Modeling Large-scale Networks with Flow Graphs

Dan Gunter
Martin Swany
Ezra Kissel

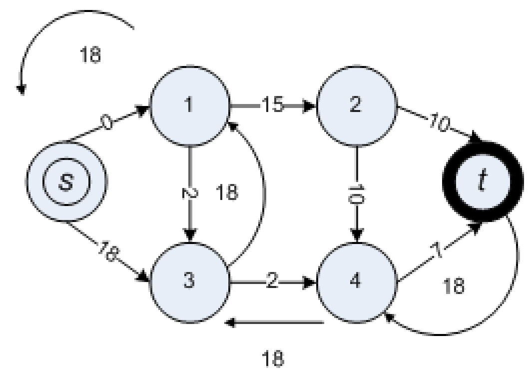
Lawrence Berkeley National Laboratory
Indiana University

COMBINE Workshop
September 11-12, 2012

Understanding large-scale networks

- Overcoming the complexity of the Internet
  - What can we say about performance today?
- What can we expect a model to answer?
  - Behavior of networks when scaled to terabit+ speeds
  - Location of bottlenecks from end-to-end perspective, meeting QoS goals
  - Network performance and function after dynamic reconfiguration, …
- Limitations of traditional simulation approaches
  - Packet-level discrete event – often too detailed, scalability issues
  - Equation-based fluid modeling – often too coarse, overly simplistic assumptions
- Comprehensive network model as a flow graph – just right?

Flow graphs



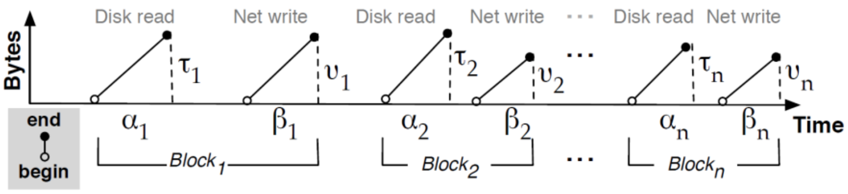
- Model computer networks as a flow network
  - Directed graph, each edge has a capacity and can receive a flow
  - Conservation of flow
  - Build upon well-known graph theory techniques
- Differentiate between types of flows
  - e.g., control and data traffic
- Allow for the modeling of end-to-end components as nodes in the graph
  - Beyond routers, NICs, and switches: backplanes, link from CPU to disk, etc.
  - Consider end-hosts: I/O devices and middleware
  - Where are the bottlenecks?
- Parallel graph algorithms for performance
  - Collaborate with Dr. Lumsdaine at IU (CREST)
  - Parallel Boost Graph Library (PBGL)
  - Optimize with prior work in high-performance MPI libraries (Gravel, Photon)

Characterizing the network

- Internet traces and measurements for broader network flow behavior
  - e.g., perfSONAR, Periscope, CAIDA, etc.

- Profiling devices
  - Empirical models for throughput behavior
  - Buffering, queue-depth, drop behavior => loss/congestion

- NetLogger-Calipers and active network probes
  - Build upon our prior work in understanding end-to-end performance
  - Micro-benchmarks



Model representation and inputs

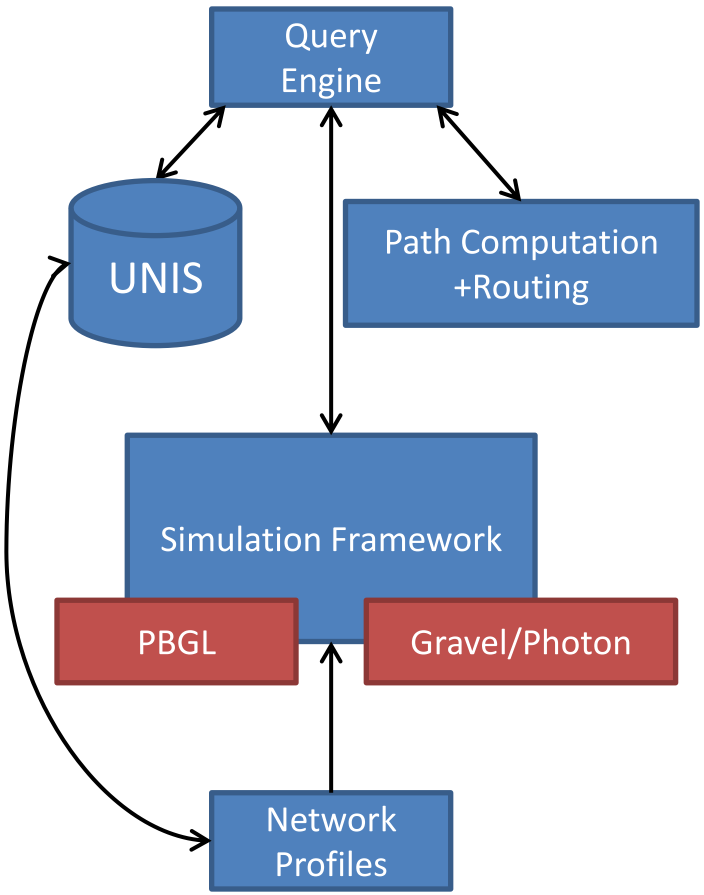
- Unified Network Information Service (UNIS)
  - A general data model for describing networks and measurements
  - Nodes, ports, links, services, paths, rules, and extensible annotations
  - Standard metadata/data format
- Routing and path computation
  - We can leverage existing databases (RouteViews, Skitter, etc.)
  - Explore scheduling and bandwidth on-demand scenarios
- RouteFlow from CPqD
  - Realizing IP routing services over SDNs
  - Extend to provide a general mechanism for explicit routing
  - A new input into our simulation model

Open research questions

- How complex are the computations?
  - Find upper and lower bounds
  - Performance analysis on existing DOE systems
- Temporal modeling of buffering or “flow accumulation”
  - Coarse time steps over pre-determined window
  - Modify continuous functions after effect is known
- Adaptive and dynamic models
  - Adjust granularity of the computation based on activity “hot spots” in the flow network
- Do we require explicit feedback?
- Can we model behaviors such as “lateral inhibition” (as in neurons)
  - Emergent phenomena from simple, basic rules

Summary

- A new application of existing parallel flow graph techniques
- Model larger scale, higher-granularity and more comprehensive end-to-end components
- Develop parallel graph algorithms and framework around existing and proven technologies
- Make use of "extreme scale" computing resources



Questions?

- Dan Gunter dkgunter@lbl.gov
- Martin Swany swany@iu.edu
- Ezra Kissel ezkissel@indiana.edu

- Parallel Boost Graph Library
  - http://www.osl.iu.edu/research/pbgl/

- Center for Complex Networks and Systems Research
  - http://cnets.indiana.edu

- RouteFlow
  - https://sites.google.com/site/routeflow/