

# Querying Prometheus with Flux (#fluxlang)

Paul Dix  
@pauldix  
[paul@influxdata.com](mailto:paul@influxdata.com)



**flux**



- Data-scripting language
- Functional
- MIT Licensed
- Language & Runtime/Engine

**Prometheus users: so what?**

# High availability?

# **Sharded Data?**

# Federation?

A screenshot of a GitHub repository page. The top navigation bar is dark with white text, featuring the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. To the right are icons for notifications (blue dot), a plus sign, and a dropdown menu. Below the header, the repository name "prometheus / prometheus" is displayed in blue. To the right are buttons for "Watch" (with 894), "Star" (with 17,928), and "Fork" (with 2). A horizontal navigation bar below the repository name includes links for Code, Issues (220, highlighted in orange), Pull requests (64), Projects (0), Wiki, and Insights. The main content area is currently empty.

---

## Labels

**component/promql**

**kind/enhancement**

# Sub query support for range selection. #1227

 Open

fabxc opened this issue on Nov 18, 2015 · 17 comments



fabxc commented on Nov 18, 2015

Member



Currently one can only do range selection on vector selectors and has to use recording rules to range-select expressions.



7



Prometheus Users ›

## Nested range vectors for counters

8 posts by 4 authors 



roman...@percona.com

11/23/16



Hi guys,

Is it possible somehow to do something like this?

```
min_over_time(rate(mysql_global_status_questions{instance="db1"}[2s])[60s])
```

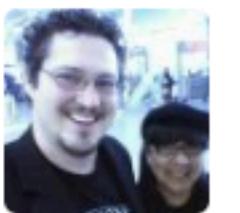
# subqueries



brian-brazil commented on Apr 6, 2017

Member + 😊

This is the issue for it. I wouldn't be too hopeful of it ever getting implemented.



virusdave commented on Apr 6, 2017

+ 😊

OK. Is that due to a philosophical issue, or a eng-resource-scarcity problem?



brian-brazil commented on Apr 6, 2017

Member + 😊

It's a semantic issue, and an operational one. It's unclear how to do this correctly, and even then its use would have to be greatly limited for safety.



brian-brazil added **priority/Pmaybe** **not-as-easy-as-it-looks** **component/promql** labels on Jul 14, 2017

~~subqueries~~  
recording rules

**Ad hoc exploration**

# PromQL: Covariance and correlation functions #2615

ⓘ Open

mattbostock opened this issue on Apr 13, 2017 · 4 comments



mattbostock commented on Apr 13, 2017

Contributor



Are there any plans to support covariance and correlation functions in PromQL?

It could be useful for post-mortem analysis to find correlations that could explain the impact or possible root causes of an incident.



1



brian-brazil commented on Apr 13, 2017

Member



In my opinion that's getting into heavy statistical analysis, and a dedicated tool such as R would be a more appropriate way to manage that.

**Focus is Strength**

# **Saying No is an Asset**



# PromCon 2017

The Prometheus conference – August 17 - 18 in Munich

[OVERVIEW](#)

[SCHEDULE](#)

[CODE OF CONDUCT](#)

Talk abstract

Analyze Prometheus Metrics like a Data Scientist

Speaker: [Georg Öttl](#)

**Liberate the silo!**

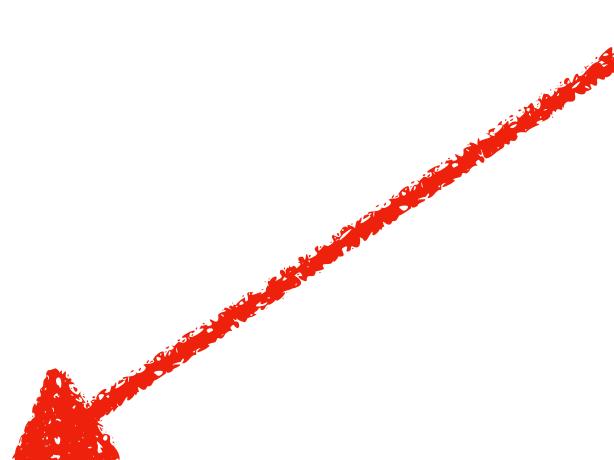


**flux**

# **Language Elements**

```
// get all data from the telegraf db
from(db:"telegraf")
    // filter that by the last hour
    |> range(start:-1h)
    // filter further by series with a specific measurement and field
    |> filter(fn: (r) => r._measurement == "cpu" and r._field == "usage_system")
```

## Comments



```
// get all data from the telegraf db
from(db:"telegraf")
    // filter that by the last hour
    |> range(start:-1h)
    // filter further by series with a specific measurement and field
    |> filter(fn: (r) => r._measurement == "cpu" and r._field == "usage_system")
```

## Functions

```
// get all data from the telegraf db
from(db:"telegraf")
  // filter that by the last hour
  |> range(start:-1h)
  // filter further by series with a specific measurement and field
  |> filter(fn: (r) => r._measurement == "cpu" and r._field == "usage_system")
```

```
// get all data from the telegraf db
from(db:"telegraf")
    // filter that by the last hour
    |> range(start:-1h)
    // filter further by series with a specific measurement and field
    |> filter(fn: r => r._measurement == "cpu" and r._field == "usage_system")
```



## Pipe forward operator

## Named Arguments

```
// get all data from the telegraf db
from(db:"telegraf")
    // filter that by the last hour
    |> range(start:-1h)
    // filter further by series with a specific measurement and field
    |> filter(fn: (r) => r._measurement == "cpu" and r._field == "usage_system")
```

## String Literal

```
// get all data from the telegraf db
from(db:"telegraf")
    // filter that by the last hour
    |> range(start:-1h)
    // filter further by series with a specific measurement and field
    |> filter(fn: (r) => r._measurement == "cpu" and r._field == "usage_system")
```

```
// get all data from the telegraf db
from(db:"telegraf")
    // filter that by the last hour
    |> range(start:-1h) ← Duration Literal (relative time)
    // filter further by series with a specific measurement and field
    |> filter(fn: (r) => r._measurement == "cpu" and r._field == "usage_system")
```

```
// get all data from the telegraf db
from(db:"telegraf")
// filter that by the last hour
|> range(start:"2018-08-09T14:00:00Z") ← Time Literal
// filter further by series with a specific measurement and field
|> filter(fn: (r) => r._measurement == "cpu" and r._field == "usage_system")
```

```
// get all data from the telegraf db
from(db:"telegraf")
    // filter that by the last hour
    |> range(start:-1h)
    // filter further by series with a specific measurement and field
    |> filter(fn: (r) => r._measurement == "cpu" and r._field == "usage_system")
```



**Anonymous Function**

# Operators

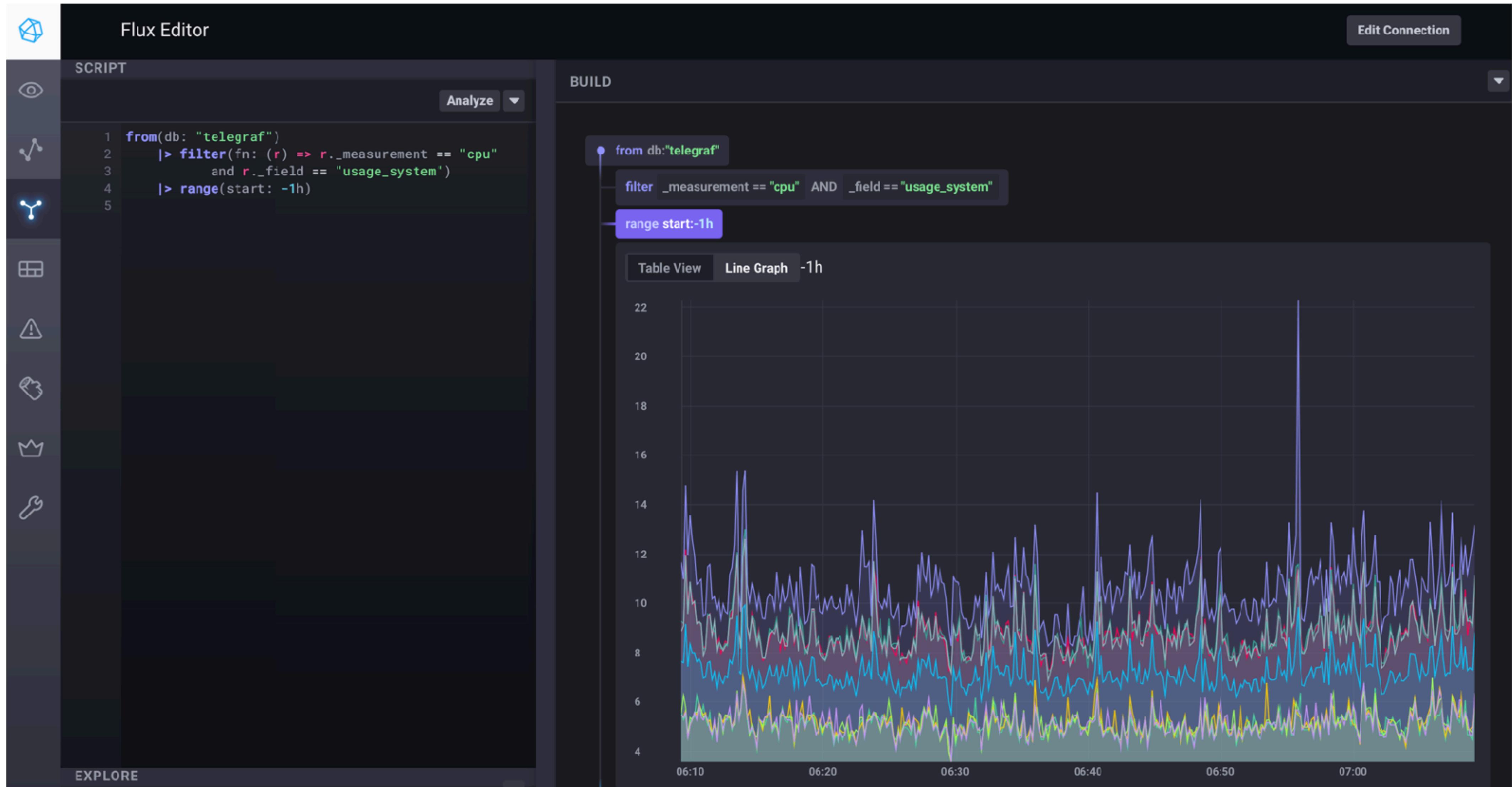
+	==	!=	( )
-	&	!~	[ ]
*	∨	==	{ }
/	≤=	=	,
%	≥=	≤-	.
			∨

# Types

- int
- uint
- float64
- string
- duration
- time
- regex
- array
- object
- function
- namespace
- table
- table stream

**Ways to run Flux - (interpreter,  
fluxd api server, InfluxDB 1.7 & 2.0)**

# Flux builder in Chronograf



# Flux builder in Grafana

The screenshot shows the Grafana Flux builder interface. At the top, there are tabs: Table, General, Metrics (which is selected), Options, Column Styles, and Time range. Below the tabs, there are buttons for Data Source (set to "Flux") and a dropdown menu. A large code editor area contains the following Flux query:

```
from(db: "telegraf")
|> filter(fn: (r) => r["_measurement"] == "cpu" AND
r["_field"] == "usage_idle")
|> range($range)
|> limit(n: 1000)
|> ma
```

To the left of the code editor, there is a panel labeled "A" containing the query. To the right, there is a panel labeled "B" with a button "Add Query". A tooltip for the "map" function is displayed over the "Functions" section:

map(fn: (r) => r)  
Applies a function to each record of the input tables.

Below the functions, there is a "FORMAT" section with the "max()" function listed.

On the far right, there is a "Result records" section.

# Flux is about:



# Time series in Prometheus

```

[~ $ flux -e 'fromProm(query:`node_cpu_seconds_total{cpu="0",mode="idle"}`) |> range(start:-1m)'
running: fromProm(query:`node_cpu_seconds_total{cpu="0",mode="idle"}`) |> range(start:-1m) |> yield(name:"result")
Result: result
Table: keys: [__name__, cpu, host, instance, job, mode, _start, _stop]
  __name__:string      cpu:string      host:string      instance:string      job:string      mode:string      _start:time      _stop:time      _value:float      _time:time
----- ----- ----- ----- ----- ----- ----- ----- -----
node_cpu_seconds_total      0      paul-laptop      localhost:9100      node_exporter      idle 2018-08-08T13:28:22.280519140Z 2018-08-08T13:29:22.280519140Z      32279.76 2018-08-08T13:28:22.000000000Z
node_cpu_seconds_total      0      paul-laptop      localhost:9100      node_exporter      idle 2018-08-08T13:28:22.280519140Z 2018-08-08T13:29:22.280519140Z      32287.56 2018-08-08T13:28:32.000000000Z
node_cpu_seconds_total      0      paul-laptop      localhost:9100      node_exporter      idle 2018-08-08T13:28:22.280519140Z 2018-08-08T13:29:22.280519140Z      32295.81 2018-08-08T13:28:42.000000000Z
node_cpu_seconds_total      0      paul-laptop      localhost:9100      node_exporter      idle 2018-08-08T13:28:22.280519140Z 2018-08-08T13:29:22.280519140Z      32304.05 2018-08-08T13:28:52.000000000Z
node_cpu_seconds_total      0      paul-laptop      localhost:9100      node_exporter      idle 2018-08-08T13:28:22.280519140Z 2018-08-08T13:29:22.280519140Z      32311.54 2018-08-08T13:29:02.000000000Z
node_cpu_seconds_total      0      paul-laptop      localhost:9100      node_exporter      idle 2018-08-08T13:28:22.280519140Z 2018-08-08T13:29:22.280519140Z      32319.13 2018-08-08T13:29:12.000000000Z
node_cpu_seconds_total      0      paul-laptop      localhost:9100      node_exporter      idle 2018-08-08T13:28:22.280519140Z 2018-08-08T13:29:22.280519140Z      32327.01 2018-08-08T13:29:22.000000000Z
~ $
```



```
// get data from Prometheus on http://localhost:9090
fromProm(query:`node_cpu_seconds_total{cpu="0",mode="idle"}`)
    // filter that by the last minute
    |> range(start:-1m)
```







e	_stop:time	_value:float	_time:time
-	-----	-----	-----
Z	2018-08-08T13:29:22.280519140Z	32279.76	2018-08-08T13:28:000000000Z
Z	2018-08-08T13:29:22.280519140Z	32287.56	2018-08-08T13:28:32.000000000Z
Z	2018-08-08T13:29:22.280519140Z	32295.81	2018-08-08T13:28:42.000000000Z
Z	2018-08-08T13:29:22.280519140Z	32304.05	2018-08-08T13:28:52.000000000Z
Z	2018-08-08T13:29:22.280519140Z	32311.54	2018-08-08T13:29:02.000000000Z
Z	2018-08-08T13:29:22.280519140Z	32319.13	2018-08-08T13:29:12.000000000Z
Z	2018-08-08T13:29:22.280519140Z	32327.01	2018-08-08T13:29:22.000000000Z

```
~ $ flux -e 'fromProm(query:`node_cpu_seconds_total{cpu="0",mode="idle"}`) |> range(start:-1m) |> keep(columns:[ "__name__ ", "cpu", "host", "mode", "_value", "_time" ])'  
running: fromProm(query:`node_cpu_seconds_total{cpu="0",mode="idle"}`) |> range(start:-1m) |> keep(columns:[ "__name__ ", "cpu", "host", "mode", "_value", "_time" ]) |> yield(name:"result")  
Result: result  
Table: keys: [ __name__, cpu, host, mode ]  
  __name__:string      cpu:string      host:string      mode:string      _value:float      _time:time  
-----  
node_cpu_seconds_total      0      paul-laptop      idle      32792.81      2018-08-08T13:39:28.000000000Z  
node_cpu_seconds_total      0      paul-laptop      idle      32800.53      2018-08-08T13:39:38.000000000Z  
node_cpu_seconds_total      0      paul-laptop      idle      32808.12      2018-08-08T13:39:48.000000000Z  
node_cpu_seconds_total      0      paul-laptop      idle      32816.3       2018-08-08T13:39:58.000000000Z  
node_cpu_seconds_total      0      paul-laptop      idle      32824.5       2018-08-08T13:40:08.000000000Z  
node_cpu_seconds_total      0      paul-laptop      idle      32832.61      2018-08-08T13:40:18.000000000Z  
node_cpu_seconds_total      0      paul-laptop      idle      32840.8       2018-08-08T13:40:28.000000000Z  
~ $
```

```
  ,mode="idle"}) |> range(start:-1m) |> keep(columns:["_name", "cpu", "host", "mode", "_value", "_time"])'  
  ,mode="idle"}) |> range(start:-1m) |> keep(columns:["_name", "cpu", "host", "mode", "_value", "_time"]) |> yield(name:"re
```

host:string	mode:string	_value:float	_time:time
paul-laptop	idle	32792.81	2018-08-08T13:39:28.00000000Z
paul-laptop	idle	32800.53	2018-08-08T13:39:38.00000000Z
paul-laptop	idle	32808.12	2018-08-08T13:39:48.00000000Z
paul-laptop	idle	32816.3	2018-08-08T13:39:58.00000000Z
paul-laptop	idle	32824.5	2018-08-08T13:40:08.00000000Z
paul-laptop	idle	32832.61	2018-08-08T13:40:18.00000000Z
paul-laptop	idle	32840.8	2018-08-08T13:40:28.00000000Z

```
~ $ flux -e 'fromProm(query:`node_cpu_seconds_total{cpu="0",mode="idle"}`) |> range(start:-1m) |> keep(columns:[ "__name__", "cpu", "host", "mode", "_value", "_time"])'  
running: fromProm(query:`node_cpu_seconds_total{cpu="0",mode="idle"}`) |> range(start:-1m) |> keep(columns:[ "__name__", "cpu", "host", "mode", "_value", "_time"]) |> yield(name:"result")  
Result: result  
Table: keys: [ __name__, cpu, host, mode]  
  __name__:string      cpu:string      host:string      mode:string      _value:float      _time:time  
-----  
node_cpu_seconds_total      0      paul-laptop      idle      32792.81      2018-08-08T13:39:28.000000000Z  
node_cpu_seconds_total      0      paul-laptop      idle      32800.53      2018-08-08T13:39:38.000000000Z  
node_cpu_seconds_total      0      paul-laptop      idle      32808.12      2018-08-08T13:39:48.000000000Z  
node_cpu_seconds_total      0      paul-laptop      idle      32816.3       2018-08-08T13:39:58.000000000Z  
node_cpu_seconds_total      0      paul-laptop      idle      32824.5       2018-08-08T13:40:08.000000000Z  
node_cpu_seconds_total      0      paul-laptop      idle      32832.61      2018-08-08T13:40:18.000000000Z  
node_cpu_seconds_total      0      paul-laptop      idle      32840.8       2018-08-08T13:40:28.000000000Z  
~ $
```

# **Multiple time series in Prometheus**

```
fromProm(query: `node_cpu_seconds_total{cpu="0",mode=~"idle|user"}`)
|> range(start:-1m)
|> keep(columns: ["name", "cpu", "host", "mode", "_value", "_time"])
```

```
[~ $ flux -e 'fromProm(query:`node_cpu_seconds_total{cpu="0",mode=~"idle|user"}`) |> range(start:-30s) |> keep(columns:[ "__name__", "cpu", "host", "mode", "_value", "_time"])'  
running: fromProm(query:`node_cpu_seconds_total{cpu="0",mode=~"idle|user"}`) |> range(start:-30s) |> keep(columns:[ "__name__", "cpu", "host", "mode", "_value", "_time"]) |> yield(name  
"result")  
Result: result  
Table: keys: [ __name__, cpu, host, mode]  
  __name__:string          cpu:string        host:string        mode:string        _value:float        _time:time  
-----  
node_cpu_seconds_total      0           paul-laptop      idle            33070.42  2018-08-08T13:45:19.00000000Z  
node_cpu_seconds_total      0           paul-laptop      idle            33078.8   2018-08-08T13:45:29.00000000Z  
node_cpu_seconds_total      0           paul-laptop      idle            33087.09  2018-08-08T13:45:39.00000000Z  
node_cpu_seconds_total      0           paul-laptop      idle            33095    2018-08-08T13:45:49.00000000Z  
Table: keys: [ __name__, cpu, host, mode]  
  __name__:string          cpu:string        host:string        mode:string        _value:float        _time:time  
-----  
node_cpu_seconds_total      0           paul-laptop      user            3563.89  2018-08-08T13:45:19.00000000Z  
node_cpu_seconds_total      0           paul-laptop      user            3564.77  2018-08-08T13:45:29.00000000Z  
node_cpu_seconds_total      0           paul-laptop      user            3565.75  2018-08-08T13:45:39.00000000Z  
node_cpu_seconds_total      0           paul-laptop      user            3566.6   2018-08-08T13:45:49.00000000Z  
~ $
```

```
[~ $ flux -e 'fromProm(query:`node_cpu_seconds_total{cpu="0",mode=~"idle|user"}`) |> range(start:-30s) |> keep(columns="result")  
running: fromProm(query:`node_cpu_seconds_total{cpu="0",mode=~"idle|user"}`) |> range(start:-30s) |> keep(columns="result")  
"result")  
Result: result  
Table: keys: [__name__, cpu, host, mode]  
    __name__:string          cpu:string          host:string          mode:string          _v  
-----  
node_cpu_seconds_total          0          paul-laptop          idle  
node_cpu_seconds_total          0          paul-laptop          idle  
node_cpu_seconds_total          0          paul-laptop          idle  
node_cpu_seconds_total          0          paul-laptop          idle  
Table: keys: [__name__, cpu, host, mode]  
    __name__:string          cpu:string          host:string          mode:string          _v  
-----  
node_cpu_seconds_total          0          paul-laptop          user  
node_cpu_seconds_total          0          paul-laptop          user  
node_cpu_seconds_total          0          paul-laptop          user  
node_cpu_seconds_total          0          paul-laptop          user  
~ $
```

```
[~ $ flux -e 'fromProm(query:`node_cpu_seconds_total{cpu="0",mode=~"idle|user"}`) |> range(start:-30s) |> keep(columns="result")  
running: fromProm(query:`node_cpu_seconds_total{cpu="0",mode=~"idle|user"}`) |> range(start:-30s) |> keep(columns="result")  
"result")  
Result: result  
Table: keys: [__name__, cpu, host, mode]  
  __name__:string          cpu:string          host:string          mode:string          _v  
-----  
node_cpu_seconds_total      0                 paul-laptop        idle  
node_cpu_seconds_total      0                 paul-laptop        idle  
node_cpu_seconds_total      0                 paul-laptop        idle  
node_cpu_seconds_total      0                 paul-laptop        idle  
Table: keys: [__name__, cpu, host, mode]  
  __name__:string          cpu:string          host:string          mode:string          _v  
-----  
node_cpu_seconds_total      0                 paul-laptop        user  
node_cpu_seconds_total      0                 paul-laptop        user  
node_cpu_seconds_total      0                 paul-laptop        user  
node_cpu_seconds_total      0                 paul-laptop        user  
~ $
```

```
[~ $ flux -e 'fromProm(query:`node_cpu_seconds_total{cpu="0",mode=~"idle|user"}`) |> range(start:-30s) |> keep(columns="result")  
running: fromProm(query:`node_cpu_seconds_total{cpu="0",mode=~"idle|user"}`) |> range(start:-30s) |> keep(columns="result")  
"result")  
Result: result  
Table: keys: [__name__, cpu, host, mode]  
    __name__:string          cpu:string          host:string          mode:string          _v  
-----  -----  -----  -----  -----  
node_cpu_seconds_total          0          paul-laptop          idle  
node_cpu_seconds_total          0          paul-laptop          idle  
node_cpu_seconds_total          0          paul-laptop          idle  
node_cpu_seconds_total          0          paul-laptop          idle  
Table: keys: [__name__, cpu, host, mode]  
    __name__:string          cpu:string          host:string          mode:string          _v  
-----  -----  -----  -----  -----  
node_cpu_seconds_total          0          paul-laptop          user  
node_cpu_seconds_total          0          paul-laptop          user  
node_cpu_seconds_total          0          paul-laptop          user  
node_cpu_seconds_total          0          paul-laptop          user  
~ $
```

n				
	host:string	mode:string	_value:float	_time:time
0	paul-laptop	idle	33070.42	2018-08-08T13:45:19.000000000Z
0	paul-laptop	idle	33078.8	2018-08-08T13:45:29.000000000Z
0	paul-laptop	idle	33087.09	2018-08-08T13:45:39.000000000Z
0	paul-laptop	idle	33095	2018-08-08T13:45:49.000000000Z
n				
	host:string	mode:string	_value:float	_time:time
0	paul-laptop	user	3563.89	2018-08-08T13:45:19.000000000Z
0	paul-laptop	user	3564.77	2018-08-08T13:45:29.000000000Z
0	paul-laptop	user	3565.75	2018-08-08T13:45:39.000000000Z
0	paul-laptop	user	3566.6	2018-08-08T13:45:49.000000000Z

**Tables are the base unit**

**Not tied to a specific data  
model/schema**

# **Filter function**

```
fromProm()
|> range(start:-1m)
|> filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and
    r.mode == "idle" and
    r.cpu == "0")
|> keep(columns: ["name", "cpu", "host", "mode", "_value", "_time"])
```

```
~ $ flux -e 'fromProm() |> filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and r.cpu == "0" and r.mode == "idle") |> range(start:-30s) |> keep(columns:[ "__name__", "cpu", "host", "mode", "_value", "_time"] )'  
running: fromProm() |> filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and r.cpu == "0" and r.mode == "idle") |> range(start:-30s) |> keep(columns:[ "__name__", "cpu", "host", "mode", "_value", "_time"] ) |> yield(name:"result")  
Result: result  
Prometheus query: {__name__="node_cpu_seconds_total",cpu="0",mode="idle"}  
start time: 2018-08-08 16:33:00.474171332 +0100 BST m=-29.914477462  
end time: 2018-08-08 16:33:30.474171332 +0100 BST m=+0.085522538  
Table: keys: [__name__, cpu, host, mode]  
  __name__:string          cpu:string          host:string          mode:string          _value:float          _time:time  
-----  -----  -----  -----  -----  -----  
node_cpu_seconds_total      0            paul-laptop        idle           36373.26  2018-08-08T15:33:00.00000000Z  
node_cpu_seconds_total      0            paul-laptop        idle           36382.05  2018-08-08T15:33:10.00000000Z  
node_cpu_seconds_total      0            paul-laptop        idle           36390.88  2018-08-08T15:33:20.00000000Z  
node_cpu_seconds_total      0            paul-laptop        idle           36398.69  2018-08-08T15:33:30.00000000Z  
~ $
```

```
[~ $ flux -e 'fromProm() |> filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and r.cpu == "0" and r.mode == "idle") |> range("cpu", "host", "mode", "_value", "_time")'  
running: fromProm() |> filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and r.cpu == "0" and r.mode == "idle") |> range("cpu", "host", "mode", "_value", "_time")) |> yield(name:"result")  
Result: result  
Prometheus query: {__name__="node_cpu_seconds_total",cpu="0",mode="idle"}  
start time: 2018-08-08 16:33:00.474171332 +0100 BST m=-29.914477462  
end time: 2018-08-08 16:33:30.474171332 +0100 BST m=+0.085522538  
Table: keys: [__name__, cpu, host, mode]  
  __name__:string          cpu:string          host:string          mode:string          _value:float  
-----  -----  -----  -----  -----  
node_cpu_seconds_total      0        paul-laptop    idle      36373.26 20  
node_cpu_seconds_total      0        paul-laptop    idle      36382.05 20  
node_cpu_seconds_total      0        paul-laptop    idle      36390.88 20  
node_cpu_seconds_total      0        paul-laptop    idle      36398.69 20  
~ $
```

```
[~ $ flux -e 'fromProm() |> filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and r.cpu == "0" and r.mode == "idle") |> range(me__, "cpu", "host", "mode", "_value", "_time")'  
running: fromProm() |> filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and r.cpu == "0" and r.mode == "idle") |> range  
", "cpu", "host", "mode", "_value", "_time"]) |> yield(name:"result")  
Result: result  
Prometheus query: {__name__="node_cpu_seconds_total",cpu="0",mode="idle"}  
start time: 2018-08-08 16:33:00.474171332 +0100 BST m=-29.914477462  
end time: 2018-08-08 16:33:30.474171332 +0100 BST m=+0.085522538  
Table: keys: [__name__, cpu, host, mode]  
  __name__:string          cpu:string          host:string          mode:string          _value:float  
-----  -----  -----  -----  -----  
node_cpu_seconds_total          0          paul-laptop          idle          36373.26 20  
node_cpu_seconds_total          0          paul-laptop          idle          36382.05 20  
node_cpu_seconds_total          0          paul-laptop          idle          36390.88 20  
node_cpu_seconds_total          0          paul-laptop          idle          36398.69 20  
~ $
```

```
[~ $ flux -e 'fromProm() |> filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and r.cpu == "0" and r.mode == "idle") |> range("2018-08-08T16:33:00Z", "2018-08-08T16:33:30Z") |> yield(name:"result")'
running: fromProm() |> filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and r.cpu == "0" and r.mode == "idle") |> range("2018-08-08T16:33:00Z", "2018-08-08T16:33:30Z") |> yield(name:"result")
Result: result
Prometheus query: { name == "node_cpu_seconds_total",cpu="0",mode="idle"}
start time: 2018-08-08 16:33:00.474171332 +0100 BST m=-29.914477462
end time: 2018-08-08 16:33:30.474171332 +0100 BST m=+0.085522538
Table: keys: [__name__, cpu, host, mode]
  __name__:string          cpu:string          host:string          mode:string          _value:float
-----  -----  -----  -----  -----
node_cpu_seconds_total      0            paul-laptop        idle           36373.26 20
node_cpu_seconds_total      0            paul-laptop        idle           36382.05 20
node_cpu_seconds_total      0            paul-laptop        idle           36390.88 20
node_cpu_seconds_total      0            paul-laptop        idle           36398.69 20
~ $
```

```
fromProm()
|> range(start:-1m)
|> filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and
    r.mode in ["idle", "user"] and
    r.cpu == "0")
|> keep(columns: ["name", "cpu", "host", "mode", "_value", "_time"])
```

```
[~ $ flux -e 'fromProm() |> filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and r.cpu == "0" and r.mode in ["idle", "user"]) |> range(start:-30s) |> keep(co
mns:[ "__name__", "cpu", "host", "mode", "_value", "_time"])
running: fromProm() |> filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and r.cpu == "0" and r.mode in ["idle", "user"]) |> range(start:-30s) |> keep(co
[ "__name__", "cpu", "host", "mode", "_value", "_time"]) |> yield(name:"result")
Result: result
Prometheus query: {__name__="node_cpu_seconds_total",cpu="0",mode=~"idle|user"}
start time: 2018-08-08 17:12:02.91391547 +0100 BST m=-29.912049977
end time: 2018-08-08 17:12:32.91391547 +0100 BST m=+0.087950023
Table: keys: [__name__, cpu, host, mode]
  __name__:string          cpu:string          host:string          mode:string          _value:float          _time:time
-----  -----  -----  -----  -----  -----
node_cpu_seconds_total      0           paul-laptop        idle            36884.49  2018-08-08T16:12:02.000000000Z
node_cpu_seconds_total      0           paul-laptop        idle            36892.48  2018-08-08T16:12:12.000000000Z
node_cpu_seconds_total      0           paul-laptop        idle            36900.5   2018-08-08T16:12:22.000000000Z
node_cpu_seconds_total      0           paul-laptop        idle            36908.79  2018-08-08T16:12:32.000000000Z
Table: keys: [__name__, cpu, host, mode]
  __name__:string          cpu:string          host:string          mode:string          _value:float          _time:time
-----  -----  -----  -----  -----  -----
node_cpu_seconds_total      0           paul-laptop        user            4094.37  2018-08-08T16:12:02.000000000Z
node_cpu_seconds_total      0           paul-laptop        user            4095.34  2018-08-08T16:12:12.000000000Z
node_cpu_seconds_total      0           paul-laptop        user            4096.26  2018-08-08T16:12:22.000000000Z
node_cpu_seconds_total      0           paul-laptop        user            4097.16  2018-08-08T16:12:32.000000000Z
~ $
```

```

|> filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and r.cpu == "0" and r.mode in ["idle", "user"])
host, "mode", "_value", "_time")
filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and r.cpu == "0" and r.mode in ["idle", "user"]) |> ran
", "mode", "_value", "_time"]) |> yield(name:"result")

e__="node_cpu_seconds_total",cpu="0",mode=~"idle|user"}
7:12:02.91391547 +0100 BST m=-29.912049977
12:32.91391547 +0100 BST m=+0.087950023
[cpu, host, mode]
  cpu:string          host:string          mode:string          _value:float
-----  -----  -----  -----
  0           paul-laptop        idle          36884.49  2018-08-08
  0           paul-laptop        idle          36892.48  2018-08-08
  0           paul-laptop        idle          36900.5   2018-08-08
  0           paul-laptop        idle          36908.79  2018-08-08
[cpu, host, mode]
  cpu:string          host:string          mode:string          _value:float
-----  -----  -----  -----
  0           paul-laptop        user          4094.37  2018-08-08
  0           paul-laptop        user          4095.34  2018-08-08
  0           paul-laptop        user          4096.26  2018-08-08
  0           paul-laptop        user          4097.16  2018-08-08

```

```
[~ $ flux -e 'fromProm() |> filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and r.cpu == "0" and r.mode in mns:[ "__name__", "cpu", "host", "mode", "_value", "_time"])'  
running: fromProm() |> filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and r.cpu == "0" and r.mode in  
[ "__name__", "cpu", "host", "mode", "_value", "_time"]) |> yield(name:"result")  
Result: result  
Prometheus query: {__name__="node_cpu_seconds_total",cpu="0",mode=~"idle|user"}  
start time: 2018-08-08 17:12:02.91391547 +0100 BST m=-29.91204997  
end time: 2018-08-08 17:12:32.91391547 +0100 BST m=+0.087950023  
Table: keys: [__name__, cpu, host, mode]  
  __name__:string          cpu:string          host:string          mode:string  
-----  -----  -----  -----  
node_cpu_seconds_total      0            paul-laptop        idle  
node_cpu_seconds_total      0            paul-laptop        idle  
node_cpu_seconds_total      0            paul-laptop        idle  
node_cpu_seconds_total      0            paul-laptop        idle  
Table: keys: [__name__, cpu, host, mode]  
  __name__:string          cpu:string          host:string          mode:string  
-----  -----  -----  -----  
node_cpu_seconds_total      0            paul-laptop        user  
node_cpu_seconds_total      0            paul-laptop        user  
node_cpu_seconds_total      0            paul-laptop        user  
node_cpu_seconds_total      0            paul-laptop        user  
~ $
```

# Aggregate functions

```
fromProm()
|> range(start:-30s)
|> filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and
    r.mode == "idle" and
    r.cpu =~ /0|1/)
|> count()
|> keep(columns: ["name", "cpu", "host", "mode", "_value", "_time"])
```

```
[~ $ flux -e 'fromProm(query:`node_cpu_seconds_total{cpu=~"0|1",mode="idle"}`) |> range(start:-30s) |> count() |> keep(columns:[ "__name__ ", "cpu", "mode", "_value", "_time"] )'  
Result: result  
Prometheus query: node_cpu_seconds_total{cpu=~"0|1",mode="idle"}  
start time: 2018-08-08 17:41:11.696067074 +0100 BST m=-29.912485372  
end time: 2018-08-08 17:41:41.696067074 +0100 BST m=+0.087514628  
Table: keys: [ __name__ , cpu, mode]  
  __name__:string          cpu:string          mode:string          _time:time          _value:int  
-----  -----  -----  -----  -----  
node_cpu_seconds_total      0           idle   2018-08-08T16:41:11.696067074Z      4  
Table: keys: [ __name__ , cpu, mode]  
  __name__:string          cpu:string          mode:string          _time:time          _value:int  
-----  -----  -----  -----  -----  
node_cpu_seconds_total      1           idle   2018-08-08T16:41:41.696067074Z      4  
~ $
```

```
[~ $ flux -e 'fromProm(query:`node_cpu_seconds_total{cpu=~"0|1",mode="idle"}`) |> range(start:-30s) |> count() |> keep(columns:[ "__name__ ", "cpu", "mode", "_value", "_time"] )'  
Result: result  
Prometheus query: node_cpu_seconds_total{cpu=~"0|1",mode="idle"}  
start time: 2018-08-08 17:41:11.696067074 +0100 BST m=-29.912485372  
end time: 2018-08-08 17:41:41.696067074 +0100 BST m=+0.087514628  
Table: keys: [ __name__, cpu, mode]  
  __name__:string          cpu:string          mode:string          _time:time          _value:int  
-----  -----  -----  -----  -----  
node_cpu_seconds_total      0           idle   2018-08-08T16:41:11.696067074Z      4  
Table: keys: [ __name__, cpu, mode]  
  __name__:string          cpu:string          mode:string          _time:time          _value:int  
-----  -----  -----  -----  -----  
node_cpu_seconds_total      1           idle   2018-08-08T16:41:41.696067074Z      4  
~ $
```

```
[~ $ flux -e 'fromProm(query:`node_cpu_seconds_total{cpu=~"0|1",mode="idle"}`) |> range(start:-30s) |> count() |> keep(columns:[ "__name__ ", "cpu", "mode", "_value", "_time"] )'  
Result: result  
Prometheus query: node_cpu_seconds_total{cpu=~"0|1",mode="idle"}  
start time: 2018-08-08 17:41:11.696067074 +0100 BST m=-29.912485372  
end time: 2018-08-08 17:41:41.696067074 +0100 BST m=+0.087514628  
Table: keys: [ __name__ , cpu, mode]  
    __name__:string          cpu:string          mode:string          _time:time          _value:int  
-----  -----  -----  -----  -----  
node_cpu_seconds_total      0           idle   2018-08-08T16:41:41.696067074Z      4  
Table: keys: [ __name__ , cpu, mode]  
    __name__:string          cpu:string          mode:string          _time:time          _value:int  
-----  -----  -----  -----  -----  
node_cpu_seconds_total      1           idle   2018-08-08T16:41:41.696067074Z      4  
~ $
```

```
fromProm(query:`node_cpu_seconds_total{cpu=~"0|1",mode="idle"}`) |> range(start:-30s) |> count() |> keep(columns:[ "__name__", "cpu", "mode", "_value", "_stop"] |> timeformat("2006-01-02T15:04:05Z")) |> groupBy(["cpu", "mode"], sort: true) |> eval(_value: int, _stop: time, _time: time) |> print
```

y: node\_cpu\_seconds\_total{cpu=~"0|1",mode="idle"}  
18-08-08 17:45:37.308462377 +0100 BST m=-29.913667178  
-08-08 17:46:07.308462377 +0100 BST m=+0.086332822

cpu	mode	_stop	_time	_value
0	idle	2018-08-08T16:46:07.308462377Z	2018-08-08T16:46:07.308462377Z	4
1	idle	2018-08-08T16:46:07.308462377Z	2018-08-08T16:46:07.308462377Z	4

```
|~ $ flux -e 'fromProm(query:`node_cpu_seconds_total{cpu=~"0|1",mode="idle"}` |> range(start:-30s) |> count() |> keep(columns:[ "__name__", "cpu", "mode", "time", "_stop"])'
Result: result
Prometheus query: node_cpu_seconds_total{cpu=~"0|1",mode="idle"}
start time: 2018-08-08 17:45:37.308462377 +0100 BST m=-29.913667178
end time: 2018-08-08 17:46:07.308462377 +0100 BST m=+0.086332822
Table: keys: [ __name__, cpu, mode, _stop]
  __name__:string          cpu:string        mode:string      _stop:time      _time:time
-----  -----  -----  -----
node_cpu_seconds_total      0            idle  2018-08-08T16:46:07.308462377Z 2018-08-08T16:46:07.308462377Z
Table: keys: [ __name__, cpu, mode, _stop]
  __name__:string          cpu:string        mode:string      _stop:time      _time:time
-----  -----  -----  -----
node_cpu_seconds_total      1            idle  2018-08-08T16:46:07.308462377Z 2018-08-08T16:46:07.308462377Z
~ $
```

```
[~ $ flux -e 'fromProm() |> filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and r.cpu == "0" and r.mode in ["idle", "user"]) |> range(start:-30s) |> keep(co
mns:[ "__name__", "cpu", "host", "mode", "_value", "_time"])
running: fromProm() |> filter(fn: (r) => r.__name__ == "node_cpu_seconds_total" and r.cpu == "0" and r.mode in ["idle", "user"]) |> range(start:-30s) |> keep(co
[ "__name__", "cpu", "host", "mode", "_value", "_time"]) |> yield(name:"result")
Result: result
Prometheus query: { __name__="node_cpu_seconds_total",cpu="0",mode=~"idle|user"}
start time: 2018-08-08 17:12:02.91391547 +0100 BST m=-29.912049977
end time: 2018-08-08 17:12:32.91391547 +0100 BST m=+0.087950023
Table: keys: [__name__, cpu, host, mode]
  __name__:string          cpu:string          host:string          mode:string          _value:float          _time:time
-----  -----  -----  -----  -----  -----
node_cpu_seconds_total      0            paul-laptop        idle           36884.49  2018-08-08T16:12:02.000000000Z
node_cpu_seconds_total      0            paul-laptop        idle           36892.48  2018-08-08T16:12:12.000000000Z
node_cpu_seconds_total      0            paul-laptop        idle           36900.5   2018-08-08T16:12:22.000000000Z
node_cpu_seconds_total      0            paul-laptop        idle           36908.79  2018-08-08T16:12:32.000000000Z
Table: keys: [__name__, cpu, host, mode]
  __name__:string          cpu:string          host:string          mode:string          _value:float          _time:time
-----  -----  -----  -----  -----  -----
node_cpu_seconds_total      0            paul-laptop        user           4094.37  2018-08-08T16:12:02.000000000Z
node_cpu_seconds_total      0            paul-laptop        user           4095.34  2018-08-08T16:12:12.000000000Z
node_cpu_seconds_total      0            paul-laptop        user           4096.26  2018-08-08T16:12:22.000000000Z
node_cpu_seconds_total      0            paul-laptop        user           4097.16  2018-08-08T16:12:32.000000000Z
~ $
```

`_start` and `_stop` are about  
windows of data

```
fromProm(query: `node_cpu_seconds_total{cpu="0",mode="idle"}`
|> range(start: -1m)
```

```
[~ $ flux -e 'fromProm(query:`node_cpu_seconds_total{cpu="0",mode="idle"}`) |> range(start:-1m) |> drop(columns: ["host", "instance", "job"])'  
running: fromProm(query:`node_cpu_seconds_total{cpu="0",mode="idle"}`) |> range(start:-1m) |> drop(columns: ["host", "instance", "job"]) |> yield(name:"result")  
Result: result  
Prometheus query: node cpu seconds total{cpu="0".mode="idle"}  
start time: 2018-08-09 09:33:00.086180849 +0200 CEST m=-59.909242575  
end time: 2018-08-09 09:34:00.086180849 +0200 CEST m=+0.090757425  
table: Keys: __name__, cpu, mode, _start, _stop  
      __name__:string          cpu:string        mode:string      _start:time      _stop:time      _value:float      _time:time  
-----  
node_cpu_seconds_total      0           idle    2018-08-09T07:33:00.086180849Z 2018-08-09T07:34:00.086180849Z 44221.17 2018-08-09T07:33:00.000000000Z  
node_cpu_seconds_total      0           idle    2018-08-09T07:33:00.086180849Z 2018-08-09T07:34:00.086180849Z 44230.08 2018-08-09T07:33:10.000000000Z  
node_cpu_seconds_total      0           idle    2018-08-09T07:33:00.086180849Z 2018-08-09T07:34:00.086180849Z 44238.91 2018-08-09T07:33:20.000000000Z  
node_cpu_seconds_total      0           idle    2018-08-09T07:33:00.086180849Z 2018-08-09T07:34:00.086180849Z 44247.66 2018-08-09T07:33:30.000000000Z  
node_cpu_seconds_total      0           idle    2018-08-09T07:33:00.086180849Z 2018-08-09T07:34:00.086180849Z 44256.58 2018-08-09T07:33:40.000000000Z  
node_cpu_seconds_total      0           idle    2018-08-09T07:33:00.086180849Z 2018-08-09T07:34:00.086180849Z 44265.54 2018-08-09T07:33:50.000000000Z  
node_cpu_seconds_total      0           idle    2018-08-09T07:33:00.086180849Z 2018-08-09T07:34:00.086180849Z 44273.67 2018-08-09T07:34:00.000000000Z  
~ $
```

```
fromProm(query: `node_cpu_seconds_total{cpu="0",mode="idle"}`
|> range(start: -1m)
|> window(every: 20s)
```

```
[~ $ flux -e 'fromProm(query:`node_cpu_seconds_total{cpu="0",mode="idle"}`) |> range(start:-1m) |> window(every:20s) |> drop(columns: ["host", "instance", "job"])' running: fromProm(query:`node_cpu_seconds_total{cpu="0",mode="idle"}`) |> range(start:-1m) |> window(every:20s) |> drop(columns: ["host", "instance", "job"]) |> yield(name:"result") Result: result Prometheus query: node_cpu_seconds_total{cpu="0",mode="idle"} start time: 2018-08-09 09:35:29.549516445 +0200 CEST m=-59.913987359 end time: 2018-08-09 09:36:29.549516445 +0200 CEST m=+0.086812641 Table: keys: [__name__, cpu, mode, _start, _stop] __name__:string cpu:string mode:string _start:time _stop:time _value:float _time:time node_cpu_seconds_total 0 idle 2018-08-09T07:35:29.510723527Z 2018-08-09T07:35:40.000000000Z 44353.66 2018-08-09T07:35:39.000000000Z Table: keys: [__name__, cpu, mode, _start, _stop] __name__:string cpu:string mode:string _start:time _stop:time _value:float _time:time node_cpu_seconds_total 0 idle 2018-08-09T07:35:40.000000000Z 2018-08-09T07:36:00.000000000Z 44362.09 2018-08-09T07:35:49.000000000Z node_cpu_seconds_total 0 idle 2018-08-09T07:35:40.000000000Z 2018-08-09T07:36:00.000000000Z 44370.08 2018-08-09T07:35:59.000000000Z Table: keys: [__name__, cpu, mode, _start, _stop] __name__:string cpu:string mode:string _start:time _stop:time _value:float _time:time node_cpu_seconds_total 0 idle 2018-08-09T07:36:00.000000000Z 2018-08-09T07:36:20.000000000Z 44378.02 2018-08-09T07:36:09.000000000Z node_cpu_seconds_total 0 idle 2018-08-09T07:36:00.000000000Z 2018-08-09T07:36:20.000000000Z 44385.8 2018-08-09T07:36:19.000000000Z Table: keys: [__name__, cpu, mode, _start, _stop] __name__:string cpu:string mode:string _start:time _stop:time _value:float _time:time node_cpu_seconds_total 0 idle 2018-08-09T07:36:20.000000000Z 2018-08-09T07:36:29.510723527Z 44394.59 2018-08-09T07:36:29.000000000Z ~ $
```

```
fromProm(query: `node_cpu_seconds_total{cpu="0",mode="idle"}`
    |> range(start: -1m)
    |> window(every: 20s)
    |> min()
```

```
[~ $ flux -e 'fromProm(query:`node_cpu_seconds_total{cpu="0",mode="idle"}`) |> range(start:-1m) |> window(every:20s) |> min() |> drop(columns: ["host", "instance", "job"])' running: fromProm(query:`node_cpu_seconds_total{cpu="0",mode="idle"}`) |> range(start:-1m) |> window(every:20s) |> min() |> drop(columns: ["host", "instance", "job"]) |> yield(name:"result") Result: result
Prometheus query: node_cpu_seconds_total{cpu="0",mode="idle"}
start time: 2018-08-09 09:37:22.091479413 +0200 CEST m=-59.913724359
end time: 2018-08-09 09:38:22.091479413 +0200 CEST m=+0.086275641
Table: keys: [__name__, cpu, mode, __start, __stop]
  __name__:string      cpu:string      mode:string      __start:time      __stop:time      __value:float      __time:time
-----  -----  -----  -----  -----  -----  -----
node_cpu_seconds_total      0      idle  2018-08-09T07:37:22.054007076Z 2018-08-09T07:37:40.000000000Z      44445.72  2018-08-09T07:37:32.000000000Z
Table: keys: [__name__, cpu, mode, __start, __stop]
  __name__:string      cpu:string      mode:string      __start:time      __stop:time      __value:float      __time:time
-----  -----  -----  -----  -----  -----  -----
node_cpu_seconds_total      0      idle  2018-08-09T07:37:40.000000000Z 2018-08-09T07:38:00.000000000Z      44453.72  2018-08-09T07:37:42.000000000Z
Table: keys: [__name__, cpu, mode, __start, __stop]
  __name__:string      cpu:string      mode:string      __start:time      __stop:time      __value:float      __time:time
-----  -----  -----  -----  -----  -----  -----
node_cpu_seconds_total      0      idle  2018-08-09T07:38:00.000000000Z 2018-08-09T07:38:20.000000000Z      44470.07  2018-08-09T07:38:02.000000000Z
Table: keys: [__name__, cpu, mode, __start, __stop]
  __name__:string      cpu:string      mode:string      __start:time      __stop:time      __value:float      __time:time
-----  -----  -----  -----  -----  -----  -----
node_cpu_seconds_total      0      idle  2018-08-09T07:38:20.000000000Z 2018-08-09T07:38:22.054007076Z      44486.76  2018-08-09T07:38:22.000000000Z
~ $
```

```
fromProm(query: `node_cpu_seconds_total{cpu="0",mode="idle"}`
    |> range(start: -1m)
    |> window(every: 20s)
    |> min()
    |> window(every:inf)
```

```
[~ $ flux -e 'fromProm(query:`node_cpu_seconds_total{cpu="0",mode="idle"}`) |> range(start:-1m) |> window(every:20s) |> min() |> window(every:inf) |> drop(columns: ["host", "instance", "job"])' running: fromProm(query:`node_cpu_seconds_total{cpu="0",mode="idle"}`) |> range(start:-1m) |> window(every:20s) |> min() |> window(every:inf) |> drop(columns: ["host", "instance", "job"]) |> yield(gt)
Result: result
Prometheus query: node_cpu_seconds_total{cpu="0",mode="idle"}
start time: 2018-08-09 09:41:28.843629594 +0200 CEST m=-59.913502623
end time: 2018-08-09 09:42:28.843629594 +0200 CEST m=+0.086497377
Table: keys: [__name__, cpu, mode, _start, _stop]
  __name__:string      cpu:string      mode:string      _start:time      _stop:time      _value:float      _time:time
----- ----- ----- ----- ----- -----
node_cpu_seconds_total      0      idle 1677-09-21T00:12:43.145224192Z 2262-04-11T23:47:16.854775807Z      44649.96 2018-08-09T07:41:38.000000000Z
node_cpu_seconds_total      0      idle 1677-09-21T00:12:43.145224192Z 2262-04-11T23:47:16.854775807Z      44658.44 2018-08-09T07:41:48.000000000Z
node_cpu_seconds_total      0      idle 1677-09-21T00:12:43.145224192Z 2262-04-11T23:47:16.854775807Z      44676.1 2018-08-09T07:42:08.000000000Z
node_cpu_seconds_total      0      idle 1677-09-21T00:12:43.145224192Z 2262-04-11T23:47:16.854775807Z      44693.2 2018-08-09T07:42:28.000000000Z
~ $
```

**Window converts N tables to M  
tables based on time boundaries**

**Group converts N tables to M  
tables based on values**

```
fromProm(query: `node_cpu_seconds_total{cpu=~“0|1”,mode=“idle”}`)
|> range(start: -1m)
```

```

[~ $ flux -e 'fromProm(query: `node_cpu_seconds_total{cpu=~"0|1",mode="idle"}`) |> range(start:-1m) |> drop(columns: ["host", "instance", "job"])'
running: fromProm(query: `node_cpu_seconds_total{cpu=~"0|1",mode="idle"}`) |> range(start:-1m) |> drop(columns: ["host", "instance", "job"]) |> yield(name:"result")
Result: result
Prometheus query: node_cpu_seconds_total{cpu=~"0|1".mode="idle"}
start time: 2018-08-09 10:04:58.039636172 +0200 CEST m=-59.913509721
end time: 2018-08-09 10:05:58.039636172 +0200 CEST m=+0.086490279
Table: keys: [__name__, cpu, mode, _start, _stop]
  __name__:string          cpu:string        mode:string      _start:time      _stop:time      _value:float      _time:time
-----+
node_cpu_seconds_total      0              idle           2018-08-09T08:04:58.039636172Z 2018-08-09T08:05:58.039636172Z 45846.75 2018-08-09T08:04:58.000000000Z
node_cpu_seconds_total      0              idle           2018-08-09T08:04:58.039636172Z 2018-08-09T08:05:58.039636172Z 45855.66 2018-08-09T08:05:08.000000000Z
node_cpu_seconds_total      0              idle           2018-08-09T08:04:58.039636172Z 2018-08-09T08:05:58.039636172Z 45864.52 2018-08-09T08:05:18.000000000Z
node_cpu_seconds_total      0              idle           2018-08-09T08:04:58.039636172Z 2018-08-09T08:05:58.039636172Z 45872.72 2018-08-09T08:05:28.000000000Z
node_cpu_seconds_total      0              idle           2018-08-09T08:04:58.039636172Z 2018-08-09T08:05:58.039636172Z 45881.64 2018-08-09T08:05:38.000000000Z
node_cpu_seconds_total      0              idle           2018-08-09T08:04:58.039636172Z 2018-08-09T08:05:58.039636172Z 45890.52 2018-08-09T08:05:48.000000000Z
node_cpu_seconds_total      0              idle           2018-08-09T08:04:58.039636172Z 2018-08-09T08:05:58.039636172Z 45899.23 2018-08-09T08:05:58.000000000Z
Table: keys: [__name__, cpu, mode, _start, _stop]
  __name__:string          cpu:string        mode:string      _start:time      _stop:time      _value:float      _time:time
-----+
node_cpu_seconds_total      1              idle           2018-08-09T08:04:58.039636172Z 2018-08-09T08:05:58.039636172Z 54380.9 2018-08-09T08:04:58.000000000Z
node_cpu_seconds_total      1              idle           2018-08-09T08:04:58.039636172Z 2018-08-09T08:05:58.039636172Z 54390.85 2018-08-09T08:05:08.000000000Z
node_cpu_seconds_total      1              idle           2018-08-09T08:04:58.039636172Z 2018-08-09T08:05:58.039636172Z 54400.79 2018-08-09T08:05:18.000000000Z
node_cpu_seconds_total      1              idle           2018-08-09T08:04:58.039636172Z 2018-08-09T08:05:58.039636172Z 54410.55 2018-08-09T08:05:28.000000000Z
node_cpu_seconds_total      1              idle           2018-08-09T08:04:58.039636172Z 2018-08-09T08:05:58.039636172Z 54420.49 2018-08-09T08:05:38.000000000Z
node_cpu_seconds_total      1              idle           2018-08-09T08:04:58.039636172Z 2018-08-09T08:05:58.039636172Z 54430.42 2018-08-09T08:05:48.000000000Z
node_cpu_seconds_total      1              idle           2018-08-09T08:04:58.039636172Z 2018-08-09T08:05:58.039636172Z 54440.31 2018-08-09T08:05:58.000000000Z
-
```

```
fromProm(query: `node_cpu_seconds_total{cpu=~“0|1”,mode=“idle”}`)
|> range(start: -1m)
|> group(columns: [“__name__”, “mode”])
```

```

~ $ flux -e 'fromProm(query: `node_cpu_seconds_total{cpu=~"0|1".mode="idle"}`) |> range(start:-1m) |> group(by: ["__name__", "mode"]) |> drop(columns: ["host", "instance", "job"])
running: fromProm(query: `node_cpu_seconds_total{cpu=~"0|1".mode="idle"}`) |> range(start:-1m) |> group(by: ["__name__", "mode"]) |> drop(columns: ["host", "instance", "job"]) |> yield(name:"result")
Result: result
Prometheus query: node_cpu_seconds_total{cpu=~"0|1",mode="idle"}
start time: 2018-08-09 10:09:43.44979915 +0200 CEST m=-59.910056239
end time: 2018-08-09 10:10:43.44979915 +0200 CEST m=+0.089943761
Table: keys: [__name__, mode]
  __name__:string      mode:string      cpu:string      _start:time      _stop:time      _value:float      _time:time
  -----  -----  -----  -----  -----  -----  -----
node_cpu_seconds_total      idle      0  2018-08-09T08:09:43.449799150Z  2018-08-09T08:10:43.449799150Z  46088.64  2018-08-09T08:09:43.000000000Z
node_cpu_seconds_total      idle      0  2018-08-09T08:09:43.449799150Z  2018-08-09T08:10:43.449799150Z  46097.51  2018-08-09T08:09:53.000000000Z
node_cpu_seconds_total      idle      0  2018-08-09T08:09:43.449799150Z  2018-08-09T08:10:43.449799150Z  46106.38  2018-08-09T08:10:03.000000000Z
node_cpu_seconds_total      idle      0  2018-08-09T08:09:43.449799150Z  2018-08-09T08:10:43.449799150Z  46114.75  2018-08-09T08:10:13.000000000Z
node_cpu_seconds_total      idle      0  2018-08-09T08:09:43.449799150Z  2018-08-09T08:10:43.449799150Z  46123.45  2018-08-09T08:10:23.000000000Z
node_cpu_seconds_total      idle      0  2018-08-09T08:09:43.449799150Z  2018-08-09T08:10:43.449799150Z  46132.22  2018-08-09T08:10:33.000000000Z
node_cpu_seconds_total      idle      0  2018-08-09T08:09:43.449799150Z  2018-08-09T08:10:43.449799150Z  46141.11  2018-08-09T08:10:43.000000000Z
node_cpu_seconds_total      idle      1  2018-08-09T08:09:43.449799150Z  2018-08-09T08:10:43.449799150Z  54662.18  2018-08-09T08:09:43.000000000Z
node_cpu_seconds_total      idle      1  2018-08-09T08:09:43.449799150Z  2018-08-09T08:10:43.449799150Z  54672.1   2018-08-09T08:09:53.000000000Z
node_cpu_seconds_total      idle      1  2018-08-09T08:09:43.449799150Z  2018-08-09T08:10:43.449799150Z  54681.96  2018-08-09T08:10:03.000000000Z
node_cpu_seconds_total      idle      1  2018-08-09T08:09:43.449799150Z  2018-08-09T08:10:43.449799150Z  54691.77  2018-08-09T08:10:13.000000000Z
node_cpu_seconds_total      idle      1  2018-08-09T08:09:43.449799150Z  2018-08-09T08:10:43.449799150Z  54701.69  2018-08-09T08:10:23.000000000Z
node_cpu_seconds_total      idle      1  2018-08-09T08:09:43.449799150Z  2018-08-09T08:10:43.449799150Z  54711.61  2018-08-09T08:10:33.000000000Z
node_cpu_seconds_total      idle      1  2018-08-09T08:09:43.449799150Z  2018-08-09T08:10:43.449799150Z  54721.55  2018-08-09T08:10:43.000000000Z
~ $

```

```

[~ $ flux -e 'fromProm(query: `node_cpu_seconds_total{cpu=~"0|1",mode="idle"}`) |> range(start:-1m) |> group(by: ["__name__", "mode"]) |> sort(cols: ["_time"]) |> drop(columns: ["host", "instance", "running: fromProm(query: `node_cpu_seconds_total{cpu=~"0|1",mode="idle"}`) |> range(start:-1m) |> group(by: ["__name__", "mode"]) |> sort(cols: ["_time"]) |> drop(columns: ["host", "instance", "job"] id(name:"result")
Result: result
Prometheus query: node_cpu_seconds_total{cpu=~"0|1",mode="idle"}
start time: 2018-08-09 10:12:13.535369346 +0200 CEST m=-59.910877632
end time: 2018-08-09 10:13:13.535369346 +0200 CEST m=+0.089122368
Table: keys: [__name__, mode]
  __name__:string      mode:string      cpu:string      _start:time      _stop:time      _value:float      _time:time
----- ----- ----- -----
node_cpu_seconds_total      idle      1 2018-08-09T08:12:13.535369346Z 2018-08-09T08:13:13.535369346Z 54810.21 2018-08-09T08:12:13.000000000Z
node_cpu_seconds_total      idle      0 2018-08-09T08:12:13.535369346Z 2018-08-09T08:13:13.535369346Z 46212.13 2018-08-09T08:12:13.000000000Z
node_cpu_seconds_total      idle      1 2018-08-09T08:12:13.535369346Z 2018-08-09T08:13:13.535369346Z 54820.11 2018-08-09T08:12:23.000000000Z
node_cpu_seconds_total      idle      0 2018-08-09T08:12:13.535369346Z 2018-08-09T08:13:13.535369346Z 46220.33 2018-08-09T08:12:23.000000000Z
node_cpu_seconds_total      idle      1 2018-08-09T08:12:13.535369346Z 2018-08-09T08:13:13.535369346Z 54830 2018-08-09T08:12:33.000000000Z
node_cpu_seconds_total      idle      0 2018-08-09T08:12:13.535369346Z 2018-08-09T08:13:13.535369346Z 46228.53 2018-08-09T08:12:33.000000000Z
node_cpu_seconds_total      idle      1 2018-08-09T08:12:13.535369346Z 2018-08-09T08:13:13.535369346Z 54839.89 2018-08-09T08:12:43.000000000Z
node_cpu_seconds_total      idle      0 2018-08-09T08:12:13.535369346Z 2018-08-09T08:13:13.535369346Z 46236.83 2018-08-09T08:12:43.000000000Z
node_cpu_seconds_total      idle      1 2018-08-09T08:12:13.535369346Z 2018-08-09T08:13:13.535369346Z 54849.8 2018-08-09T08:12:53.000000000Z
node_cpu_seconds_total      idle      0 2018-08-09T08:12:13.535369346Z 2018-08-09T08:13:13.535369346Z 46245.05 2018-08-09T08:12:53.000000000Z
node_cpu_seconds_total      idle      1 2018-08-09T08:12:13.535369346Z 2018-08-09T08:13:13.535369346Z 54859.54 2018-08-09T08:13:03.000000000Z
node_cpu_seconds_total      idle      0 2018-08-09T08:12:13.535369346Z 2018-08-09T08:13:13.535369346Z 46251.53 2018-08-09T08:13:03.000000000Z
node_cpu_seconds_total      idle      0 2018-08-09T08:12:13.535369346Z 2018-08-09T08:13:13.535369346Z 46259.34 2018-08-09T08:13:13.000000000Z
node_cpu_seconds_total      idle      1 2018-08-09T08:12:13.535369346Z 2018-08-09T08:13:13.535369346Z 54869.35 2018-08-09T08:13:13.000000000Z

```

# Sub query support for range selection. #1227

 Open

fabxc opened this issue on Nov 18, 2015 · 17 comments



fabxc commented on Nov 18, 2015

Member



Currently one can only do range selection on vector selectors and has to use recording rules to range-select expressions.



7



# Nested range vectors

```
fromProm(host:"http://localhost:9090")
|> filter(fn: (r) => r.__name__ == "node_disk_written_bytes_total")
|> range(start:-1h)
// transform into non-negative derivative values
|> derivative()
// break those out into tables for each 10 minute block of time
|> window(every:10m)
// get the max rate of change in each 10 minute window
|> max()
// and put everything back into a single table
|> window(every:inf)
// and now let's convert to KB
|> map(fn: (r) => r._value / 1024.0)
```

# Multiple Servers

```
dc1 = fromProm(host:"http://prom.dc1.local:9090")
|> filter(fn: (r) => r.__name__ == "node_network_receive_bytes_total")
|> range(start:-1h)
|> insertGroupKey(key: "dc", value: "1")
```

```
dc2 = fromProm(host:"http://prom.dc2.local:9090")
|> filter(fn: (r) => r.__name__ == "node_network_receive_bytes_total")
|> range(start:-1h)
|> insertGroupKey(key: "dc", value: "2")
```

```
dc1 |> union(streams: [dc2])
|> limit(n: 2)
|> derivative()
|> group(columns: ["dc"])
|> sum()
```

# Work with data from many sources

- `from() // influx`
- `fromProm()`
- `fromMySQL()`
- `fromCSV()`
- `fromS3()`
- `...`

# Defining Functions

```
fromProm(query: `node_cpu_seconds_total{cpu="0",mode="idle"}`  
        |> range(start: -1m)  
        |> window(every: 20s)  
        |> min()  
        |> window(every:inf)
```

# Defining Functions

```
windowAgg = (every, fn, <-stream) => {
  return stream |> window(every: every) |> fn() |> window(every:inf)
}
```

```
fromProm(query: `node_cpu_seconds_total{cpu="0",mode="idle"}`)
|> range(start: -1m)
|> windowAgg(every:20s, fn: min)
```

# Packages & Namespaces

```
package "flux-helpers"
```

```
windowAgg = (every, fn, <-stream) => {
    return stream |> window(every: every) |> fn() |> window(every:inf)
}
```

```
// in a new script
import helpers "github.com/pauldix/flux-helpers"
```

```
fromProm(query: `node_cpu_seconds_total{cpu="0",mode="idle"}`)
    |> range(start: -1m)
    |> helpers.windowAgg(every:20s, fn: min)
```

# Project Status

- Everything in this talk is prototype (as of 2018-08-09)
- Proposed Final Language Spec
- Release flux, fluxd, InfluxDB 1.7, InfluxDB 2.0 alpha
- Iterate with community to finalize spec
- Optimizations!
- <https://github.com/influxdata/flux>

# Future work

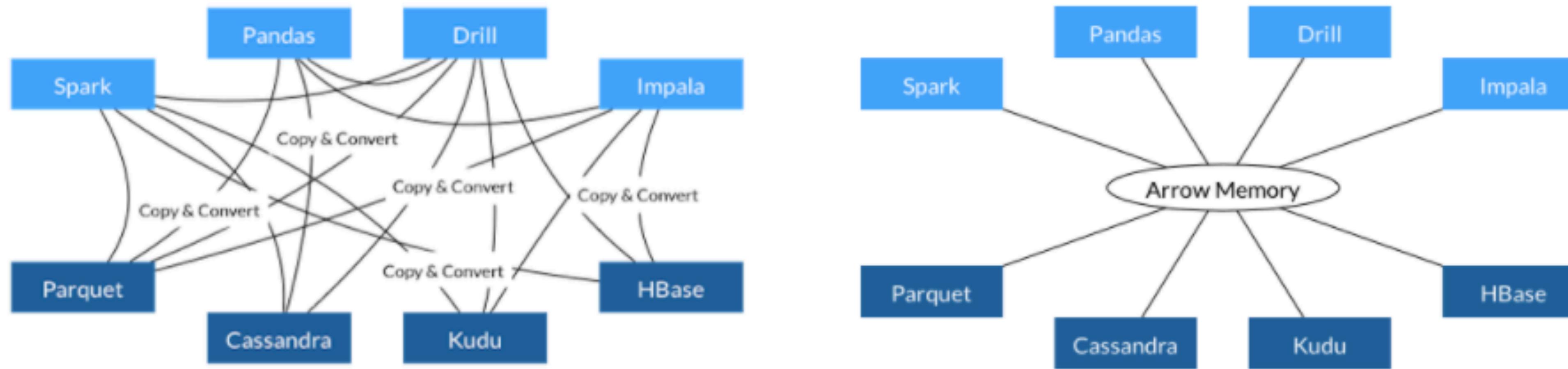
**More complex Flux  
compilations to PromQL?**

**PromQL parser for Flux  
engine?**

# Add Flux into Prometheus?

# Arrow API for Prometheus

# Apache Arrow



# **Stream from Prometheus**

# **Pushdown matcher and range**

**Later pushdown more?**

# **Standardized Remote Read API?**

**Arrow is becoming the lingua franca in data science and big data**

```
fromProm(query: `__name__=~/node_.*/`)
|> range(start:-1h)
|> toCSV(file: "node-data.csv")
|> toFeather(file: "node-data.feather")
```

**Much more work to be done...**

**Prometheus + Flux =  
Possibilities**

# Thank you

Paul Dix  
@pauldix  
[paul@influxdata.com](mailto:paul@influxdata.com)