Adaptive Data Structures for Networking Systems

Adaptive Methods

- **Challenge:** Networking systems are hard to program
  - Optimal solutions
    - hard to pre-determine (NP-complete, etc.)
  - often depend on the traffic patterns and thus change over time
- **Solution** – automated run-time adaptation to changes in
  - Input patterns
  - Execution environment
- **Adaptive methods**
  - Monitor the execution of processing tasks in a network element
  - Dynamically self-reconfigure (adapt) at run-time
- **What to reconfigure on the data plane?**
  - Resources re-assignment
  - Run-time code restructuring
  - Data structures adaptation

Adaptive Data Structures

- **Search methods**
  - Typically a tree-based data structure
  - Typically not optimized for the expected search keys
  - Key bottleneck: # of memory accesses = tree depth
- **Adapting to changes in input traffic patterns**
  - utilize locality in the search keys
- **If we knew the hit statistics, what could we do better?**
  - Modify or augment the tree according to the hits distribution
- **Challenges:**
  - Data structure monitoring
  - Data structure adaptation

Data Structure Monitoring

- Efficient statistics gathering:
  - gather maximum statistics with minimal overhead (low number of counters)

Adaptation - Network of Shortcuts (NoS)

- Introduce shortcuts into the search tree to reach frequently accessed nodes more rapidly
- Build a Network of Shortcuts (NoS) based on precomputed gain

Solutions

Example search (decision) tree:

certain paths travelled more often than others

Researchers:
Lukas Kencl, Andrea Bergamini

Collaborators:
Nils Kammenhueber, TU Munich

Research at Intel