Redmine - Patch #39847

Cache the result of `Journal#attachments`

2023-12-14 12:10 - Go MAEDA

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
<th>Start date:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
<td>Due date:</td>
<td>0%</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Go MAEDA</td>
<td>% Done:</td>
<td>0%</td>
</tr>
<tr>
<td>Category:</td>
<td>Performance</td>
<td>Estimated time:</td>
<td>0.00 hour</td>
</tr>
<tr>
<td>Target version:</td>
<td>6.0.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Description**

When rendering an issue history, Journal#attachments is called twice while rendering a single journal.

We can improve efficiency by caching the result of the method. With caching, unnecessary iterations over details and calls to Attachment.where can be reduced.

**Before:**

```
Attachment Load (0.4ms) SELECT "attachments".* FROM "attachments" WHERE "attachments"."id" IN ( $1, $2) [["id", 39], ["id", 38]]
↳ app/models/journal.rb:146:in `sort_by'

CACHE Attachment Load (0.0ms) SELECT "attachments".* FROM "attachments" WHERE "attachments"."id" IN ( $1, $2) [["id", 39], ["id", 38]]
↳ app/models/journal.rb:146:in `sort_by'
```

**After:**

```
Attachment Load (0.4ms) SELECT "attachments".* FROM "attachments" WHERE "attachments"."id" IN ( $1, $2) [["id", 39], ["id", 38]]
↳ app/models/journal.rb:147:in `sort_by'
```

**Associated revisions**

Revision 22581 - 2023-12-31 03:02 - Go MAEDA

Cache the result of `Journal#attachments` (#39847).

Patch by Go MAEDA (@maeda).

**History**

#1 - 2023-12-25 15:23 - Go MAEDA
- Target version set to 6.0.0

#2 - 2023-12-31 03:02 - Go MAEDA
- Status changed from New to Closed
- Assignee set to Go MAEDA

Committed the patch in r22581.

**Files**

cache-journal-attachments.patch  646 Bytes  2023-12-14  Go MAEDA