

Item 238

git_comments:

git_commits:

1. **summary:** IGNITE-5479 - Move constant
message: IGNITE-5479 - Move constant

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** PersistentStoreConfiguration improvements
description: Improve `PersistentStoreConfiguration` to be consistent with other configuration classes:
* `LockWaitTime` - should be `long` and in milliseconds * `getCheckpointingPageBufferSize` should return `long` instead of `Long` * `WalFlushFrequency` - should be `long` and in milliseconds * `WalFsyncDelay` - should be `long` and in milliseconds. Add javadoc.

jira_issues_comments:

1. Agree on all but: - `getCheckpointPageBufferSize` - this unique long is used to determine the default setting override. - `fsync` delay should be fine-grained enough to wait less than a millisecond because `fsync` latency itself may be 0.5 ms. Will rename this method to `getWalFsyncDelayNanos` to emphasize the different time unit
2. **body:** Agree on `fsync` delay. Do not agree on page buffer size. Internally we may maintain a nullable value, but user-facing method must be `long` (otherwise it is inconsistent and confusing; setter and getter should be of same type). Can we do this?

```
public long getCheckpointingPageBufferSize() {
    return checkpointingPageBufSize == null ? DFLT_CHECKPOINTING_PAGE_BUFFER_SIZE :
    checkpointingPageBufSize; }
```

label: code-design
3. **body:** This does not work: we need `Long` to distinguish between the following two cases:

```
new PersistentStoreConfiguration();
```

 // Ignite will automatically adjust checkpoint buffer size and

```
new PersistentStoreConfiguration().setCheckpointPageBufferSize(someValue);
```

 // Ignite will NOT adjust the checkpoint buffer size even if `someValue == DEFAULT_VALUE` This can be solved by adding a flag `isDefaultPageBufferSize` on `PersistentStoreConfiguration`, but the approach with `Long` looks more clear to me. What do you think?
label: code-design
4. **body:** What is the point of `DFLT_CHECKPOINTING_PAGE_BUFFER_SIZE` then? The actual value is never used, we just use a reference comparison? Let's change both setter and getter to `Long`, default to null, and state in the Javadoc that `null` means "automatic buffer size".
label: code-design
5. [~ptupitsyn] Can you please review the .NET part? The branch name is ignite-5479
6. **body:** I'm still not sure why `DFLT_CHECKPOINTING_PAGE_BUFFER_SIZE` exists. Also, on the second thought, null values are inconvenient for the user, so maybe we should switch to `long` and use `0` as "automatic".
label: code-design
7. Merged to ignite-2.1