Jumpgate: In-Network Processing as a Service for Data Analytics

Craig Mustard, Fabian Ruffy, Anny Gakhokidze, Ivan Beschastnikh, Alexandra Fedorova
University of British Columbia
In-Network Processing **Can Accelerate Data Analytics**

- **Smart NICs (FPGAs)**: 96% increased throughput [Floem]
- **Programmable Switches (P4)**: 2-8x speedup [NetAccel, DAIET]
- >1000x less traffic [Sonata]
There are many places to do In-Network Processing.
There are many places to do In-Network Processing

- Storage Cluster
- NIC
- Programmable Switch
- Alternative Data Path
- Software Middleboxes
- 4.5x speedup [NetAgg]
- Compute Cluster
- NIC
- Programmable Switch
- Off-path Opportunities
  - NPUs
  - ASIC/FPGAs
  - Ephemeral VMs
There are many places to do In-Network Processing

- Off-path
- Opportunities
- Ephemeral VMs
- ASIC/FPGAs
- NPUs

There are many places to do In-Network Processing. There is a 2-16x speedup on Apache Spark when performing filter, project, shuffle, aggregation somewhere in the network.

4.5x speedup on [NetAgg]
Challenges to actually using NPs

Target Devices
- Switches
- Smart NICs
- Ephemeral VMs
- N(etwork) PUs
- FPGAs
- D(ata) PUs
- Storage System

→ Tough to program:
  - Diverse hardware
  - Requires high performance software
  - Packet-oriented NOT flow-oriented
  - Storage limits (e.g., very little cross-packet state)

→ Manage multiple devices at the same time
  - Specialized devices not good at all parts of a query

→ Integration with storage and analytics systems
  - Need suitable protocols and data formats for NPs to operate on data

See our paper or come talk to me for details!
How should we incorporate solutions into systems?

**Target Devices**
- Switches
- Smart NICs
- Ephemeral VMs
- N(etwork) PUs
- FPGAs
- D(ata) PUs
- Storage System
How should we incorporate? One (bad) option:

Target Devices

- Switches
- Smart NICs
- Ephemeral VMs
- N(etwork) PUs
- FPGAs
- D(ata) PUs
- Storage System
How should we incorporate? One (bad) option:

- **Target Devices**
  - Switches
  - Smart NICs
  - Ephemeral VMs
  - N(etwork) PUs
  - FPGAs
  - D(ata) PUs
  - Storage System

**Problems:**
- Not scalable to all analytics systems
- Not future-proof to new devices
- Hard to share code
Our proposal: Network Processing as a Service

Target Devices
- Switches
- Smart NICs
- Ephemeral VMs
- N(etwork) PUs
- FPGAs
- D(ata) PUs
- Storage System

Network Processing as a Service (NPaaS)
Our proposal: Network Processing as a Service

Advantages:

➔ Abstracts devices and management
➔ Existing systems need to change once
➔ New devices and systems can be added easily
Jumpgate: a prototype NPaaS, addressing three problems

1. **Abstraction**
   - Client API
     - read data
     - filter
     - proj.
     - group by

2. **Programmability**
   - Compiler
     - Maps logical to physical ops.
     - Physical Plan

3. **Management**
   - Orchestrator
     - Deploys NP pipelines

**Available Physical Operators**
- Filter + Project in Storage
- Shuffle in Switch
- Partial Agg in SW

**Available Devices**
- Virtual Machines
- Switches
- NICs
- NPUs

**Deployment Constraints**
Jumpgate: example deployment

Storage Cluster -> NIC -> Programmable Switch

Filter + Project in Storage

Original Data Path

NPUs
ASIC/FPGAs
Ephemeral VMs

Jumpgate Data Path

Client API

read data
proj.
filter
group by

SQL

Storage Cluster

Programmable Switch

Shuffle in Switch

Programmable Switch

NPUs
ASIC/FPGAs
Ephemeral VMs

Partial Agg in SW

Compute Cluster
Open Questions:

We plan to use Jumpgate to investigate these questions and more.

➔ What are the right protocols and formats to use for different NPs?
  ◆ Protocols and formats are dependent on NP restrictions

➔ What are the best devices, and what is the best offload strategy?
  ◆ How to adapt existing query optimizations?

➔ How should we allocate devices w.r.t network topology?
  ◆ How much do we need to know about the topology to compute a good plan?

➔ Failure handling
  ◆ How should NPaaS interact with the client application on failures?
  ◆ Propagate to the client, or automatic recovery?
Takeaways:

➔ In-network processors can be **on-demand accelerators** for data analytics tasks.
➔ But, large **challenges** remain to using them.
➔ Instead of building solutions into **every** analytics framework, **we need NPaaS to provide abstractions** for using NPs.
➔ Jumpgate is our NPaaS prototype to address API, compilation, and orchestration challenges, and to enable future research in this area.

Thanks for listening! Happy to talk more!

Questions?