CS290B: Java-Centric Cluster & Concurrent Computing
John-Olav Storvold, Marianne Melhoos, Johannes Lier, Julie Rekdal
UCSB, Spring 2015

CX: A Scalable, Robust Network for Parallel Computing

The network-based computational exchange, CX, is designed to integrate variations of work stealing, non-blocking tasks, eager scheduling, and space-based coordination. It has a simple and compact API that separates application logic from the interprocess communication and fault tolerance logic.

The *computational model* models the constraint on networked computation and the compute producers that is available for short periods of time. Computation is modeled with a DAG similar to to Cilk threads.

The *programming model* says that all the applications should communicate with the system through the task server. The programmer sees the task server as a single server, although, the task server is a network of servers. A *Consumer* stores a task into the task server and receives the result after one of the *Producers* computes it.

Recovering from a server failure should complete in just a few seconds and without no human interaction. Making the system tolerable to another server failure should also require no human interaction and complete in under a minute.
The architecture consists of four entities: *Consumers* that places tasks on the task server to be computed. *Producers* that computes the tasks. *Task Server* that coordinates the tasks among the producers. And finally, the *Producer Network* which is a network of task servers and their producers.

Since network computations often are decomposed such that each task is sufficiently complex to minimize communication latency, CX is not very suitable for computations with short-latency feedback loops. Human interactions are also unwanted because they are too slow, expensive and too unreliable.

Java is the preferred language to write a CX network. First of all, because of the Java Virtual Machine (JVM). Second of all, the time the programmer saves by writing in Java makes up for the execution time lost compared to some other languages. Finally, many programmers like to program in Java and should therefore be taken into consideration.

To support the task graph model of computation, there is an isolated cluster that tolerates producer failure. This makes it possible to do a computation without any transactions. To tolerate server failures, the server state has to be recoverable, from for example a transaction log. This is designed by organizing the server network as a sibling-connected fat tree.

CX is a network-based computational exchange and can be used in many different environments implemented in Java. It has a simple and compact API, has a fat tree of servers that self-repairs and has a simple diffusion process for task distribution. To hide the large communication latency in networks, CX uses caching/replication and two levels of pre-fetching.