**Blue Waters Petascale Semester Curriculum v1.0**

**Unit 9: Optimization**

**Lesson 2: Code Optimization Patterns**

**References / Further Reading**

*Developed by* *David A. Joiner for the Shodor Education Foundation, Inc.*



*Except where otherwise noted, this work by The Shodor Education Foundation, Inc. is licensed under CC BY-SA 4.0. To view a copy of this license, visit*[*https://creativecommons.org/licenses/by-sa/4.0*](https://creativecommons.org/licenses/by-sa/4.0)

*Browse and search the full curriculum at*[*http://shodor.org/petascale/materials/semester-curriculum*](http://shodor.org/petascale/materials/semester-curriculum)

*We welcome your improvements! You can submit your proposed changes to this material and the rest of the curriculum in our GitHub repository at*[*https://github.com/shodor-education/petascale-semester-curriculum*](https://github.com/shodor-education/petascale-semester-curriculum)

*We want to hear from you! Please let us know your experiences using this material by sending email to* [*petascale@shodor.org*](mailto:petascale@shodor.org)

There is an excellent description of loop and compiler optimizations available from the canonical text by Severance and Dowd. Unfortunately the text has not been updated for modern hardware and languages, but the principles are still sound. Chapter 4 sections 1 and 4 speak specifically to optimizing compilers and loop optimizations.

It is available as a community commons license.

<https://cnx.org/contents/u4IVVH92@5.2:bEZZukPR@1/Introduction-to-the-Connexions-Edition>

While memory hierarchy numbers are constantly changing, Wikipedia’s entry on the memory hierarchy is surprisingly up to date.

<https://en.wikipedia.org/wiki/Memory_hierarchy>

Other thoughts: <https://wiki.gentoo.org/wiki/GCC_optimization>