**Plan for isolating/removing the memory overflow problem that is creating instability in the site.**



1. Are similiar problems being reported in STACKOVERFLOW or on heroku's developer forum

[ — not on the developer forum ]

[ — StackOverflow? ... ]

2. Open an trouble ticket with heroku: priority: high: production site down [DONE]

3. create a reproducable test case and demonstrate that it is only occuring on the heroku production stack ( -- stack bamboo-ree-187 ) not on the development stack running on our development machines.

Will just browsing cause it? consistently, with the same path?

If so, build a mechanize task that will genereate that path.

4. Rebuild the application on heroku's 'cedar' stack

1. recompile using Ruby 1.9.3 [ DONE: sorting out problems this created ]

2. migrate to Rails 3.1.5 without enabling the asset path

3. migrate the mechanise script to 1.9.3/Rails 3.1+ and run it against this implementation

4. implement the asset path, and Sass precompilation of css assets

(Note: can't take this one into production unless/until we can migrate the soap4r and netsuite-client libraries to this compiler/application server pair. But if it does not have the problem on this more recent ( 'standard' as of June ) stack, it would probably be more worthwhile to invest engineering time porting these libraries than trying to work with heroku's support/engineering staffs to fix Rails 3.0. )

5. Are there any tools that will let us break down and analyze the memory image of a running Rails server? Merb team?

6. Try removing product and category images from the heroku/Amazon application servers: separate them onto Amazon S3 [DONE]

7. Try removing subsystems that run as separate Rails engines / Rake processes

—registration/login: devise

—netsuite interactions, soap library, client-side javascript libraries