Notes on Distributed Programming Models – CSE 40822
Prof. Douglas Thain, University of Notre Dame, February 2016

Caution: These are high level notes that I use to organize my lectures. You may find them useful for reviewing main points, but they aren't a substitute for participating in class.

References:

- Andrew Tannenbaum and Maarten Van Steen, "Distributed Systems: Principles and Paradigms", Pearson, 2007.

Systems like Condor and Amazon can provision machines, store data, and run jobs. This is a very low-level view of the overall computation, not unlike assembly language, where the programmer is responsible for allocating registers, loading and storing from memory, and selecting particular instructions. To be productive, we need a higher level view.

Most large scale cloud applications are built around some sort of distributed programming model. The model gives a basic structure and set of expectations for both the application and the underlying resources, so that the programming can focus on the structure of the application.

Things to consider about a programming model:

API Semantics
Concurrency
Fault Tolerance

There are a handful of widely used distributed programming models. Most systems in use today can be classified into one of the following categories:

Submit-Wait (Work Queue)

```
id = submit(task);
(id,result) = wait();
remove(id);
```

Bulk Synchronous Parallel (SIMD, Hadoop)

```
apply( f(x), array a )
```

Message Passing (MPI)

```
send(msg,nodeid);
msg = recv(nodeid);
msg = recvAny();
```

Directed Graphs (Makeflow)

Assemble files and programs into a large graph.
Concurrency is achieved by running programs simultaneously.

Fault-tolerance is achieved by retry using saved files.

```
id = submit(graph)
(id,result) = wait();
```

Data Spaces (Linda)

```
put(item);
item = get(pattern);
```

Publish-Subscribe (DBus)

```
subscribe(topic)
publish(event,topic)
deliver(event)
```

Persistent Messaging (Rabbit MQ)

```
queue = createQ();
put(msg,queue);
msg = get(queue);
```

We have seen some examples of these already:

Makeflow: Directed Graph

Work Queue: Submit-Wait