Observability Day
NORTH AMERICA

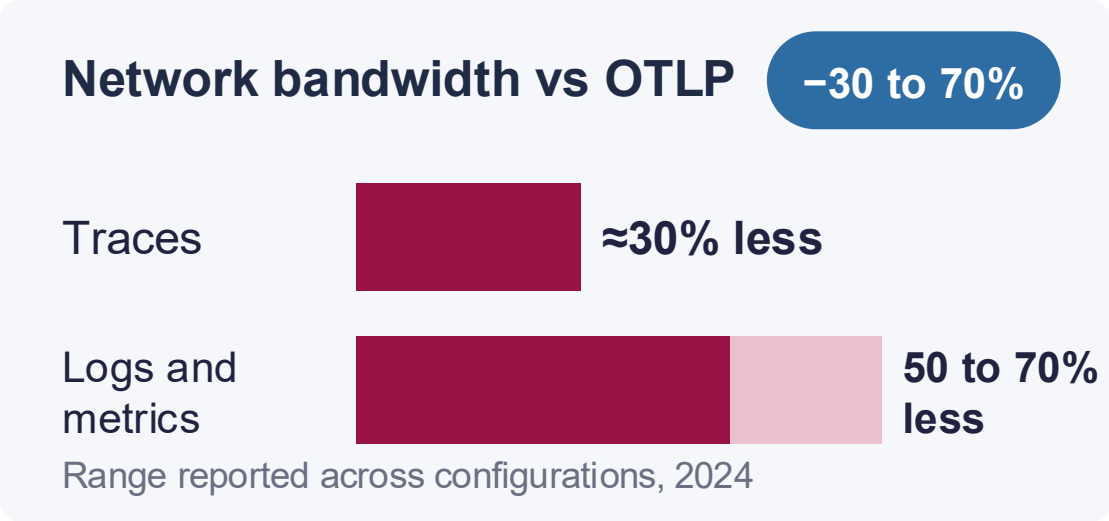
Rethinking Telemetry Pipelines

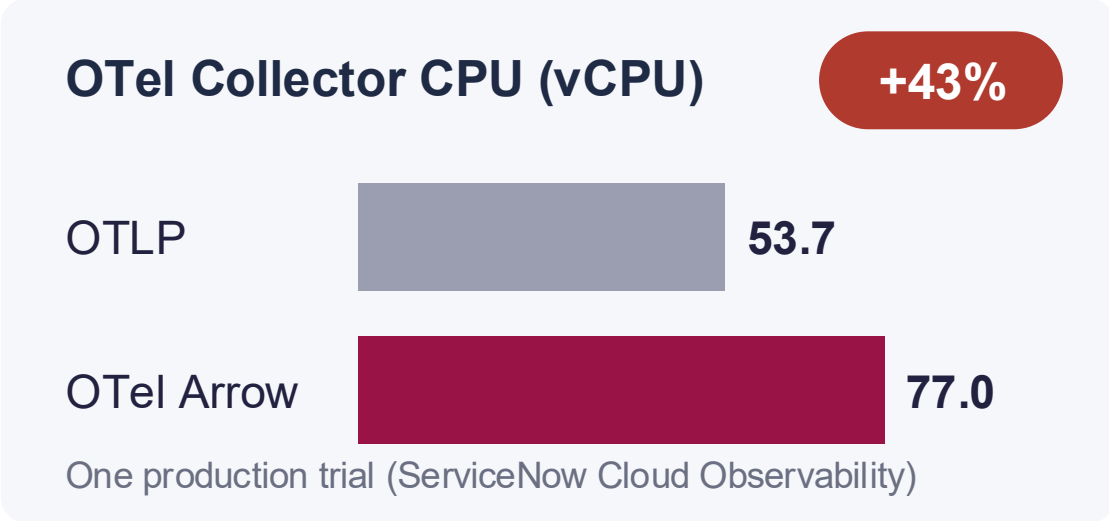
What happens when telemetry stays columnar end to end?

Laurent Quérel, F5 *Joshua MacDonald, Microsoft*

Columnar transport saved bandwidth but added CPU cost

Observability Day NORTH AMERICA





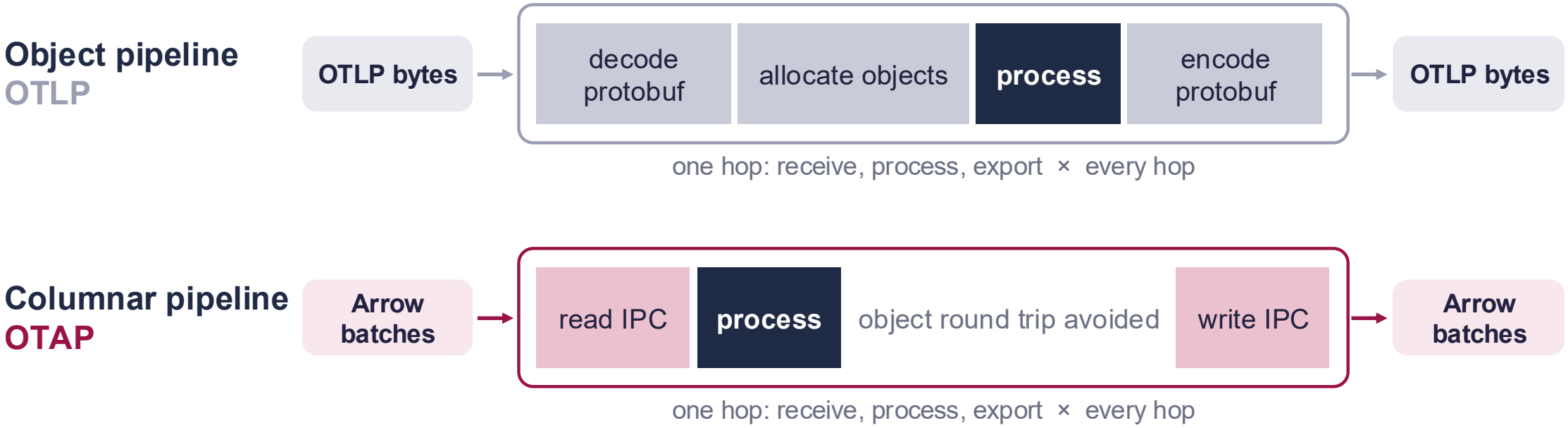
Why: the pipeline in between still worked on OTLP objects, so every batch was converted into Arrow at the exporter. Columnar stopped at the wire.

So what happens if telemetry never leaves columnar form?

Source: “OpenTelemetry Protocol with Apache Arrow in Production”, OpenTelemetry blog, 2024. CPU: ServiceNow Cloud Observability trial, OTel Collector v0.105.

Every hop pays a conversion tax

Observability Day
NORTH AMERICA



CPU per batch ≈ **conversion tax** + **useful work**

Illustrative, not to scale. Compression applies on both paths; IPC = Arrow stream reading and writing.

Same telemetry, stored as related columnar tables

Observability Day NORTH AMERICA

OTLP: a tree of objects, one per record

Resource › Scope › LogRecords



Pointer chasing: every arrow is a dependent load, often a cache miss, repeated per record

OTAP: Arrow record batches, linked by ID

Logs

id	severity
0	9
1	17
2	9

Log attributes

parent_id	key	value
0	0	/checkout
1	1	Timeout
2	0	/cart

severity column in memory

9	17	9	9	13	9	17	9

One contiguous buffer: vectorizable scans

Key dictionary

index	key
0	url.path
1	exception.type

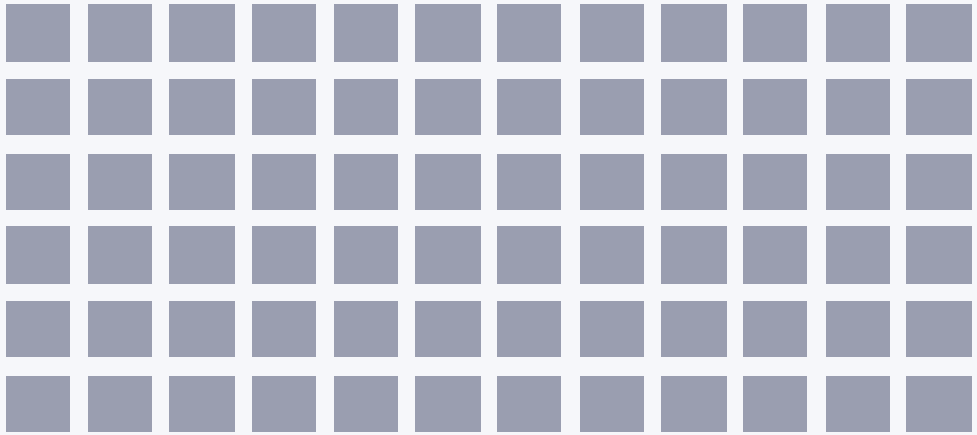
Lossless conversion to and from OTLP: logs, metrics, traces

Simplified: time, body, resource and scope tables omitted. See docs/data_model.md in the otel-arrow repository.

One rename can update a whole batch

Observability Day NORTH AMERICA

Object pipeline: visit every record



N records → **N attribute map updates**

OTAP: edit the key dictionary

idx	key
0	url.path
1	exception.type

→

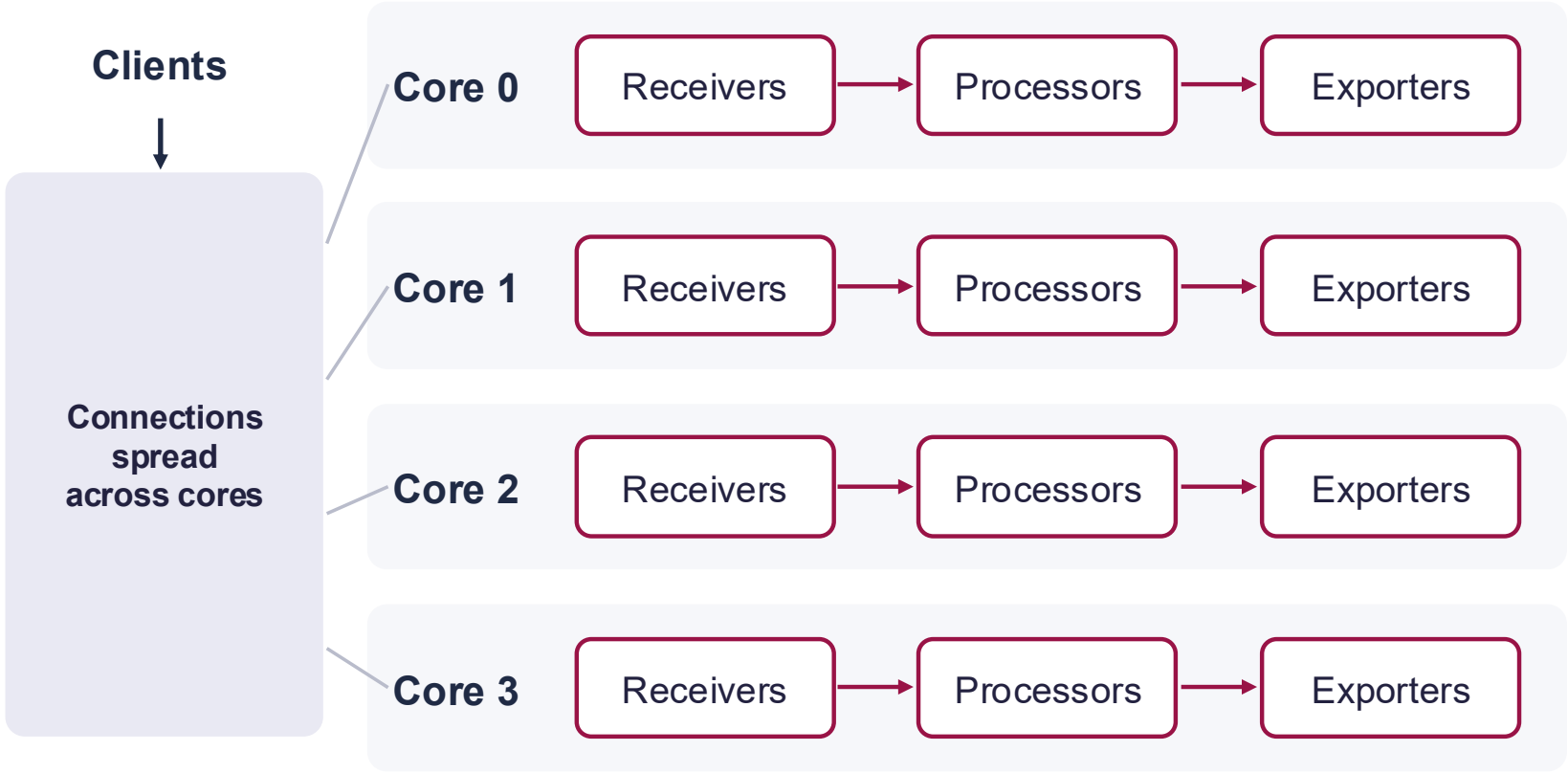
idx	key
0	url.path
1	exception.kind

Record indices and value buffers can stay unchanged

1 dictionary entry updated

Fast path when keys are dictionary encoded and the new key does not collide. Otherwise the key column is rebuilt, still in bulk.

Pipeline data stays on its core, from receive to export

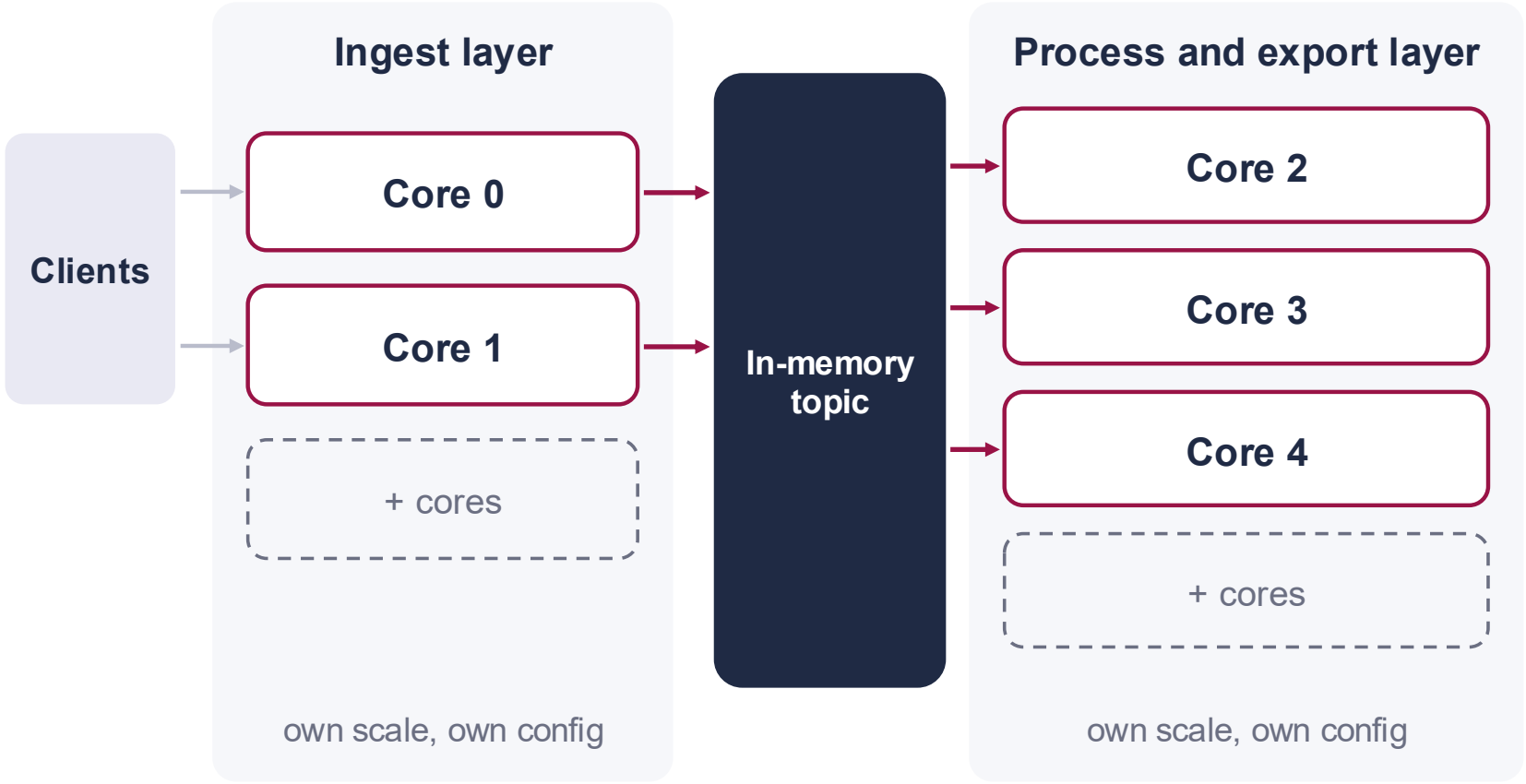


Pinned thread per core
with a local async runtime: no locks on the hot path

No implicit work stealing
a batch is processed on the core that received it

Thread-per-core, shared-nothing, in the spirit of Seastar. NUMA-friendly today (NUMA-aware placement is planned)

Crossing cores explicitly with an in-memory topic



Crossing cores is explicit
by design, never as a side effect of scheduling

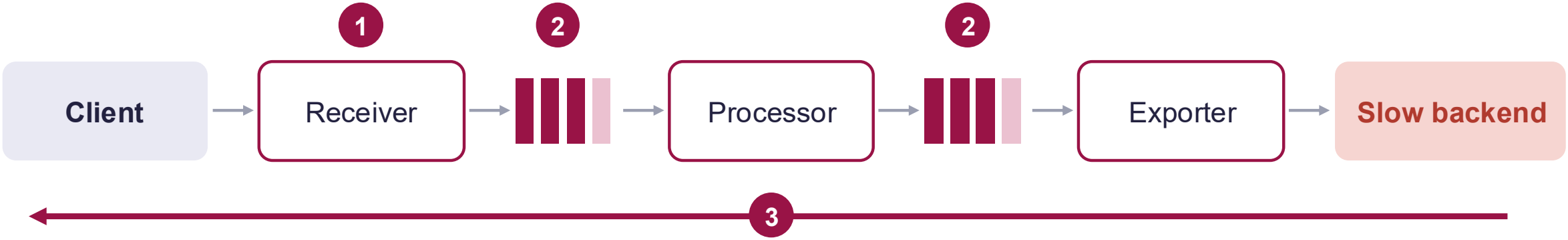
Live reconfiguration
replace or resize a pipeline through the admin API, core by core, no restart

Each layer evolves on its own: resize or reconfigure one without touching the other. A balanced topic sends each batch to one processing instance.

Topics are declared in the configuration and validated by the controller at startup. Ack and Nack can flow back across the hop.

A resource-bounded runtime: admit what fits, bound the rest, push back

Observability Day
NORTH AMERICA



1 Admit only what fits

- Receivers cap concurrent requests, request size and byte rate
- A process-wide memory limiter sheds ingress before decoding when memory runs hot

2 Bound everything inside

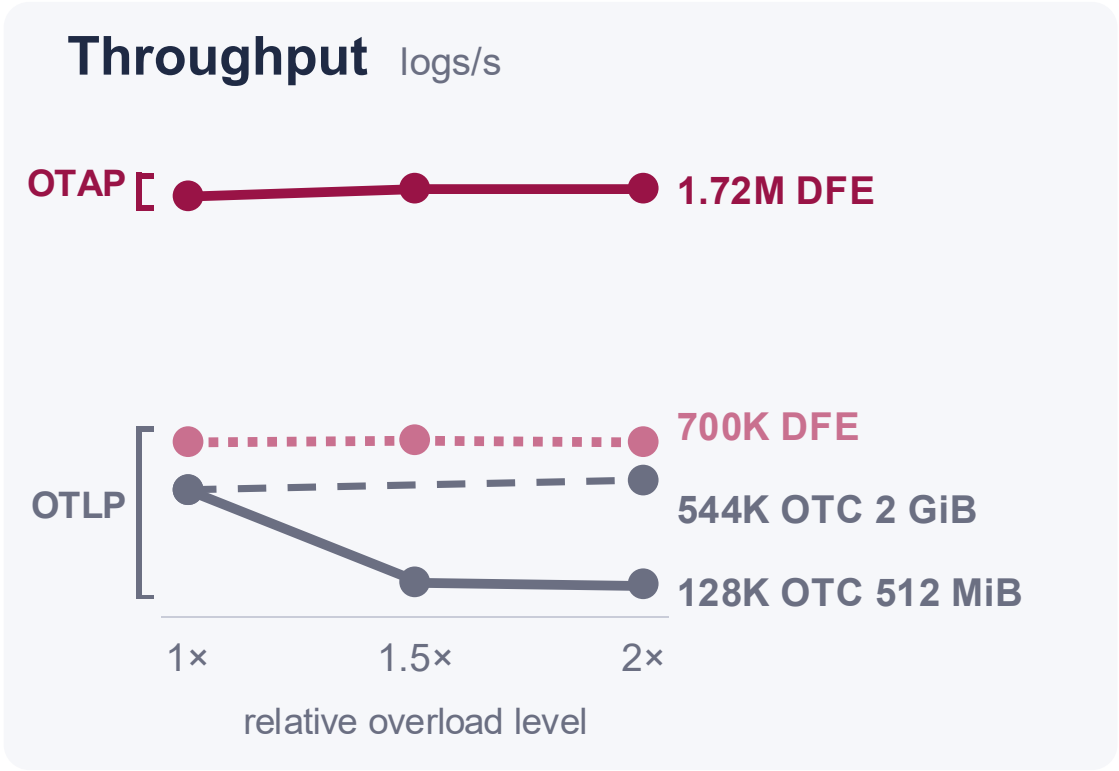
- Bounded channels and state, size-limited batches
- Rust ownership frees buffers at known points: no garbage collector, no GC pauses

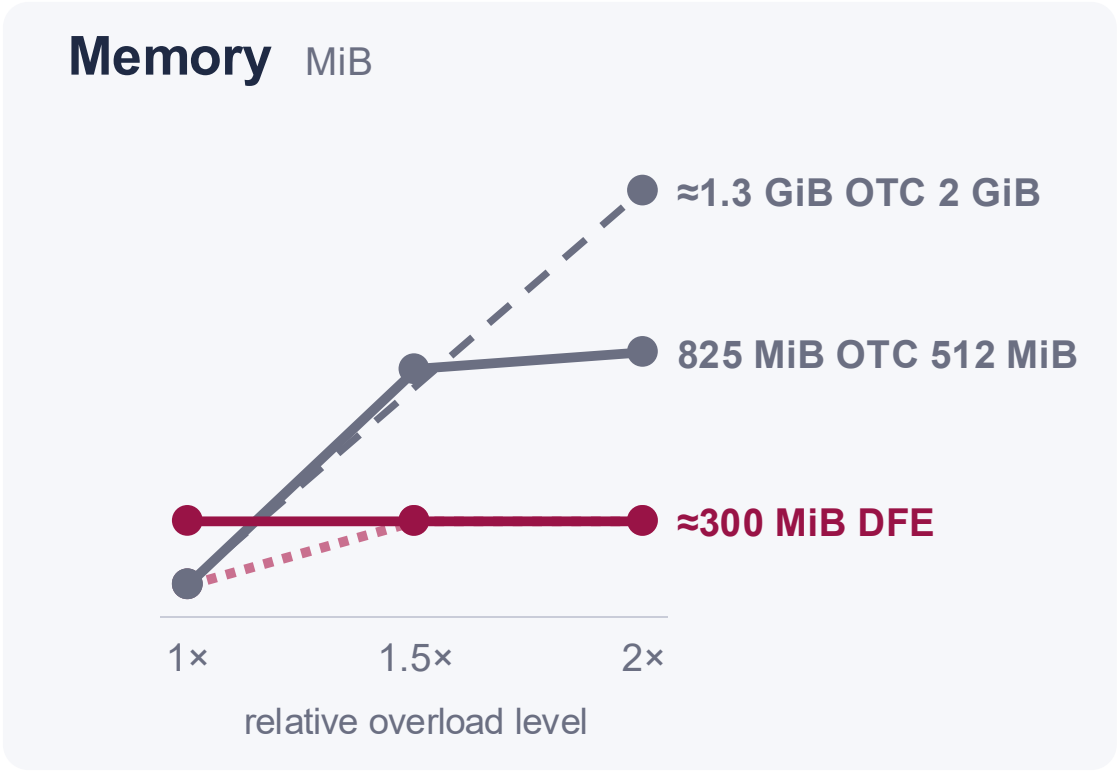
3 Push back, do not pile up

- A slow backend fills queues upstream instead of growing memory
- Receivers slow intake or answer with a clear status; Ack and Nack carry the outcome

Overload: DFE holds throughput and memory

Observability Day
NORTH AMERICA





OTLP input: OTC, 512 MiB limit — OTC, 2 GiB limit ····· DFE | **OTAP input:** DFE

OTC buys throughput with memory; DFE holds both without the trade.

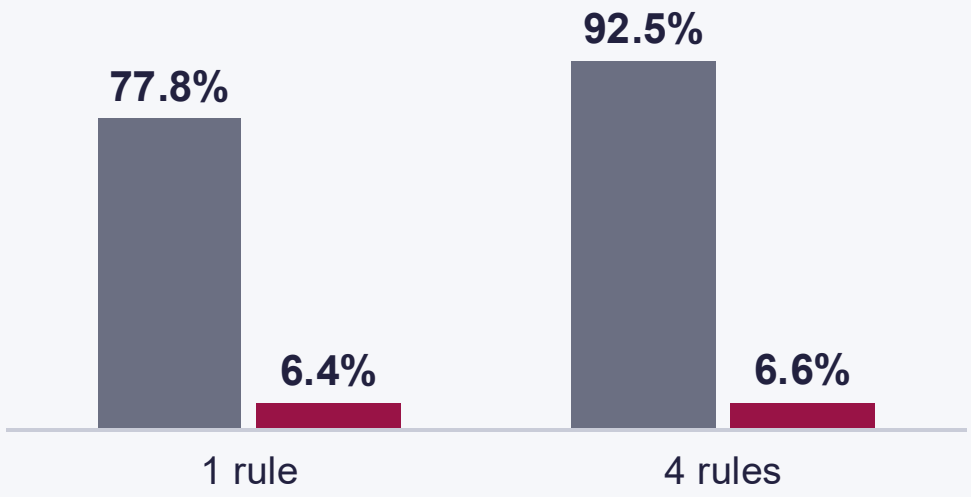
OTC: OpenTelemetry Collector. DFE: OTel Arrow Dataflow Engine. Phase 2, Figure 1, June 2026, ~300-byte logs, Xeon Platinum 8581C.

With columnar data, extra operations are cheap and batch size controls cost

Observability Day NORTH AMERICA

More rename rules 200K logs/s

CPU usage, % of a single core

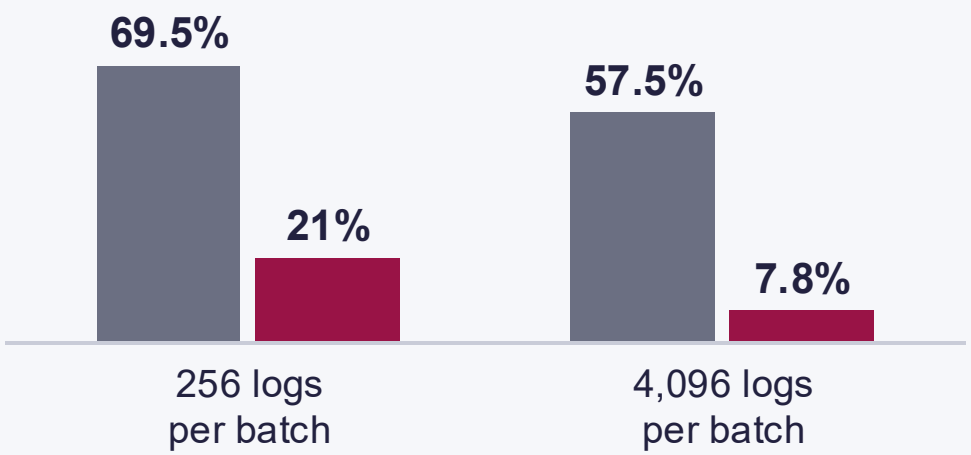


OTC +19% **DFE OTAP +3%**

CPU change with 3 more rules

Larger batches 400K logs/s

CPU usage, % of a single core



OTC −17% **DFE OTAP −63%**

CPU change with 16× larger batches

OTC, OTLP in memory as objects

DFE, OTAP end to end

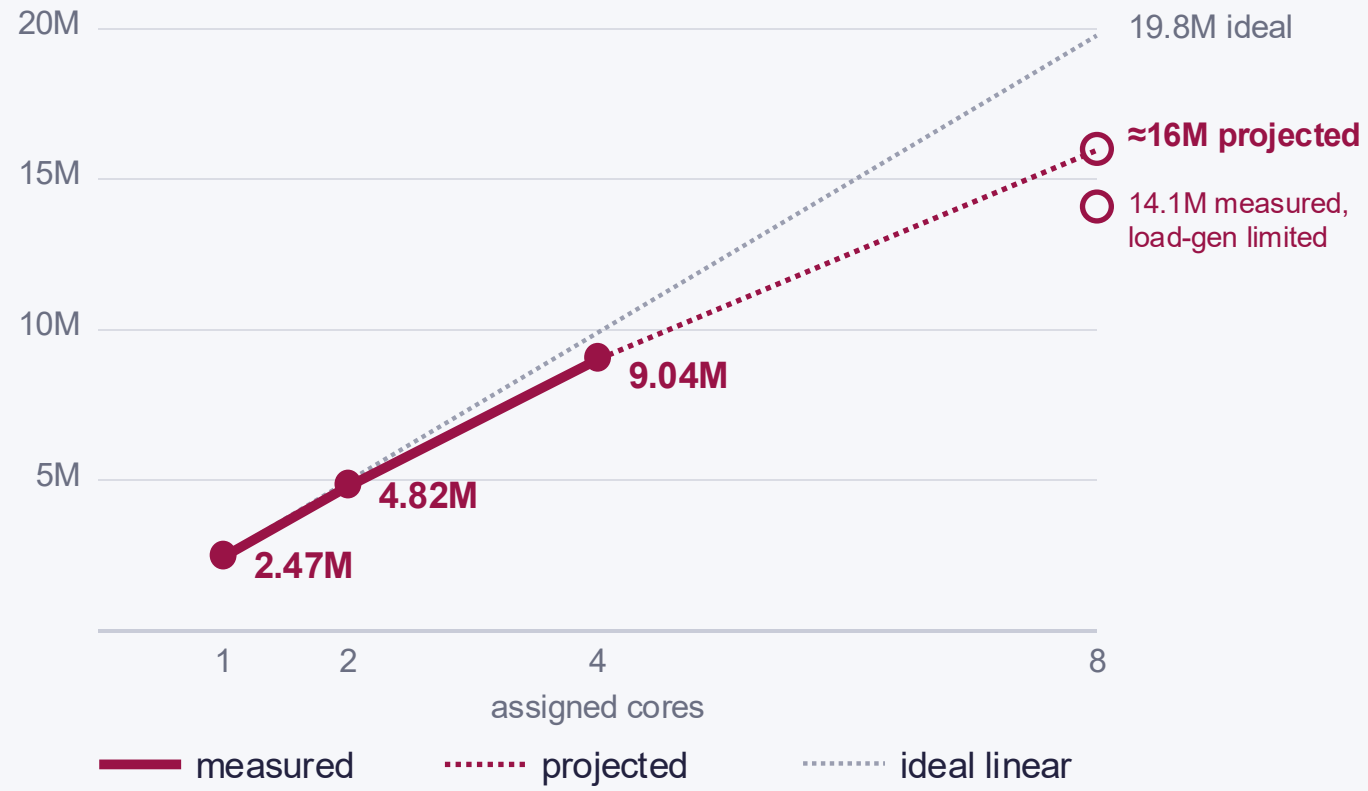
CPU usage of the single core running the pipeline, same scale on both charts. Source: OTel Arrow Phase 2 blog post, Figure 1, June 2026.

OTAP scales near-linearly to 9M logs/s on 4 cores

Observability Day NORTH AMERICA

DFE throughput, OTAP in and out

M logs/s, 1 KB logs



2.47M

logs/s on one core
the per-core starting point

3.7×

on 4 cores, measured
91% of linear scaling (98% on 2 cores)

≈16M

projected on 8 cores
the load generator ran out at 14.1M before the engine did

Phase 2, Figure 3, June 2026: 1 KB log bodies (512 unique), 64-core Xeon 8358. 8-core run load-generator limited; ≈16M is the post's projection.

Where columnar processing helps most, and where to measure first

Observability Day NORTH AMERICA

Strong fit

- High volume, where CPU and memory drive cost
- Same fields across many records: rename, filter, route, redact
- OTAP kept across several hops
- Batches large enough to amortize overhead

Measure first

- Small batches or tight latency budgets
- Per-value work dominates: parsing, hashing, regex
- Short pipelines with OTLP on both edges
- High-cardinality attributes

The Collector and the Dataflow Engine work together

Observability Day NORTH AMERICA

OpenTelemetry Collector

- General purpose, broad component ecosystem, in Golang
- Complete OpenTelemetry protocol implementation
- OTel Arrow receiver and exporter components included in the Contrib release



OTel Arrow Dataflow Engine

- Columnar end to end, thread-per-core, in Rust
- Incubating: APIs and configuration will change
- Ready for experiments, not for production yet

Interoperates with OTLP and OTAP senders and backends. The OTAP specification, the benchmarks and the Dataflow Engine are open source.

What to remember

Observability Day NORTH AMERICA

Telemetry pipelines can do less work when transport and processing share a columnar representation.

1. **Fewer representation changes**
   Keep telemetry columnar across hops and count the conversions on each one.

2. **Work shared across a batch**
   Rename, filter and route operate on columns and dictionaries, not record by record.

3. **Local execution, bounded work in flight**
   Thread-per-core and backpressure keep the gains under load.

Questions? Try it, break it, tell us

github.com/open-telemetry/otel-arrow

#otel-arrow on CNCF Slack

Bring workloads that stress it: small batches, high cardinality, heavy transforms

Observability Day
NORTH AMERICA

Backup

A1. Data-plane work in progress

Observability Day NORTH AMERICA

Capability	What it does	Status
Live reconfiguration	Roll pipeline changes core by core, roll back on failure	Implemented
Topics	Connect pipelines in memory, balanced or broadcast	Early: in memory only
Durable buffer	Persist batches for crash recovery	Experimental
OpAMP agent	Remote configuration and health reporting	Early
OPL and WASM extensions	Arrow-native transforms and custom logic	Experimental
Pluggable codecs	Carry encoded bytes, decode on demand	In development

A2. DFE OTLP path: 14.6× scaling across 16 cores

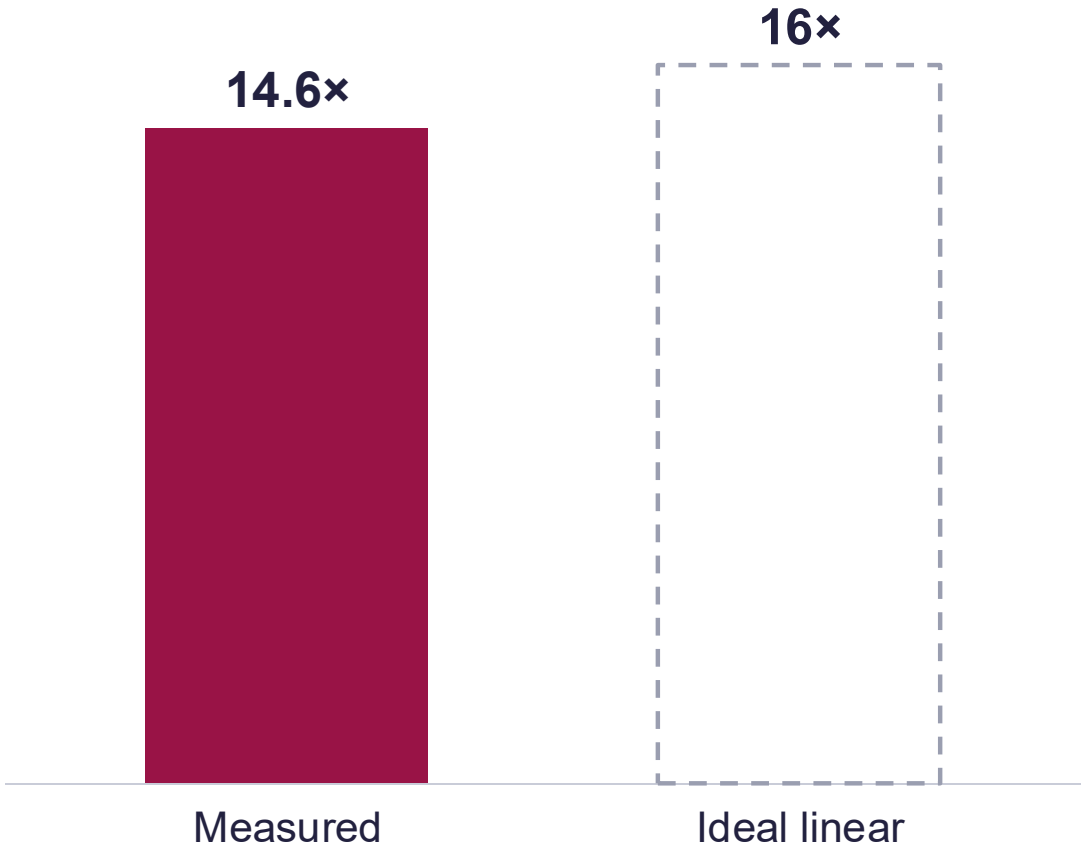
Observability Day NORTH AMERICA

14.6×

speedup from 1 to 16 cores (ideal: 16×)

1.91M

logs per second on 16 cores, OTLP input



Separate scaling experiment on a 64-core, 2-socket Xeon 8358 host: engine OTLP path, 1-core baseline ≈131K logs/s. Phase 2, June 2026.

A3. Phase 2 experiment environments

Observability Day NORTH AMERICA

Experiment	Host and placement	Workload
Transformation and overload	Xeon Platinum 8581C, 16 cores, 118 GiB, Debian 12. One pinned core for the pipeline; generator and backend on the others.	≈300-byte logs, attribute rename, rule count and batch size varied
Saturation and scaling	Xeon 8358, 64 cores, 2 sockets, 1,024 GiB. Separate generator and backend capacity.	1 KB logs; OTLP 1 to 16 cores, OTAP 1 to 8 cores

Reproduce from the recorded commit and configuration: protocol, batch size, compression, rule count, offered and received rates, core pinning.

Benchmark definitions: docs/benchmarks.md. Nightly results: open-telemetry.github.io/otel-arrow/benchmarks/nightly

A4. A simple CPU model for one batch

Observability Day NORTH AMERICA

Object pipeline (OTLP)

C ≈ decode + allocate + Σ process(objects) + encode + runtime

Tax paid on every hop, whatever the processing

Columnar pipeline (OTAP)

C ≈ Arrow IPC + buffers + Σ process(columns) + runtime

Lower entry fee and cheaper batch-wide operations

CPU work per batch with matched semantics, not a latency model. Count each unit of work once. A DFE fed with OTLP adds decode, convert and encode terms.