The Power of Compound Caches in the OVS Pipeline

How OVS achieves high performance for complex pipelines

Ben Pfaff
blp@ovn.org

Nov. 7, 2016

Software Switch Performance

Keys to high performance in a software switch:

1. Fast packet I/O.
2. Low per-packet overhead.
3. Low per-packet processing cost.

Distraction: kernel vs. userspace vs. mix

Not key to performance. It is incidental.

(DPDK in kernel?)

Key #1: Packet I/O Methods

Examples:

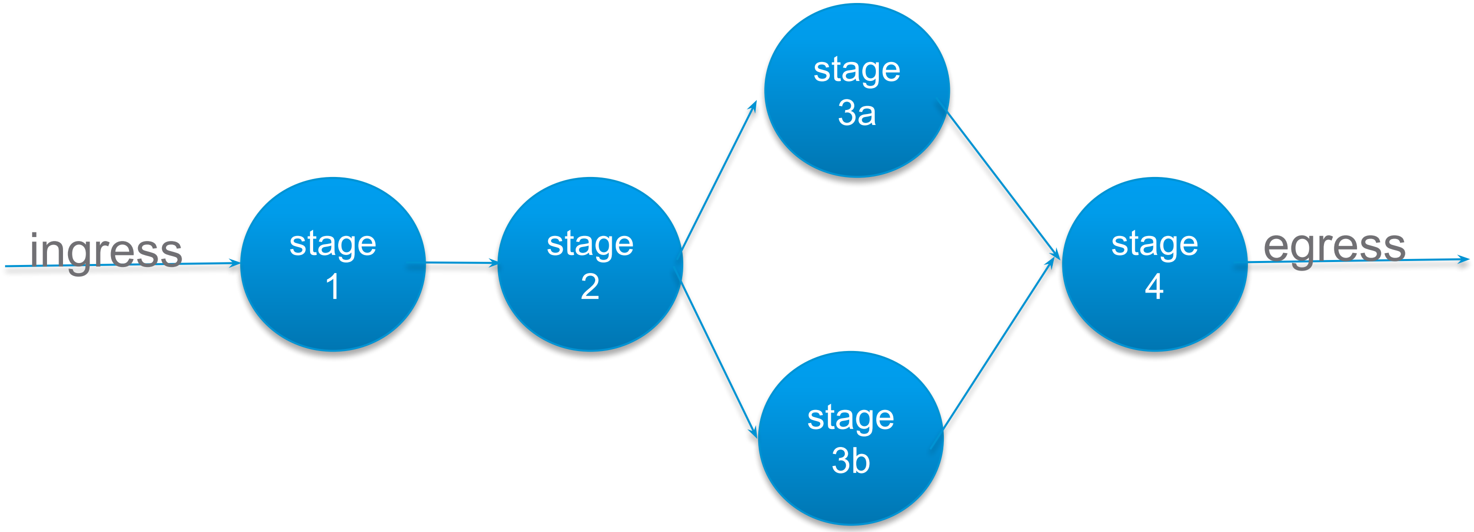
- AF_PACKET sockets
- Open vSwitch kernel module
- Netmap
- DPDK
- (e)BPF

Fast packet I/O is:

- Important
- Orthogonal to switch architecture
- Not comparable to complete software switches
- Not what I’m interested in today

Key #2: Low per-packet overhead

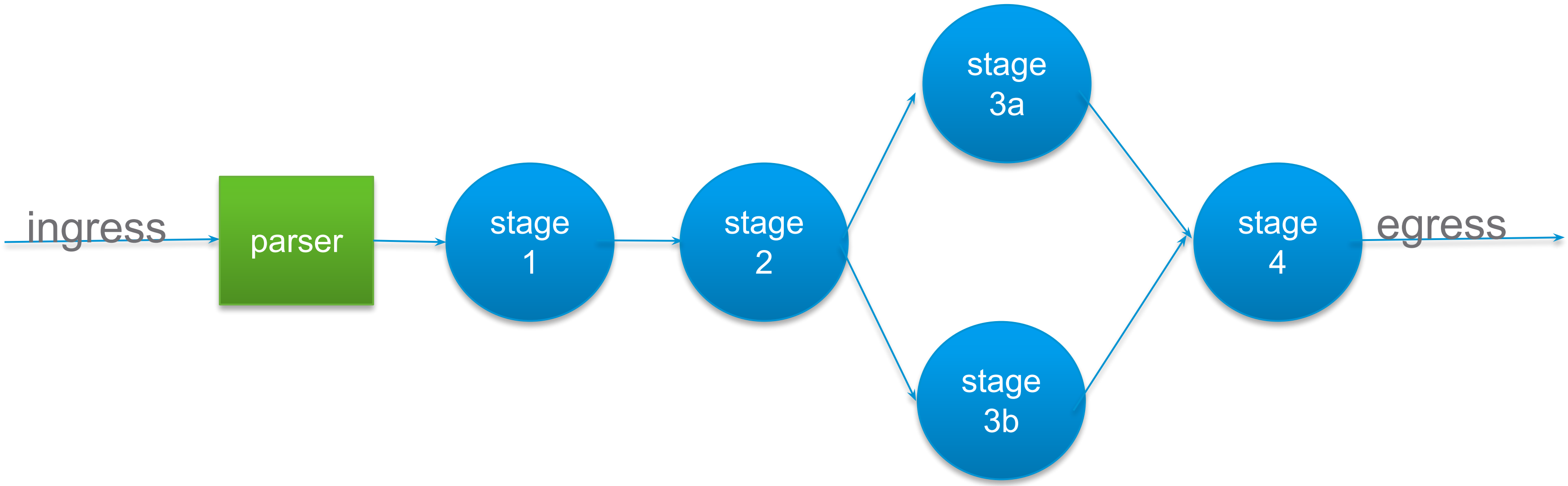
Common software switch organization:



Very low overhead.
Loose coupling.

Key #2: Low per-packet overhead

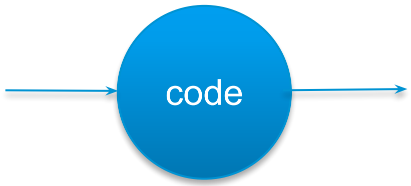
Open vSwitch adds a parser to the pipeline:



Oops. This increases overhead.
Bad idea?

Key #3: Low Packet Processing Cost

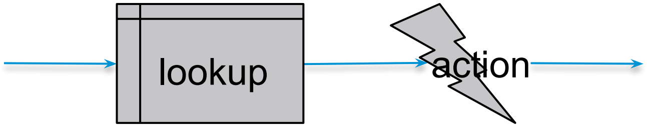
A stage is often just a function:



- Can do anything or nothing.
- No overhead.
- Commonly tested with “null pipeline”.

Key #3: Low Packet Processing Cost

Open vSwitch stages are expensive classifier tables:



- Classifier lookups are expensive.
- One lookup for every stage.
- Limited choice of actions.

So… OVS is Slow?

Two strikes against OVS performance:

- High per-packet overhead.
- High per-packet processing cost.

Therefore, OVS must be slow.

So… OVS is Slow?

Two strikes against OVS performance:

- High per-packet overhead.
- High per-packet processing cost.

Therefore, OVS must be slow.

Except: Caching

Caches in Networking Software

Caches have a long history in networking, e.g.:

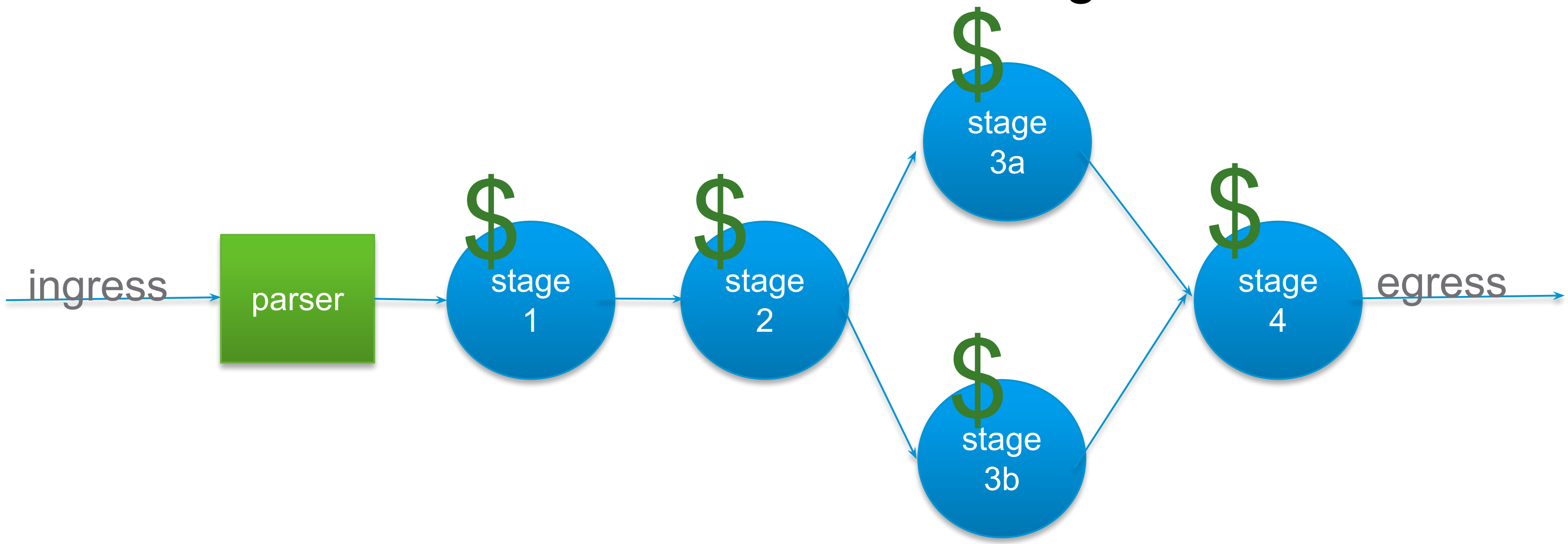
- MAC learning.
- ARP cache in IPv4 (and ND cache in IPv6).
- Route caches.

Caches also have a bad reputation in networking:

- Hit rate depends on traffic→unpredictable performance.
- Invalidation can be tricky.
- Most famously, Linux removed its route cache.

Simple Caching

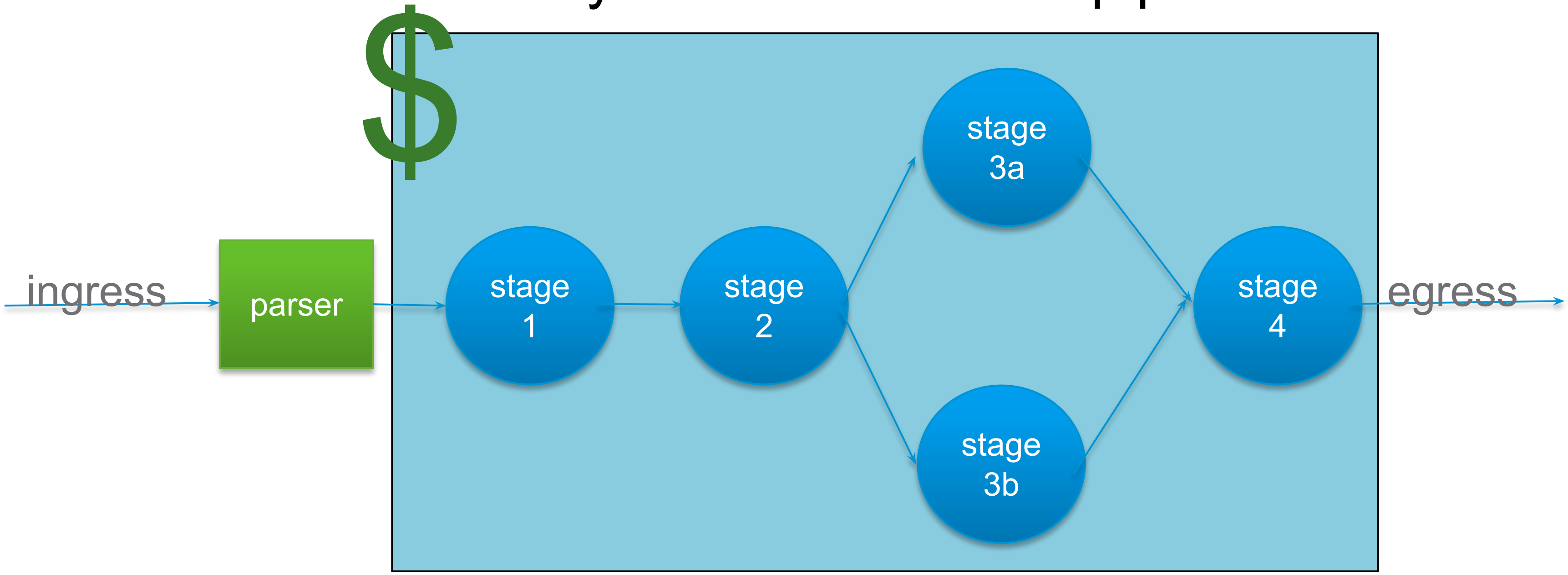
Add a cache to each stage:



Helpful, but works for any kind of switch, not just OVS.
So OVS still has extra overhead and will still be slower.

Compound Caching

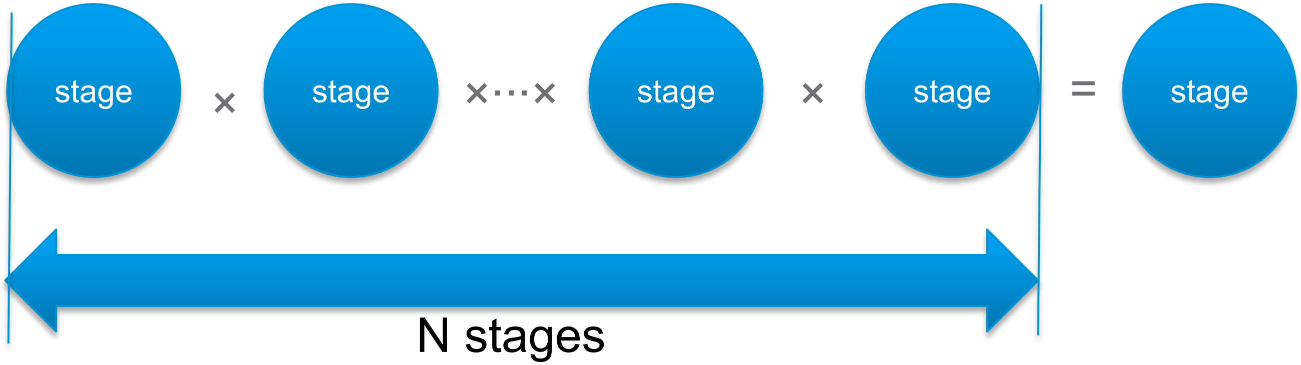
OVS actually caches the entire pipeline:



A single cache hit bypasses the entire pipeline.
Other software switches can’t do this trick.

More About Compound Caching

Cross-product N stages into 1 stage (see NSDI 2015 paper):



- High base per-packet cost (parsing + 1 classifier lookup).
- N stages costs per-packet about the same as 1 stage.
- Therefore: null pipeline is slow, complex pipeline is fast.
- Hardware classification offload is possible.

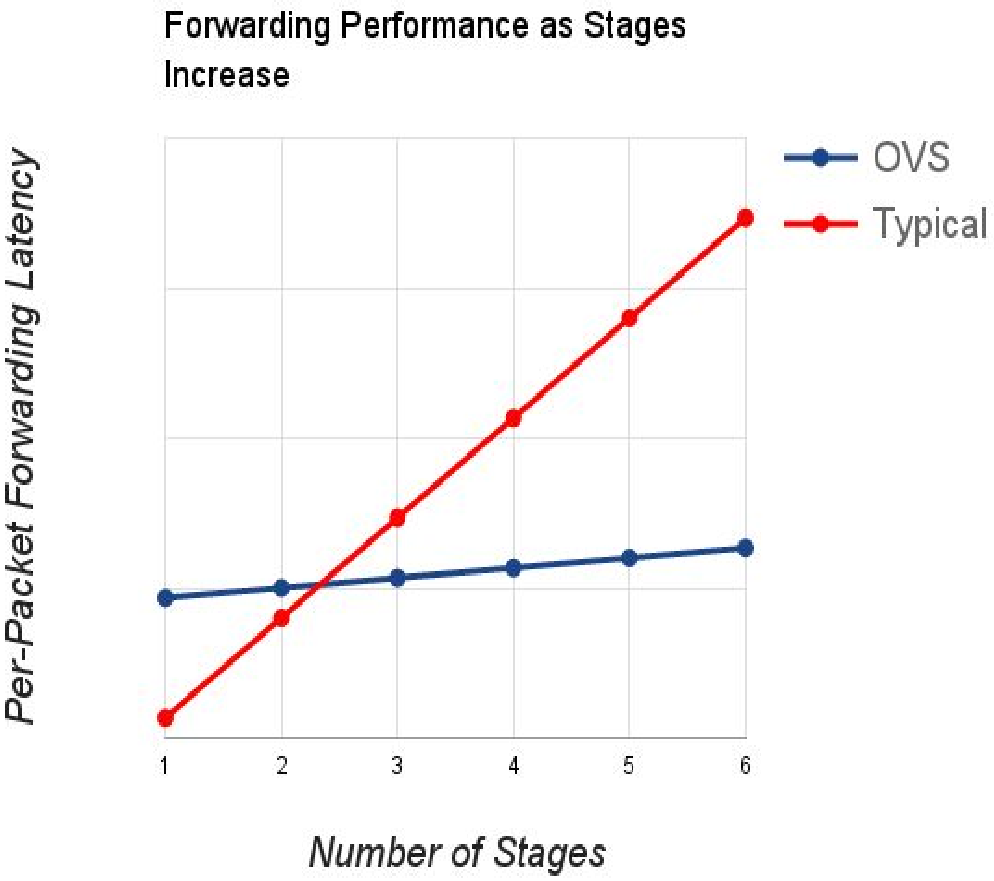
OVS Performance Comparison

Typical switches:

- Fast with few stages
- Slow as stages increase

OVS:

- Slow with few stages
- Not much slow as stages increase
- Cache hit rate is paramount so we’ve invested (see NSDI)



Final Thoughts

Could other switches adopt compound caches? Yes:

- Stages must record their read and write dependencies.
- Probably this requires stages to be rewritten.

Why are long, complex pipelines useful?

- Network virtualization: OVN, NSX, NVP

What’s a good benchmark?

- Hard to say.

Questions?

Ben Pfaff
blp@ovn.org

For more information about Open vSwitch, please visit openvswitch.org.