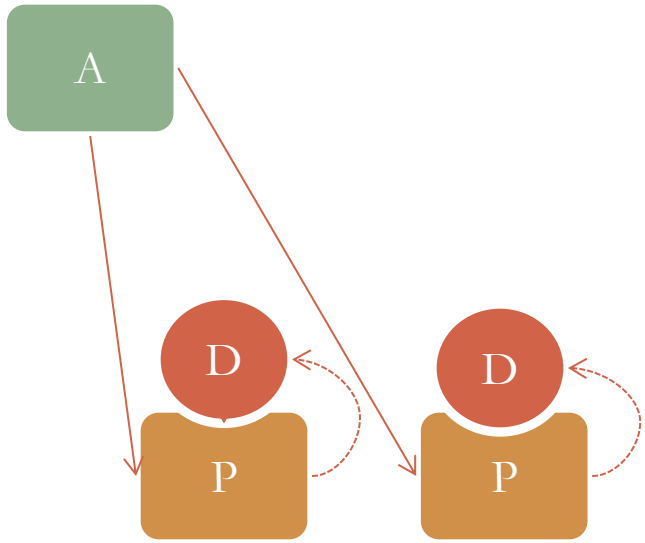
- Multiple publishers with multiple dispatchers
- It might be acceptable if each publisher flushes the cache on each dispatcher if the no. of publishers and dispatcher is 2. This would not be acceptable when no. of publishers or dispatchers increases
- Designate one or two publishers as flushing publisher for each dispatcher and configure the dispatcher cache flush agent for that dispatcher on those publishers



Again race condition might occur in this configuration. Reason for configuring more then one publisher to clear cache on a dispatcher is to reduce the probability of occurrence of the race condition

Some Dispatcher Cache Flushing Scenarios

- Multiple publishers with multiple dispatchers with each publisher having its corresponding dispatcher
- Dispatcher and publisher are hosted on the same server
- This configuration becomes easy for configuring dispatcher cache flush agent.
- Agent can be configured on the Author, mentioning localhost or 127.0.0.1 as dispatcher address and replicated to all publishers



This configuration would not be possible if web server needs to be deployed in the DMZ.

Disabling Dispatcher Cache for select clients

- There might be scenarios when we might need to bypass dispatcher cache and access page in AEM from some client for test purpose
- One approach to do this is to set a cookie and have server check this cookie to decide if it needs to fetch the page from cache or serve it from the source directly
- Dispatcher does not use cookie as a parameter to decide caching behavior
- To handle such scenario, use url rewrite on Apache to fetch the cookie, and set it as url parameter
- Configure dispatcher farm to not ignore this url parameter for caching

THANK YOU

Feedback and suggestions welcome. Please write to ashokkumar.ta@gmail.com