Philo - Bug #63

Tree modeling

11/09/2010 10:04 PM - Joseph Spiros

Status:	Resolved	**Start date:**	11/09/2010
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Core	**Estimated time:**	0.00 hour
Target version:			

Description

We currently use a naive adjacency list model for modeling the tree of nodes. We should probably switch to a more efficient implementation.

Existing implementations to consider:

https://tabo.pe/projects/django-treebeard/
https://github.com/django-mptt/django-mptt/

Related issues:

Related to Philo - Bug #30: Disallow tree infinite recursion	**Closed**	**10/27/2010**
Related to Philo - Feature #62: Navigation generation	**Resolved**	**11/09/2010**
Related to Philo - Feature #29: TreeModel get_branch/has_descendent methods	**Resolved**	**10/27/2010**

History

#1 - 11/11/2010 04:36 PM - Stephen Burrows

I like django-mptt a lot. Treebeard implements an adjacency tree, a materialized path tree, and an mptt. Adjancency is what we already have, materialized path is... very strange, in that it doesn't have parent/child foreign keys and "every step in the path has a fixed width and has no separators" - in other words, we wouldn't be able to store url patterns easily as far as I can tell. As for mptt - django-mptt's implementation is much better documented. I like that it maintains a parent ForeignKey, which can be used for ordinary join queries, while supplying the efficiency of lft/rgt queries.
Both mptt implementations use raw SQL for some functionality. I haven't been able find comments for either implementation as to what versions of which engines support the SQL used...

mptt is a complex enough system that I don't think it would be worth our while to make our own implementation, especially if we can incorporate one that already exists.

#2 - 11/11/2010 04:56 PM - Stephen Burrows

Incidentally, looking at Treebeard's benchmarking, they show stats for MySQL, PostGreSQL, and SQLite3 for both treebeard and django-mptt. In some situations, Treebeard's mptt implementation can be faster, particularly when it comes to unsorted moves and deletes. However, I'm don't know what version of django-mptt and treebeard were tested... Also, the point of nested sets is that maintainance costs are higher for the sake of faster reads. I'm curious how much of the speed of moves and deletes for treebeard is due to their use of a custom queryset with a delete() method as opposed to mptt's use of a model delete method only.

#3 - 11/15/2010 04:01 PM - Stephen Burrows

Implemented an mptt branch of philo - initial commit beb034eda30e8af8a169c4d43c5aba83812337e4. There's a great speed increase for get_path and get_with_path! Depending on the situations we're expecting, it might be possible to make additional optimizations, for example to QuerySetMapper.

#4 - 01/06/2011 10:11 AM - Joseph Spiros

- Status changed from Feedback to Resolved

This was resolved by using django-mptt, merged in commit commit: caafce88e26fef7f0cd5be751c1f0fa8bd0b5a06.