In-Memory Columnar Store for PostgreSQL

Horizontal data representation

Symbol	Day	Open	High	Low	Close	Volume
AAA	2014-04-27	10.11	10.25	10.08	10.15	125
AAB	2014-04-27	40.33	40.50	40.20	40.45	70
ABB	2014-04-27	25.04	27.13	25.04	26.64	908
ADC	2014-04-27	108.06	110.76	105.03	110.45	745

Vertical data representation

Symbol	Day	Open	High	Low	Close	Volume
AAA	2014-04-27	10.11	10.25	10.08	10.15	125
AAB	2014-04-27	40.33	40.50	40.20	40.45	70
ABB	2014-04-27	25.04	27.13	25.04	26.64	908
ADC	2014-04-27	108.06	110.76	105.03	110.45	745

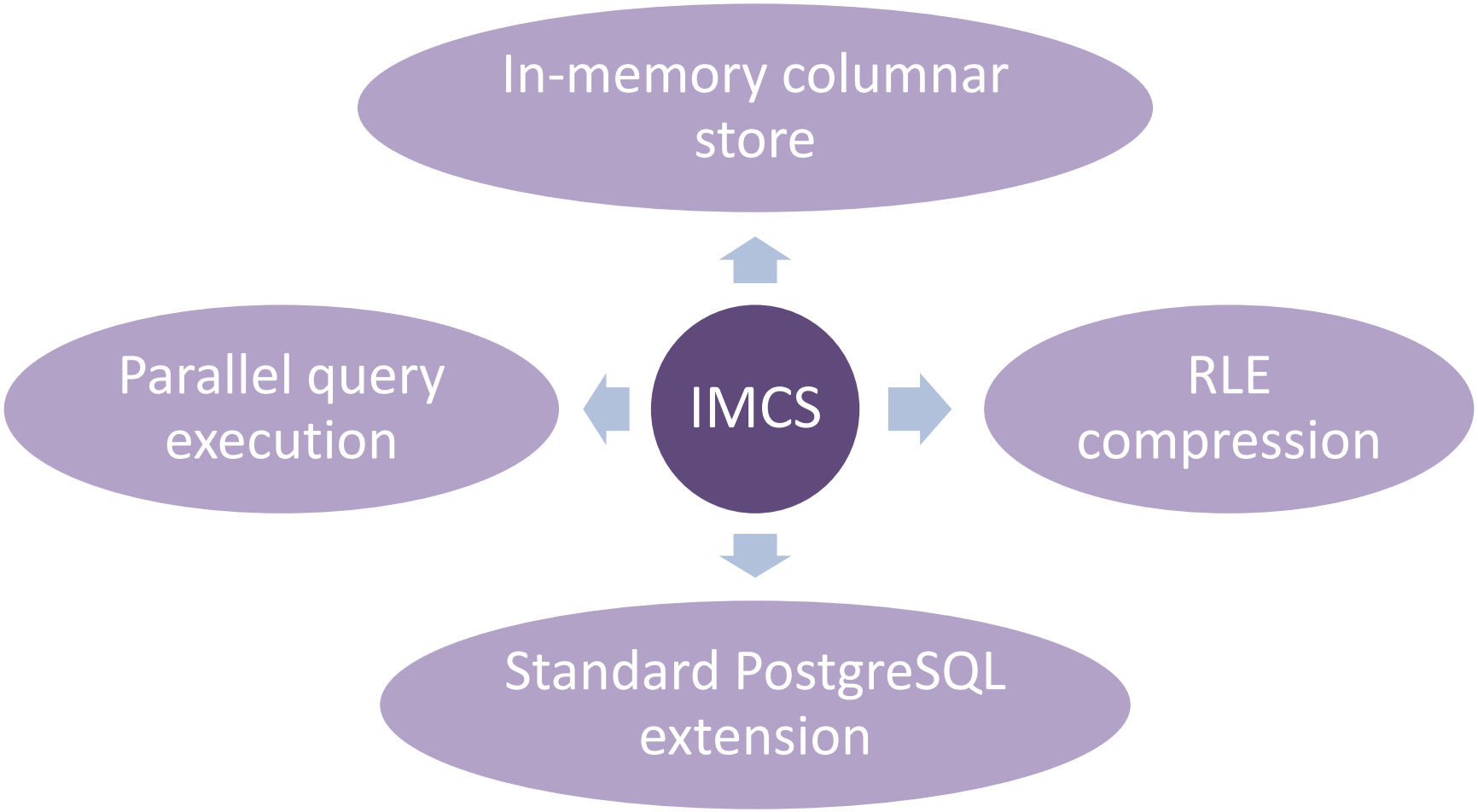
Advantages of vertical model

- ✓ Reducing size of fetched data: only columns involved in query are accessed.
- ✓ Vector operations. Applying an operator to set of values makes it possible to minimize interpretation cost.
- ✓ Use SIMD instructions of modern processors to accelerate execution of vector operations.
- ✓ Compression of data. Such simple compression algorithm like RLE allows not only to reduce used space, but also minimize number of performed operations.

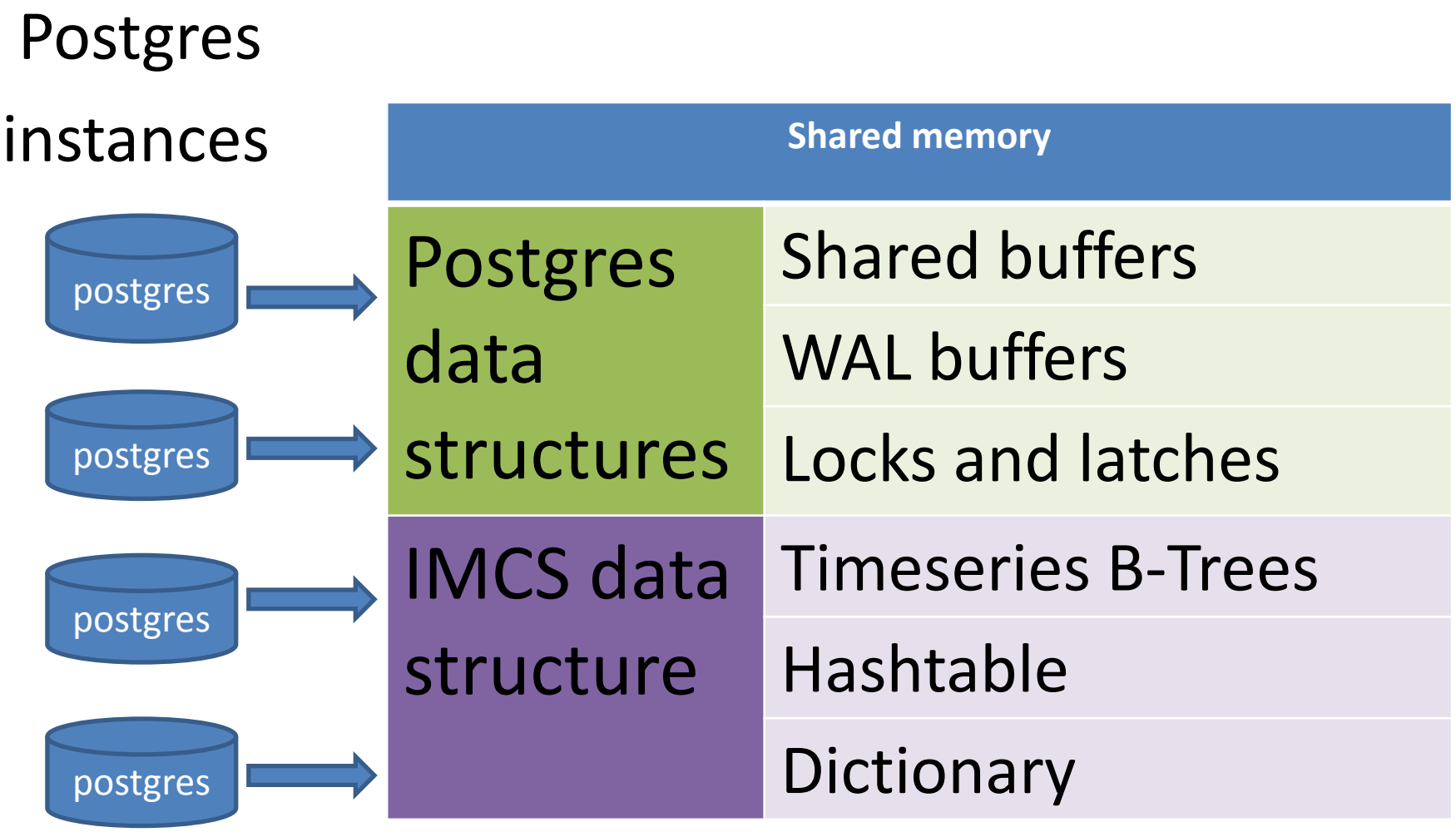
PostgreSQL and OLAP

- PostgreSQL is optimized for OLTP and can execute many queries concurrently, but it is not able to provide parallel execution of single complex OLAP query.
- PostgreSQL uses MVCC model which cause larger per-record space overhead, comparable for some timeseries with size of element.
- Pool management, locking and transaction overhead.

IMCS principles

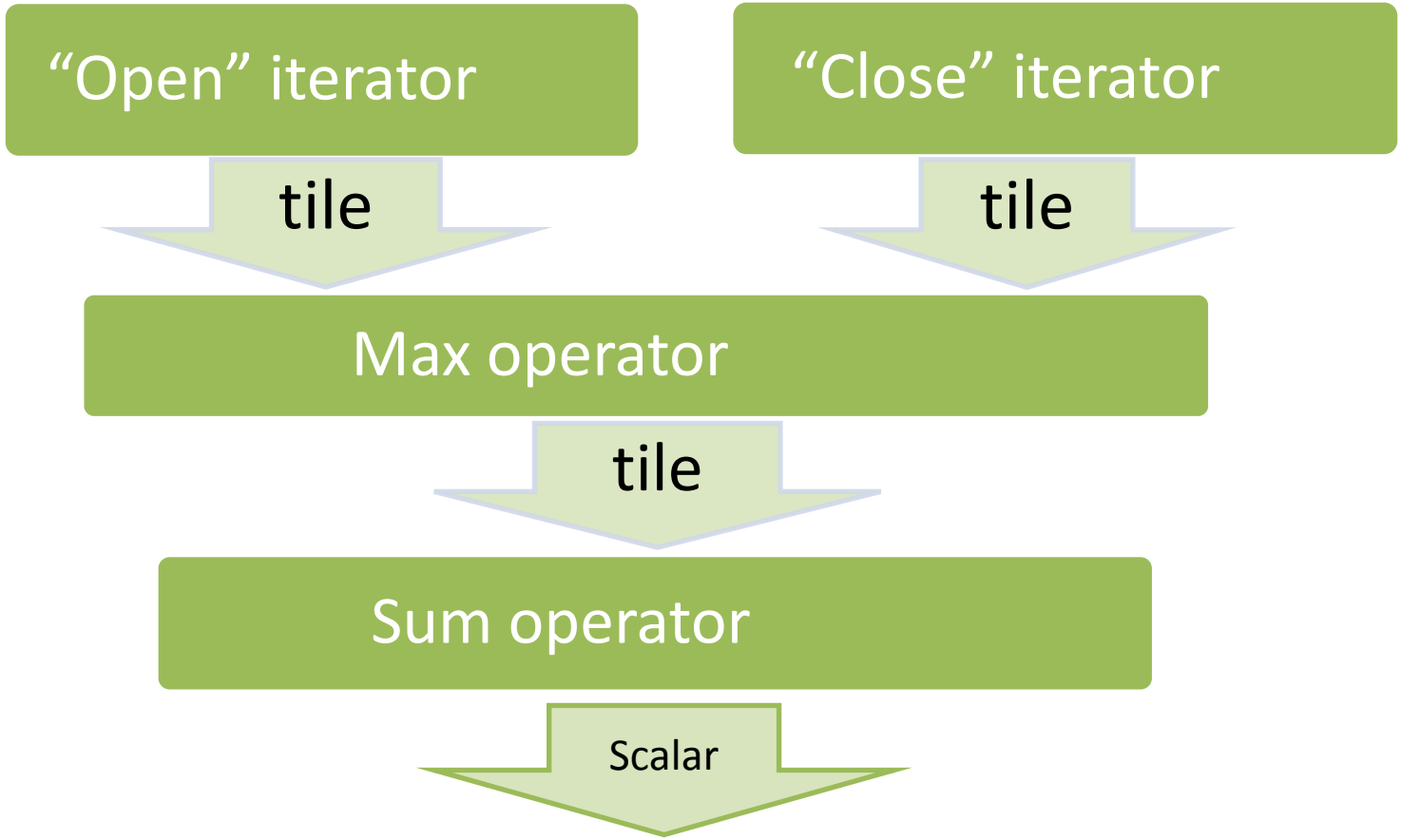


IMCS architecture



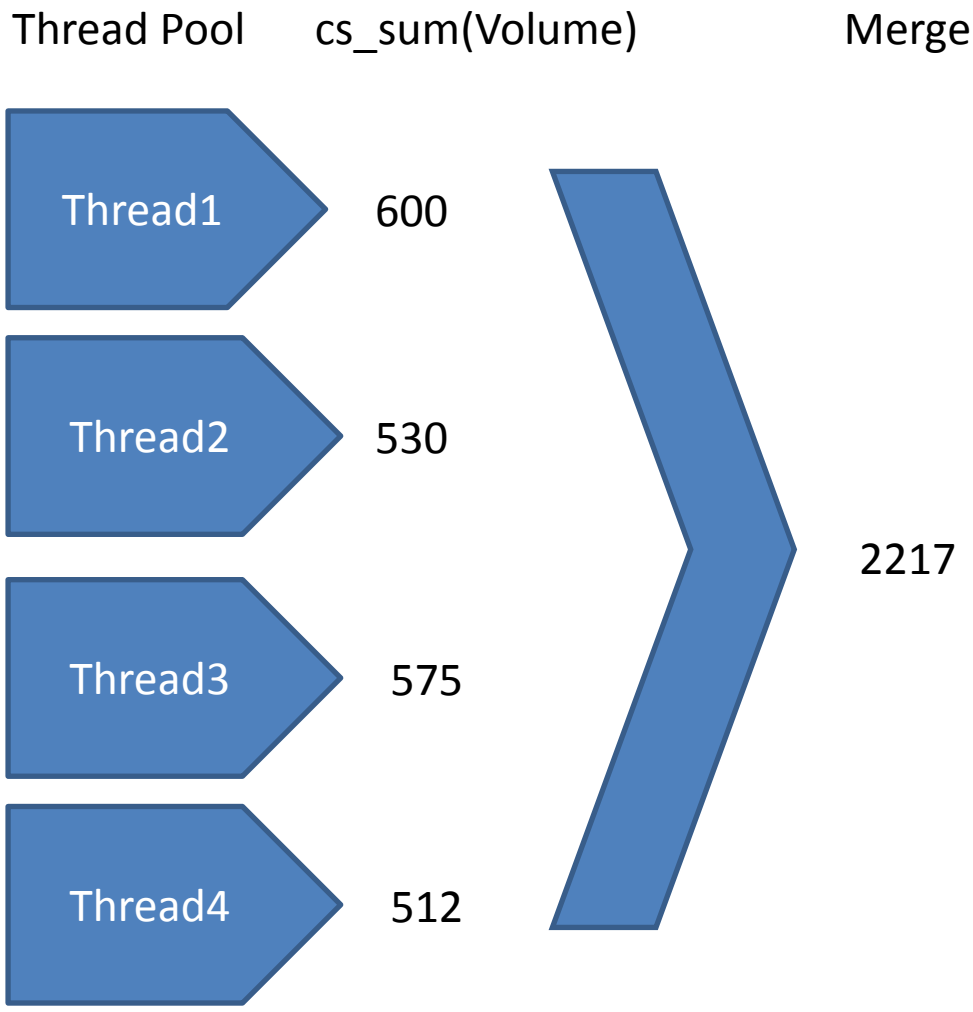
Operators pipeline

select cs_sum(cs_max(Open, Close)) from Quote_get();



Parallel query execution

Day	Volume
04/30/2014	100
05/01/2014	300
05/02/2014	200
05/05/2014	150
05/06/2014	170
05/07/2014	210
05/08/2014	190
05/09/2014	180
05/12/2014	205
05/13/2014	185
05/13/2014	177
05/15/2014	150



IMCS usage

Load IMCS module

```
create extension imcs;
```

Generate functions

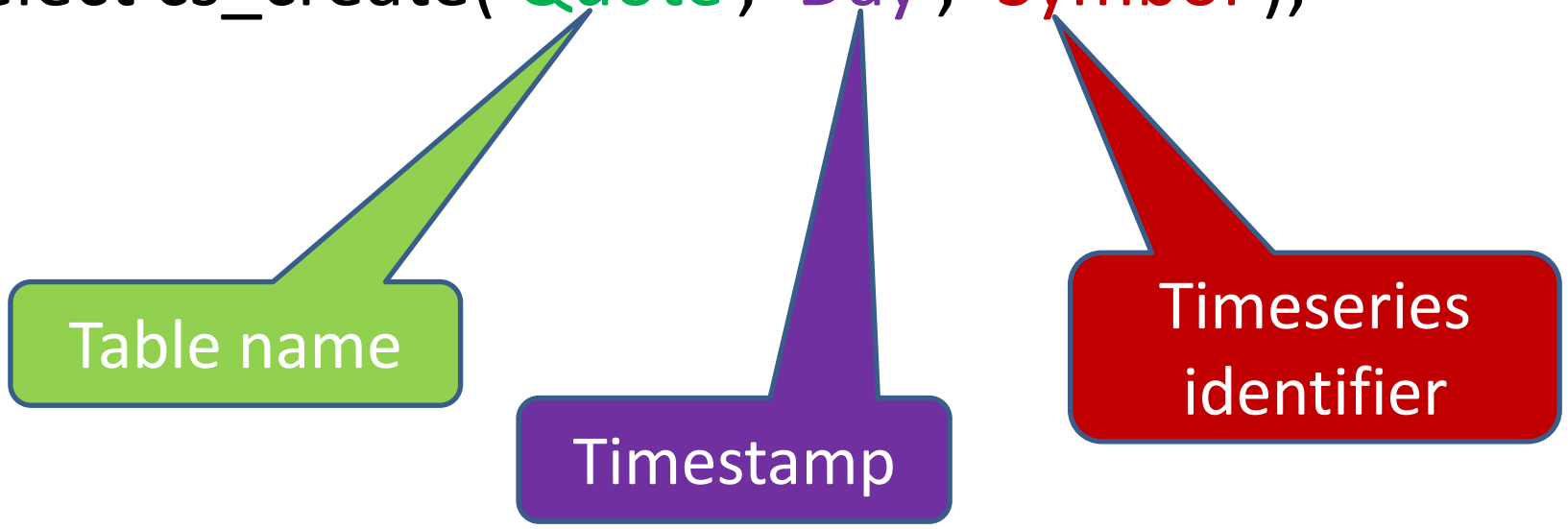
```
select cs_create('Quote', 'Day', 'Symbol');
```

Import data

```
select Quote_load();
```

IMCS function generator

```
select cs_create('Quote', 'Day', 'Symbol');
```



Special functions for IMCS queries

Standard SQL query

- select sum(Close) from Quote where Symbol=‘ABB’;

IMCS query

- select cs_sum(Close) from Quote_get(‘ABB’);

Projection from vertical to horizontal

Show top 10 IBM quotes with maximal close price for first quarter of 2014:

```
select (Quote_project(ibm.*,
          cs_top_max_pos(Close, 10))).*
from Quote_get('IBM',
                    '01-Jan-2014',  '31-Mar-2014') ibm;
```

PosgreSQL user defined operators

```
select cs_filter(Close > (High – Low) / 2, Date)
from Quote_get('IBM');
```

```
select cs_hash_sum(Close*Volume,
                   Exchange||Symbol)
from Stock_get();
```

Specialized operators

Calculate volume weighted average price (VWAP)

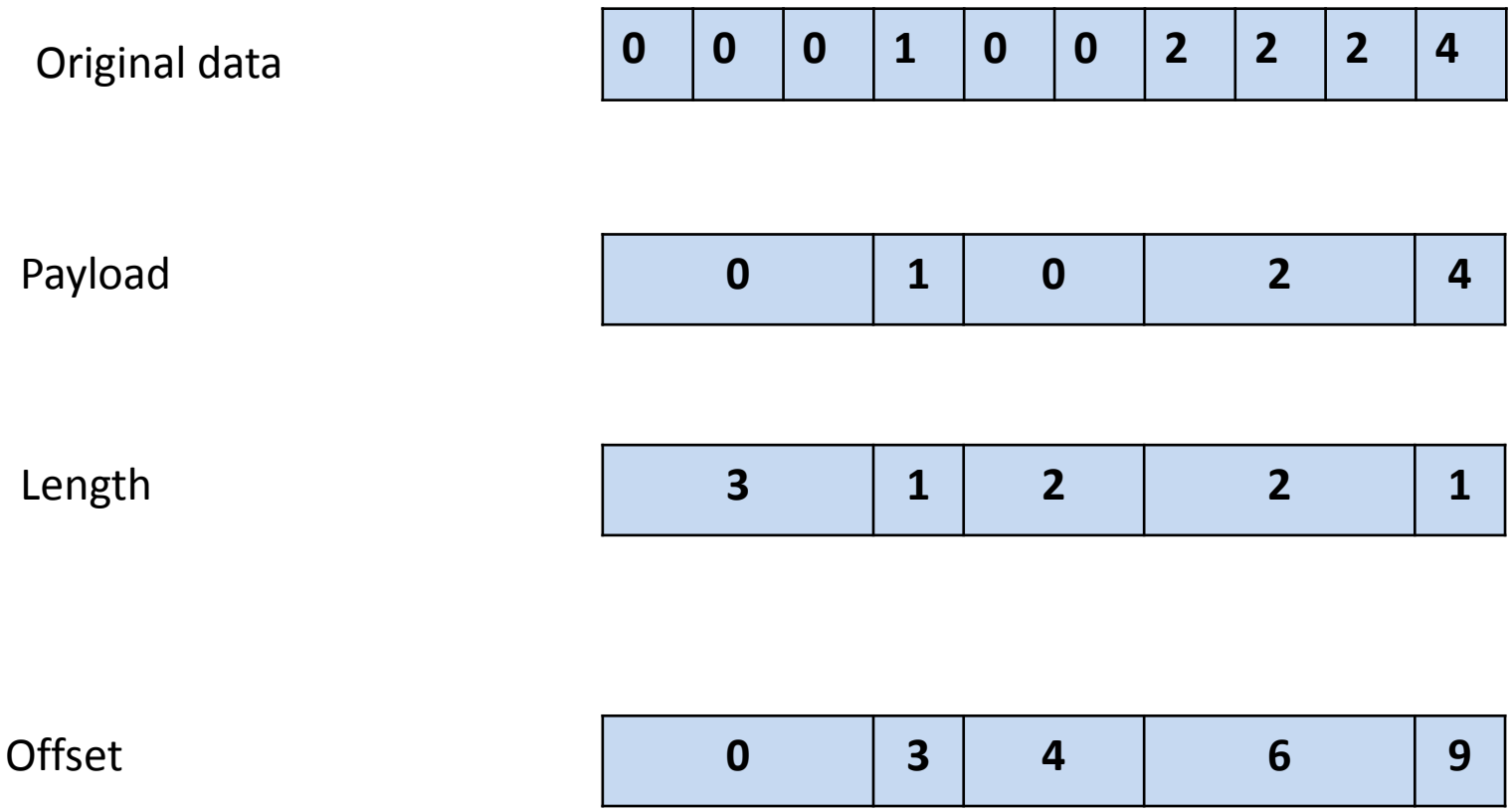
Standard SQL

```
select
    sum(Close*volume)
 / sum(Volume) as
   VWAP
from Quote;
```

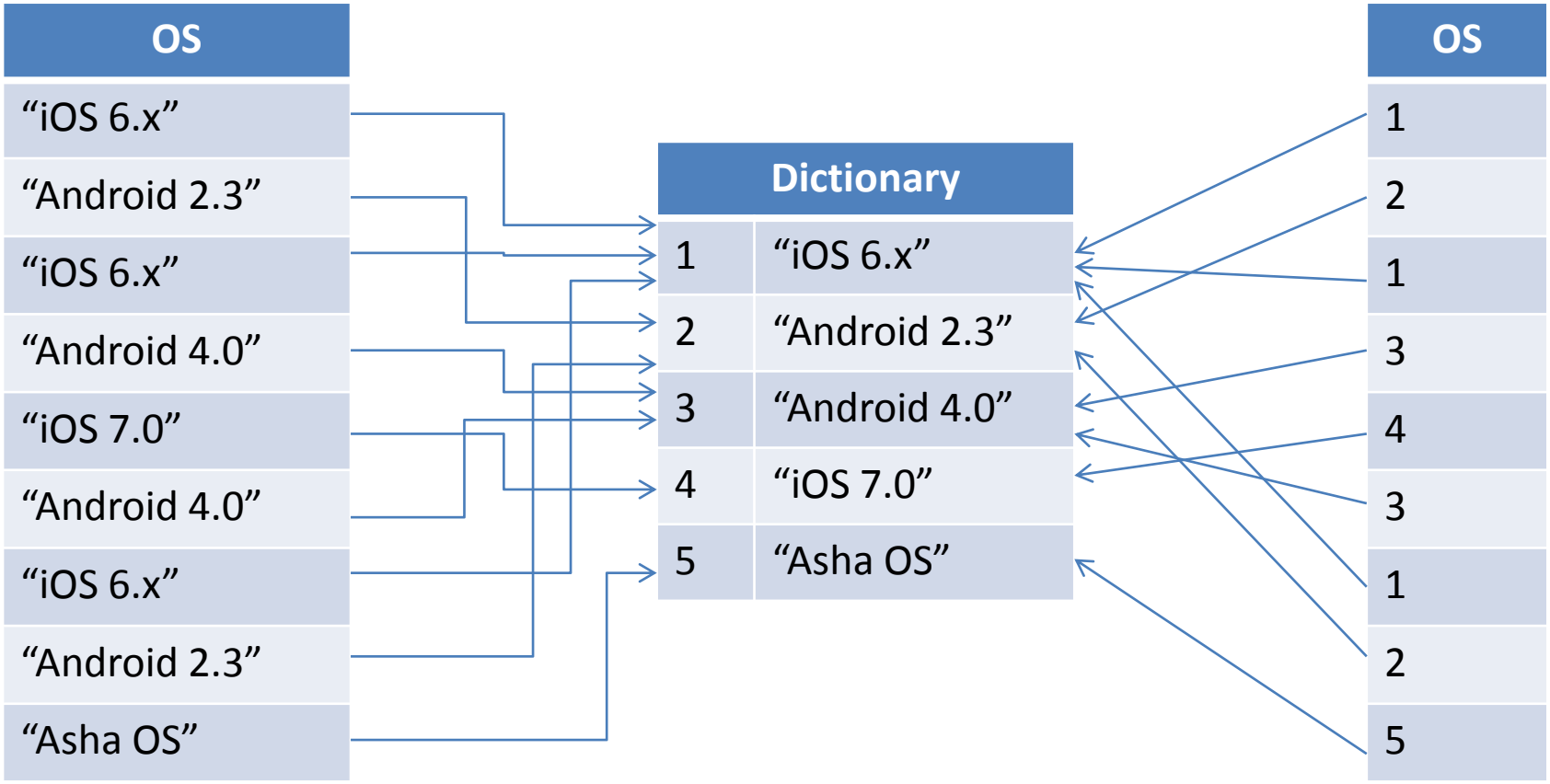
IMCS

```
select
    Volume//Close as
    VWAP
from Quote_get();
```

RLE encoding



Replacing strings with integer codes



Timeseries slices

Description	IMCS function
Get data for 01-May-2014	Quote_get('IBM', '01-May-2014')
Get data for May	Quote_get('IBM', '01-May-2014', '31-May-2014')
Get first three elements of timeseries	Quote_span('IBM', till_pos:=2)
Get last ten elements of timeseries	Quote_span('IBM', from_pos:=-10)
Get all quotes for ABB and IBM starting from 1 of May	Quote_get(array['ABB','IBM'], from_ts:='1-May-2014')

Aggregates

Grand aggregates

- select cs_sum(Close) from Quote_get();

Group-by aggregates

- select cs_group_max(Volume, Day/7) from Quote_get();

Grid aggregates

- select cs_grid_avg(Close, 5) from Quote_get()

Window (moving) aggregates

- select cs_window_avg(Close, 3) from Quote_get();

Hash aggregates

- Select cs_hash_sum(Volume, Exchange) from Quote_get();

Cumulative aggregates

- Select cs_prd(SplitFactor) from Split_get();

Queries for financial indicator

Average True Range (ATR) indicator with 14 days period for first quarter of ABB

```
select cs_window_atr(cs_maxof(High-Low, 0|||cs_maxof(
cs_abs((High<<1) – Close), cs_abs((Low<<1) – Close))),
14)  << 13
from Quote_get('ABB', '01-Jan-2014', '31-Mar-2014');
```

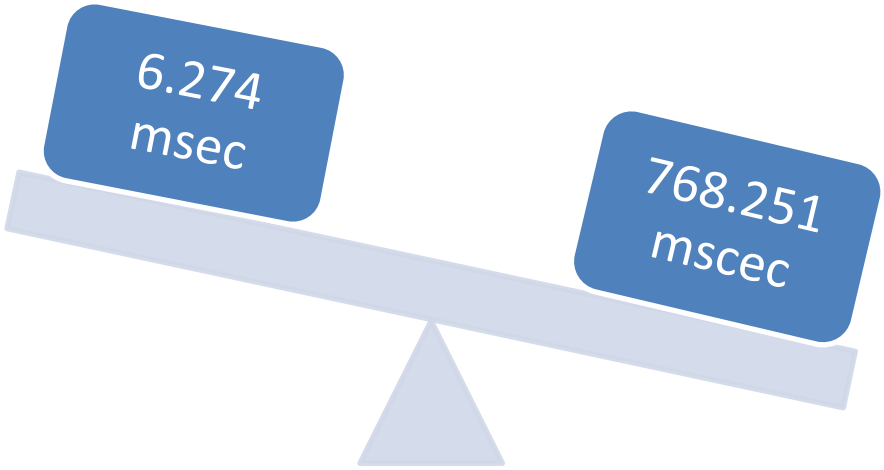
Relative Strength Index (RSI) indicator with 14 days period for first quarter of ABB

```
select 100-(100/(1+cs_window_ema(cs_maxof(cs_diff(Close), 0),
14) / cs_window_ema(cs_maxof(-cs_diff(Close), 0), 14)))
from Quote_get('ABB', '01-Jan-2014', '31-Mar-2014');
```

Performance comparison

select cs_sum (Close>Open*1.1) from Quote_get()

select count(*) from Quote where Close>Open*1.1;



More info?	http://www.garret.ru/imcs/user_guide.html
Try?	http://github.com/knizhnik/imcs
Contact?	mailto://knizhnik@garret.ru