Some suggestion for the mandatory assignment 3 (option 2). Please see this project as a possibility to combine your own research or study interests with Parallel Computing. The projects below are just some suggestions. Feel free to come up with your own ideas. Also use the course book to look for ideas.

- *Parallel Particle Simulation using far field forces*
  - Starting with: Ananth Grama, Vipin Kumar, Ahmed H. Sameh: Scalable Parallel Formulations of the Barnes-Hut Method for n-Body Simulations. Parallel Computing 24(5-6): 797-822 (1998)
  - Follow-up of mandatory assignment 2, possible by using a parallelization of Barnes-Hut algorithm, quad-trees should be known)
  - May not be accessible for analytical approaches
  - Maybe significant implementation requirements, 3 persons recommended (maybe 4 possible)
- *Stochastirator*
  - In Blackboard there is a subfolder "stochastirator". It contains the paper "Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels" by Michael A. Gibson and Jehoshua Bruck (J. Phys. Chem. A, 2000). The idea is to parallelize the sequential software "stochastirator" (also in Blackboard) or also another stochastic simulation program (like StochKit2). A straightforward way to do this would be to introduce "space" in the simulation.
  - Note that the parallelization of the exact method itself seems quite complicated to me, as a "priority queue" (the central data-structure used in the "stochastirator") is difficult to handle in parallel.
  - at least 2 students are necessary here
- *Dynamic Programming (DP)*
  - One example for parallelization of a DP problem is "Sequence Alignment" (but many other problems exist).
  - Start with reading DM813 Mandatory Assignment 1: Dynamic Programming and Sequence Alignment `http://www.imada.sdu.dk/~daniel/DM813/Assignments/mand1/`
  - Global / Local sequence alignment
  - Necessary: Reading Chapter 12 ("Dynamic Programming"), and using those methods

- *Lower Bound proof for the Unit Cluster Problem*
  - See `http://www.sciencedirect.com/science/article/pii/S0304397513005112` for the definition of the problem and a algorithm (i.e. an upper bound).
  - The idea is to find a *lower bound*. This will require to traverse a tree of unknown size (methods presented in Chapter 11).
  - If you are interested into online algorithms, that might be an interesting problem to solve.