Resilient X10 over Fault Tolerant MPI

Sara Hamouda[1], Benjamin Herta[2], Josh Milthorpe[1,2], David Grove[2], Olivier Tardieu[2]
[1]Australian National University, [2]IBM T. J. Watson Research Center

Resilient X10

X10 is an APGAS programming language that is designed to provide a simple and clean programming model for developing scale-out applications.

As supercomputers grow larger, the Mean Time Between Failure reduces, and the need for writing fault tolerance applications becomes more critical.

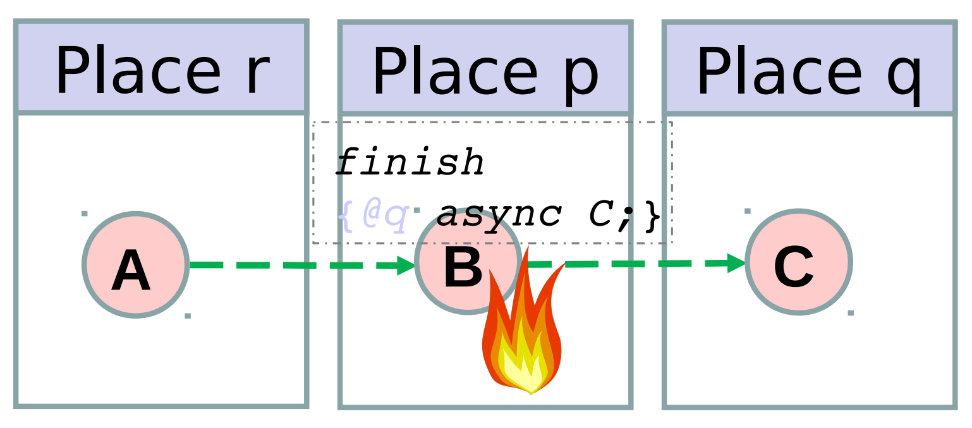
```
val wordCount = new AtomicInteger();
val refCount = GlobalRef(wordCount);
finish for (p in Place.places()) {
  val files = getFilesForPlace(p);
  at (p) async { //create task at place p
    val pCount = countWords(files, “ibm”);
    at (refCount.home)
      refCount().addAndGet(pCount);
  }
} print(wordCount);
```

Sample X10 program performing distributed word count

Resilient X10 [1] allows X10 programs to survive process failures.
By introducing the *Happens Before Invariance Principle*, it guarantees the correct repair of the global program structure after a failure.

Happens Before Invariance Principle (HBI):
Failure of a place should not alter the happens before relationship between statements at the remaining places.

```
try{ /*Task A*/
 at (p) { /*Task B*/
  finish { at (q) async { /*Task C*/ } }
 }
} catch(dpe:DeadPlaceException){ /*recovery steps*/}
D;
```



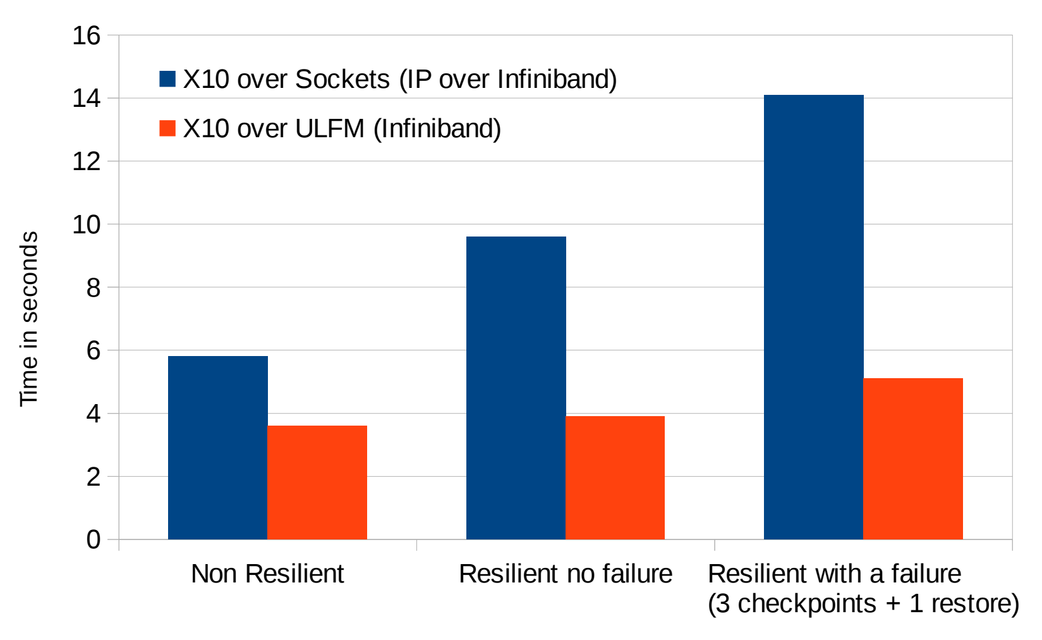
By applying the HBI principle, Resilient X10 will ensure that statement D executes after Task C finishes, despite the loss of the synchronization construct (finish) at place p

Resilient X10 over MPI ULFM

Although MPI is the preferred transport layer for scale-out computing, Resilient X10 was initially supported only over sockets.

ULFM (User-Level Failure Mitigation) is the most recent proposed specification for fault tolerant MPI [2]. An implementation of ULFM is available based on OpenMPI 1.7.

We integrated X10 with ULFM to allow Resilient X10 applications to benefit from the scalability and performance of MPI.



The performance improvement due to using ULFM v1.0 for running the LULESH proxy application [3] (a shock hydrodynamics stencil based simulation) running on 64 processes on 16 nodes with problem size 20^3 per process. The cluster is an AMD64 Linux cluster, each node having 16G RAM and 2 quad core AMD Opteron 2356 processors.

Conclusion: Using a fault tolerant MPI implementation (ULFM), resilient X10 applications can achieve better performance with the optimized MPI communication routines and the support for high speed network protocols provided by MPI (e.g. Infiniband verbs).

References:
[1] D. Cunningham, D. Grove, B. Herta, A. Iyengar, K. Kawachiya, H. Murata, V. Saraswat, M. Takeuchi, and O. Tardieu. "Resilient X10: Efficient failure-aware programming." ACM SIGPLAN Notices 49, no. 8 (2014): 67-80.
[2] W. Bland, A. Bouteiller, T. Herault, J. Hursey, G. Bosilca, and J. J. Dongarra. An evaluation of user-level failure mitigation support in MPI. Springer Berlin Heidelberg, 2012.
[3] J. Milthorpe, D. Grove, B. Herta, and O. Tardieu. Exploring the APGAS programming model using the LULESH proxy application. In Runtime Systems for Extreme Scale Programming Models and Architectures Workshop, SC 2015.

THE AUSTRALIAN NATIONAL UNIVERSITY