MiST: A Platform for Mobile-Cloud Computing in Streams

Fan Yang, Zhengping Qian, Ivan Beschastnikh, Li Zhuang, Mao Yang, Amre Shakimov, Jacky Shen, and Lidong Zhou

Cloud: Unleash the Power of Mobile
Reduce the overhead of continuous operations
Leverage massive parallel computing resources

Stream as a First-Class Citizen
//stream construction, define “schema”
Stream<T> X = new Stream<T> (Query, interval);
//relational model: stream as table
//also fit well for data-parallel computing in cloud
var X = from x in X where P(x) select F(x);
//event-driven model: update triggers new ops
//annotation: execution place; scoping: scope of stream
[Cloud] var Z = when X or Y[latest 1 day]… select F(x, y);

Code Snippet: Continuous Route Planner
[Mobile] Stream<Location> locations = ...; //①
[Cloud] Stream<Traffic> traffic = ...; //②
[Cloud] Stream<Plan> plans =
    from l in locations combine t in traffic
    select MakePlan(l, t); //③
//consumption, functional reactive programming (FRP)
[Mobile] plans.Subscribe((Action<Plan>)Update); //④

Data Flow Graph (DFG)

Advantages of Stream Abstraction
Ease of application development
Combine imperative, declarative, and FRP languages
Sync stream hides the communication complexity
Leverage advances in distributed systems and database
Stream filter/slacks; Stream layout; Stream partition

Current Status
Implemented MiST and four applications
Validated the effectiveness of stream-based optimizations