D4Science Infrastructure - Task #5818

Identify log-jams in use-case source code

Nov 21, 2016 01:39 PM - Scott Large

Status:	Closed		**Start date:**	Nov 21, 2016
Priority:	Normal		**Due date:**	
Assignee:	Scott Large		**% Done:**	100%
Category:			**Estimated time:**	0.00 hour
Target version:	UnSprintable			
Infrastructure:	Development			

Description

For ICES Stock Assessment:

1) Identify specific log-jams in the use-case source code and identify functions (here, referred to as base functions) that can be modified to be more computationally efficient and alleviate the log-jams.
#1768 and #5517 both use optimization routines that run in serial.

#1768: uses FLCore::fmle() https://github.com/slarge/fmle_bb
#5517: uses a genetic algorithm: ga()

History

#1 - Mar 15, 2017 08:18 AM - Scott Large

- Status changed from New to Closed

- % Done changed from 0 to 100

Explored several examples of stock assessment code to identify specific log-jams (i.e., slow code, poor optimization routines, etc) in the code that could be optimized using e-infrastructure resources. Several log-jams were identified and the source code was explored. Most problems could likely be sped up efficiently using C++ (with R package RCpp), TMB (Template Model Builder), or vectorizing R code. I was not able to identify a single instance that would benefit from parallelization - further, parallelizing some of the code would be a massive undertaking. For most uses, the greatest advantage of the e-infrastructure will be in the data delivery, collaborative research, and perhaps the cloud computing should be viewed more as a side benefit rather than an objective.