

Real Interoperability Among Libraries

Anshu Dubey and Hans Johansen

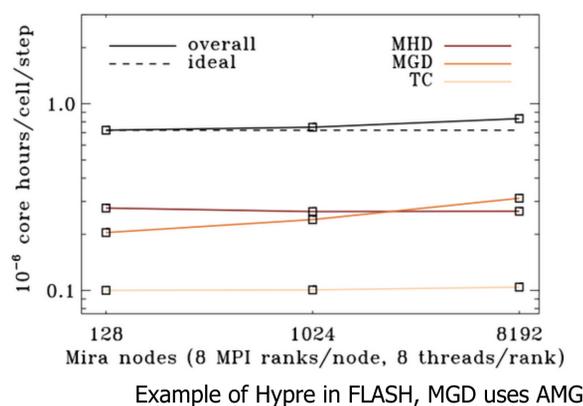
Many scientific and numerical libraries available today offer a wide range of well designed and robust solvers. However, using any combination of a subset of solvers from multiple libraries remains an elusive target. One possible way of mitigating this challenge is to focus on pairwise interaction of libraries and use the heterogeneity of resources to stage coupling between libraries.

Why Use Libraries

- ❑ Reduce development time : Offloading some of the development to the libraries
 - ❑ If there are libraries available that provide some of the needed solutions code can be developed and deployed faster.
- ❑ Easier parallelization and performance
 - ❑ If the data layout of the application is matched with that of the library, high performance computing and parallelization of the code may come for free.
- ❑ Quality of solution : Individual groups lack the expertise that developers of specialized libraries have
 - ❑ If the fidelity of solution matters, they have to either acquire expertise or use libraries
 - ❑ Resources are scarce, acquiring expertise in everything is not possible
 - ❑ Many specialized algorithms take years to develop, if they are in the libraries they become much more widely available for science
 - ❑ Re-inventing lower quality wheel over and over again is a waste of resources

Why Libraries Are Not Used As Much As They Could Be

- ❑ Library's data layout may be incompatible with that of the code
 - ❑ Wrappers work but take a performance hit

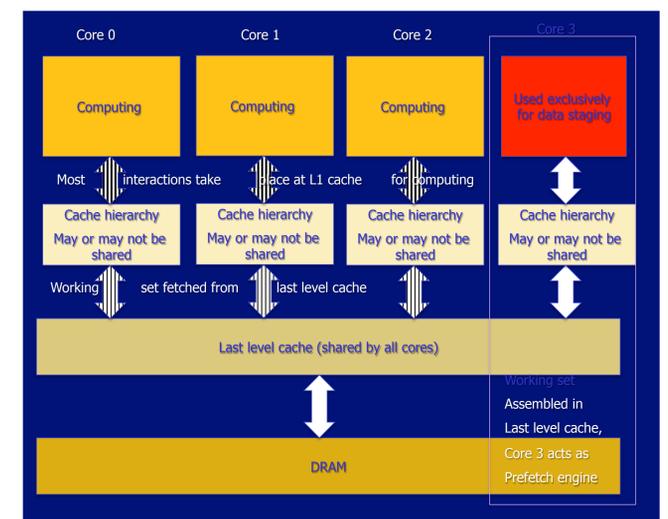


- ❑ Acquiring hard dependence on third party software
 - ❑ This is especially a challenge for users of high end HPC platforms
- ❑ Use of more than one library in the code can be very challenging
 - ❑ Challenges of building
 - ❑ Consistency in system software used to build
 - ❑ Different data layouts and assumptions about control
- ❑ Examples:
 - ❑ Chombo and PETSc scale, mapping from Chombo to PETSc does not
 - ❑ AMP effort can quickly create a new capable multiphysics code very quickly through combination of libraries, the resultant code is good for prototyping, but taking the same code to production requires a great deal of custom optimization
- ❑ Library based solutions have found a lot more use among codes that use low to mid-level high performance computing resources. Because of difficulties outlined above, not many leadership computing users use libraries.
- ❑ Node level heterogeneity makes this problem harder
 - ❑ Some codes are dropping libraries because of threads interoperability

How to make libraries more attractive to high end application users

- ❑ The first hurdle to cross is difficulty on version matching and build
 - ❑ There are resource to help with that
 - ❑ Examples:
 - ❑ A document compiled by Mark Miller
 - ❑ <https://wci.llnl.gov/codes/smartlibs/UCRL-JRNL-208636.pdf>
 - ❑ The IDEAS project is focused on unifying and simplifying the build process for the participating libraries.
 - ❑ <https://ideas-productivity.org/focus-areas/xsdk/>
 - ❑ Parallel interoperability is another focus area

- ❑ Exploit heterogeneity for data interoperability among libraries
 - ❑ Data staging is a well known concept in I/O and analysis
 - ❑ Selective prefetching is used to bring data that might be next needed by the computation
 - ❑ Working set assembly can utilize both of these concepts
 - ❑ Libraries can interchange their working sets on a pair-wise basis
- ❑ Libraries tend to operate in clusters because of the nature of physics that applications are addressing
 - ❑ Examine the use-pattern to identify library pairs that occur together in applications most frequently.
 - ❑ Set aside some resources on the node to stage the data to be transferred between libraries as needed.
 - ❑ It can be viewed as equivalent of assembling working set between steps.
 - ❑ See figure below for an example of data staging



- ❑ Working sets can be assembled using last level cache in the memory hierarchy and dedicated cores as pre-fetch engines
 - ❑ Assuming that working sets can fit in other cache levels