INTRODUCTION
The context of the research is a co-design project that has the goals of designing hardware systems to match application requirements and mapping applications to the hardware efficiently. To determine application requirements, we characterize the application using platform-independent locality metrics. Next, we use locality data to predict cache performance of sequential versions of the applications for various cache configurations. The prediction code that we are using initially is LULESH benchmark, which serves as a proxy for full shock physics applications CTH and ALEGRA. We can use a straightforward analytical model to predict cache misses for a fully associave cache, and we can use a probabilistic model to predict cache misses for a set-associative cache. We got some initial results for LULESH benchmark too.

RESULTS
Results for cache configuration (Intel Nehalem-EP) of L1 cache size= 32KB, L1 cache line size= 64, L1 Associativity= 1ru, L2 cache size= 256KB, L2 cache line size= 64, L2 Associativity= 1ru, L3 cache size= 2048KB, L3 cache line size= 64, L3 Associativity= 1ru.

GOAL
The overall context of the CoDAASH hardware-software co-design project has the goal of designing hardware to match the requirements of computational chemistry and physics algorithms important to the materials sciences problems of interest. To design the optimal cache configuration for a given algorithm is our main goal.

APPROACH
Data Access Pattern:
Our first step is to evaluate tools for obtaining platform-independent locality metrics and to validate the results. We are evaluating the PMAC locality measurement tool from SDSC and the MACPO data access analysis tool from TACC. We also use the PAPI hardware counter library to sanity check results returned by these tools. The PMAC locality measurement tool instruments the application using PEBIL in order to measure reuse distances and strides for data accesses. It computes reuse distances per basic block rather than by data structure, but we are working with SDSC on refining the tool to obtain per data structure metrics. The MACPO data access analysis tool reports reuse distance per non-serial variable and also reports the strides with which these data structures are accessed. PAPI gives the information of different events related to cache memory like cache hits, cache misses etc.

In summary, we are performing the following steps:
1. Use the PMAC tool to get the spatial and temporal locality data for the application.
2. Use the PAPI tool to get the hardware counters.
3. Use the PAPI tool to get the hardware counters.
4. Use the PMAC tool to get the spatial and temporal locality data for the application.
5. Use the PAPI tool to get the hardware counters.
6. Use the PMAC tool to get the spatial and temporal locality data for the application.

CONCLUSION
In the poster, we report the results of our evaluations of locality measurement tool from SDSC. We also describe initial results from our characterization of the LULESH benchmark. Our next step will be to use this characterization to predict cache performance for different cache configurations.

ACKNOWLEDGEMENT
This material is based upon work supported by the Air Force Office of Scientific Research under AFOSR Award No. FA9550-12-1-0476