Parallel program = Operator + Schedule + Parallel data structure

**OPERATOR FORMULATION**

- **Active element**
  - Site where computation is needed
- **Operator**
  - Computation at active element
- **Activity**
  - Application of operator to active element
- **Neighborhood**
  - Set of nodes/edges read/written by activity
  - Distinct usually from neighbors in graph
- **Ordering**
  - Scheduling constraints on execution order of activities
  - Unordered algorithms: no semantic constraints but performance may depend on schedule
  - Ordered algorithms: problem-dependent order
- **Amorphous data-parallelism**
  - Multiple active nodes can be processed in parallel subject to neighborhood and ordering constraints

**MULTI-LEVEL PROGRAMMING**

- **Joe Programmer:** Operator and Schedule Specification
- **Stephanie Programmer:** Parallel Data Structures
- **Obi-Wan Programmer:** Synchronization, NUMA, Scalable Runtime

**Native Performance**

Intel HPC Study: Galois implementations are comparable to hand-written and optimized code.

“Navigating the maze of graph analytics frameworks” Nadathur et al SIGMOD 2014

**Scaling**

Numa-aware runtime and data structures for high multi-core scaling. Real performance — single-thread performance is on par with optimized serial code.

**Think Beyond A Vertex**

The best algorithm may not be expressible as a vertex program
- Connected components with union-find
The best algorithm may require application-specific scheduling
- Priority scheduling for SSSP
Autonomous scheduling required for high-diameter graphs
- Coordinated scheduling uses many rounds and has too much overhead

**ONE MODEL, MANY TARGETS**

- **GPUs**
  - Optimized implementation strategies for coalescing, synchronization, high-thread counts
  - Scheduling and load balancing for highly-parallelism
  - Multi-gpu support
- **FPGAs**
  - Optimized scheduling and conflict detection
  - Targets network of FPGAs architectures
- **Distributed Memory**
  - Transparent support for distributed memory for arbitrarily complex irregular algorithms
- **Xeon PHI**
  - Numa and memory optimizations lead to out-performing simple pthread/openMP/openCL codes
- **Heterogeneous and emerging**
  - Mixed GPU/CPU
  - In progress: Cluster of multicore/CPU, coherent CPU/FPGA