

Redmine - Defect #37151

The done ratio of a parent issue may not be 100% even if all subtasks have a done ratio of 100%

2022-05-24 02:44 - Go MAEDA

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Go MAEDA	% Done:	0%
Category:	Issues	Estimated time:	0.00 hour
Target version:	5.0.2	Affected version:	
Resolution:	Fixed		
Description			
The steps to reproduce:			
1. Create an issue			
2. Create 11 subtasks for the issue			
3. Set the done ratio of all subtasks to 100%			
4. Set the estimated time for 10 subtasks to 10 hours and 1 subtask to 9 hours (this means that the total estimated hours will be 109 hours)			
parent.png subtasks.png			
Related issues:			
Related to Redmine - Defect #27848: The progress exceeding 99.5% is displayed...		Closed	
Related to Redmine - Defect #33576: Done ratio of a parent issue may be shown...		Closed	

Associated revisions

Revision 21626 - 2022-06-08 17:40 - Go MAEDA

The done ratio of a parent issue may not be 100% even if all subtasks have a done ratio of 100% (#37151).

Patch by Go MAEDA.

Revision 21628 - 2022-06-10 11:17 - Go MAEDA

Merged r21626 from the trunk to 5.0-stable (#37151).

History

#1 - 2022-05-24 03:07 - Go MAEDA

- Related to Defect #27848: The progress exceeding 99.5% is displayed as 100% added

#2 - 2022-05-24 04:02 - Go MAEDA

The following change works in the reported case.

```
diff --git a/app/models/issue.rb b/app/models/issue.rb
index ab9f794db..d957d5933 100644
--- a/app/models/issue.rb
+++ b/app/models/issue.rb
@@ -1851,7 +1851,7 @@ class Issue < ActiveRecord::Base
     estimated * ratio
   end
   progress = done / (average * children.count)
-  p.done_ratio = progress.floor
+  p.done_ratio = progress.to_f.floor
   end
 end
end
```

```
irb(main):029:0> done
=> 0.109e5
irb(main):030:0> average
=> 0.990909090909090909090909090909091e1
irb(main):031:0> children.count
```


As a slight optimization however, I don't think it is necessary to use `to_s` in the numbers before parsing the Rationals. I think `Kernel#Rational` also accepts Floats and Integers at least down to Ruby 2.4.

Thus, I believe this should be functionally equivalent to your patch in [#37151#note-5](#) above:

```
diff --git a/app/models/issue.rb b/app/models/issue.rb
index ab9f794db..519b26fd8 100644
--- a/app/models/issue.rb
+++ b/app/models/issue.rb
@@ -1838,19 +1838,20 @@ class Issue < ActiveRecord::Base
   if children.any?
     child_with_total_estimated_hours = children.select {|c| c.total_estimated_hours.to_f > 0.0}
     if child_with_total_estimated_hours.any?
-       average =
-         child_with_total_estimated_hours.sum(&:total_estimated_hours).to_d /
-         child_with_total_estimated_hours.count
+       average = Rational(
+         child_with_total_estimated_hours.sum(&:total_estimated_hours),
+         child_with_total_estimated_hours.count
+       )
     else
-       average = BigDecimal('1.0')
+       average = Rational(1)
     end
     done = children.sum do |c|
-       estimated = (c.total_estimated_hours || 0.0).to_d
-       estimated = average unless estimated > 0.0
+       estimated = Rational(c.total_estimated_hours.to_f)
+       estimated = average unless estimated > 0
       ratio = c.closed? ? 100 : (c.done_ratio || 0)
       estimated * ratio
     end
     progress = done / (average * children.count)
     p.done_ratio = progress.floor
   end
end
```

#8 - 2022-05-24 16:32 - Go MAEDA

Holger Just wrote:

As a slight optimization however, I don't think it is necessary to use `to_s` in the numbers before parsing the Rationals. I think `Kernel#Rational` also accepts Floats and Integers at least down to Ruby 2.4.

Thank you for optimizing the patch. But I found that it breaks an existing test.

Failure:

```
IssueSubtaskingTest#test_done_ratio_of_parent_with_completed_children_should_not_be_99 [test/unit/issue_subtasking_test.rb:250]:
```

```
Expected: 100
```

```
Actual: 99
```

```
rails test test/unit/issue_subtasking_test.rb:244
```

This is because the value of `Rational("8.1")` and `Rational(8.1)` are not the same. `Rational("8.1")` represents 0.8 while `Rational(8.1)` represents 8.09999999999999964472.... So, I think it is safe to convert a float value to string before passing the value to `Kernel#Rational`.

```
irb(main):001:0> Rational("0.1")
```

```
=> (1/10)
```

```
irb(main):002:0> Rational(0.1)
```

```
=> (3602879701896397/36028797018963968)
```

#9 - 2022-05-25 01:49 - Go MAEDA

Using `Float#to_d` instead of `Float#to_s` also works as expected. It may be better because using `Float#to_d` makes it clear that floating-point errors are taken into account.

```
diff --git a/app/models/issue.rb b/app/models/issue.rb
index ab9f794db..2b488c0fd 100644
--- a/app/models/issue.rb
+++ b/app/models/issue.rb
@@ -1838,19 +1838,20 @@ class Issue < ActiveRecord::Base
   if children.any?
     child_with_total_estimated_hours = children.select {|c| c.total_estimated_hours.to_f > 0.0}
```

```

-     if child_with_total_estimated_hours.any?
-         average =
-             child_with_total_estimated_hours.sum(&:total_estimated_hours).to_d /
-             child_with_total_estimated_hours.count
+         average = Rational(
+             child_with_total_estimated_hours.sum(&:total_estimated_hours).to_d,
+             child_with_total_estimated_hours.count
+         )
-     else
-         average = BigDecimal('1.0')
+         average = Rational(1)
-     end
-     done = children.sum do |c|
+         estimated = (c.total_estimated_hours || 0.0).to_d
+         estimated = Rational(c.total_estimated_hours&.to_d || 0)
+         estimated = average unless estimated > 0.0
+         ratio = c.closed? ? 100 : (c.done_ratio || 0)
+         estimated * ratio
-     end
-     progress = done / (average * children.count)
+     progress = Rational(done, average * children.count)
+     p.done_ratio = progress.floor
- end
- end

```

#10 - 2022-05-26 16:31 - Go MAEDA

- File 37151-v2.patch added
- Target version changed from Candidate for next minor release to 5.0.2

Setting the target version to 5.0.2.

#11 - 2022-06-08 17:40 - Go MAEDA

- Status changed from New to Resolved
- Assignee set to Go MAEDA
- Resolution set to Fixed

Committed the patch.

#12 - 2022-06-10 11:17 - Go MAEDA

- Status changed from Resolved to Closed

Merged to 5.0-stable in [r21628](#).

Files

parent.png	264 KB	2022-05-24	Go MAEDA
subtasks.png	171 KB	2022-05-24	Go MAEDA
37151.patch	2.58 KB	2022-05-24	Go MAEDA
37151-v2.patch	1.37 KB	2022-05-26	Go MAEDA