Rule The Next Generation Supercomputers With X10
Vivek Kumar

Why a New Language?
- Frequency Wall – Inability to follow past frequency scaling trends.
- Memory Wall – Inability to support a coherent uniform-memory access model with reasonable performance.
- Scalability Wall – Inability to utilize all levels of available parallelism in the system[1].

What is X10?
- X10 is a new language developed in the IBM PERCS project as part of the DARPA program on High Productivity Computing Systems (HPCS)[2].
- X10 is an instance of the APGAS framework in the Java family.

Why X10?
- Is more productive than current parallel programming models as well as more convenient and accurate than Java.
- Can support high levels of abstraction.
- Can exploit multiple levels of parallelism and non-uniform data access.
- Is suitable for multiple architectures, and multiple workloads.

X10 Activities (Threads)
public static def main(args:Rail[Str])
val sums = Rail.make[Int](2, (Int) => 0);
finish {
  async
  { sums(0) = sum(1, 100, (i:Int) => i*i); }
  async
  { sums(1) = sum(1, 1000, (i:Int) => i); }
}
val t = sums(0) + sums(1);
x10.io.Console.OUT.println("t=", t);

X10 Places (Processes)
def addTo(a:DistArray[Int], b:DistArray[Int])
  {a.dist += b.dist;}
val D = a.dist;
for(p in D.places())
  for(i in D.get(p))
    a(i) += b(i);

Research Areas – X10 Runtime
- Jikes RVM: Jikes Research Virtual Machine[3] is implemented in the Java™ programming language, which runs on itself without requiring a second virtual machine. Ongoing work for extending Jikes RVM as X10 Java runtime.
- MPI: Runtime support for Point to point communication in X10 code existing. Ongoing work for implementation of collective communication among X10 team object comprising of threads and processes.
- Cell & CUDA: Designing and implementing X10 runtime for Cell processors and CUDA architecture is also a very promising research area.

References

Supervised by Dr. Stephen Blackburn & Dr. Alistair P. Rendell