EGU2020-9732
https://doi.org/10.5194/egusphere-egu2020-9732
EGU General Assembly 2020

Refactoring the Memory Access Pattern to Improve Computational Performance in NEMO

Italo Epicoco[1,2], **Francesca Mele**[1], Silvia Mocavero[1], Marco Chiarelli[1], Alessandro D'Anca[1], and Giovanni Aloisio[1,2]

[1]Euro-Mediterranean Centre on Climate Change, Foundation, Italy ({francesca.mele, silvia.mocavero, marco.chiarelli, alessandro.danca}@cmcc.it)

[2]University of Salento, Dep. Engineering for Innovation, Lecce, Italy ({italo.epicoco, giovanni.aloisio}@unisalento.it)

In the roadmap of modern parallel architectures development, the computing power of a node grows much more quickly than main memory performance (capacity, bandwidth). This leads to an even much higher gap between computing and memory resources. An efficient use of the cache memory is becoming ever more essential as optimization technique.
The NEMO model uses a finite difference integration method and a regular cartesian grid for space discretization. The NEMO code reflects this choice: a generic field is represented in memory as a 3D array; and the code is mainly composed of three-level nested loops. These loops often include only a few operations in the body; the results are stored in a temporary 3D array and then used in subsequent loops until the final calculation.
The aim of this work is to make better use of the cache memory by fusing DO loops together. The loop fusion is a transformation which takes two or more adjacent loops that have the same iteration space traversal and combines their bodies into a single loop.
The fusion of the loops is not trivial, and it could require introducing additional redundant operations to solve data dependencies. Unfortunately, this leads to a drawback of the overall performance. To avoid the redundant operation, we can adopt pointers to arrays and implement a pointer rotation at each loop iteration.
We have developed the loop fusion transformation in an advection kernel extracted from the NEMO oceanic model. We have compared 3 different versions of the optimized advection kernel, with 3 different levels of loop fusion.
The first prototype refers to the implementation where the extreme fusion is applied, and all loops in the routine have been fused. In this version, the operations are replicated up to 3 times. In the second prototype the buffer rotation has been applied only in the outermost loop. In the third prototype, the buffer rotation has also been implemented for the second dimension, and this version introduces only a limited amount of redundant operations.

The tests have been performed on the Athena cluster located at the CMCC supercomputing center. The supercomputing infrastructure is based on the Intel Xeon E5-2670 processors. The memory hierarchy is composed of 32KB of L1 cache, 256KB of L2 and 20MB L3 cache shared among the cores. The results clearly proved the effectiveness of the loop fusion approach that

reaches a speedup of 2x with a high number of cores. The third prototype has proven to be the most promising solution. Prototypes 1 and 2 provide a good improvement up to 256 cores then the redundant operations lead to a loss of performance.
A deeper analysis measuring the Last Level Cache misses also showed how the loop transformation significantly reduced the number of cache misses.
Despite the good results achieved with the loop fusion optimization, we can remark that this optimization is strictly linked to the computing architecture. A fully portable performance improvement can be ensured by the adoption of a DSL (Domain Specific Language).