Use Cases for BNF of block based encoding

Contents

- Efficient storage of binding values
  - Naive append of binding values
  - Out of line storage of data
  - Nav_header optimization
- Heterogeneous type handling

Query

- Select s.sid, s.type, s.readings from sensors s

Schema

Sensors:[{ sid: number, stype: string, readings:

[{ time: timestamp, rvalue: number}]

}]

Data

Data: [{sid: 1, stype: 'temperature', readings:

[{time: '2014-01-01:00:00:00', rvalue: 50

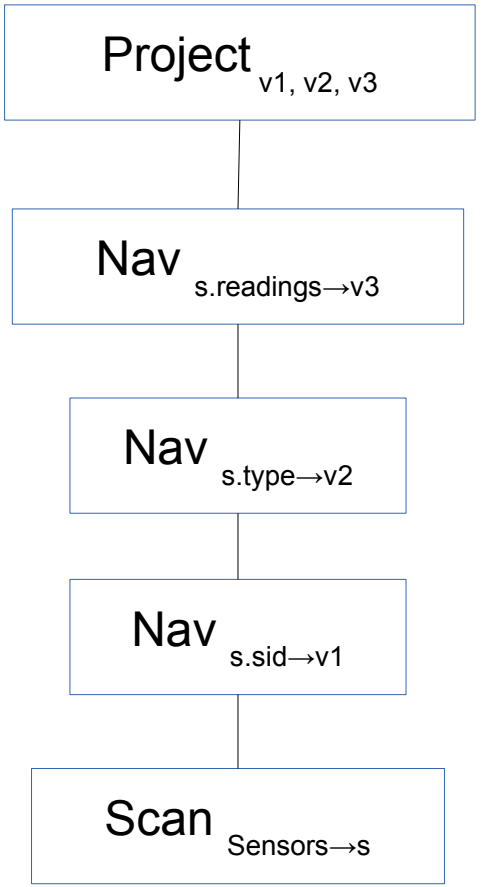
}... 10 items]

},{sid: 2, stype: 'pressure' ,readings:

[{time: '2014-01-01:00:00:00', rvalue: 101}... 10 items]

}]

Plan


Project
v1, v2, v3
Nav
s.readings→v3
Nav
s.type→v2
Nav
s.sid→v1
Scan
Sensors→s

Naive append of binding values - Header

- First input binding at Project

array_header:0x05

Tuple_header:0x04 is_open:0 num_of_attributes:4

attribute_header: attribute_name:s header:tuple_header:0x04 is_open:0 num_of_attributes:3

attribute_header: attribute_name:sid header:number_header:0x03

attribute_header: attribute_name:type header:string_header:0x09

attribute_header: attribute_name:reading header:array_header:0x05

Tuple_header:0x04 is_open:0 num_of_attributes:2

attribute_header: attribute_name:time header:time: timestamp_header:0x10

attribute_header: attribute_name:value header:number_header:0x03

Header copied because of Nav

attribute_header: attribute_name:v1 header:number_header:0x03:

attribute_header: attribute_name:v2 header:string_header:0x09

attribute_header: attribute_name:v3 header:array_header0x05

Tuple_header:0x04 is_open:0 num_of_attributes:2

attribute_header: attribute_name:time header:time: timestamp_header:0x10

attribute_header: attribute_name:value header:number_header:0x03

Naive append of binding values - Data

- First Input binding at Project

header_ref:xx num_of_items:1 item_id:1 static_header_item:tuple_item:dense_tuple_item:0x01 item_bytes:xx

static_header_item:tuple_item:dense_tuple_item:0x01 item_bytes:xx

static_header_item:tuple_item:dense_tuple_item:0x01 item_bytes:xx

item:number_item:1 item:string_item:temperature
item:array_item:overflow:no_overflow:0x00 item_bytes:xx num_of_items:10

tuple_item:dense_tuple_item:0x01 item_bytes:xx

timestamp_item:2014-01-01:00:00:00 number_item:50

.

.

10 items

Binding values copied because of Nav

static_header_item:number_item:1

static_header_item:string_item:temperature

static_header_item: array_item:overflow:no_overflow:0x00 item_bytes:xx num_of_items:10

tuple_item:dense_tuple_item:0x01 item_bytes:xx

timestamp_item:2014-01-01:00:00:00 number_item:50

.

.

10 items

Problem

- Problem: Too much copying of data uses processor time in addition to extra space

- Idea: Store data *out of line* and allow pointers in binding values

Out of line representation - Query

Select s.sid, s.type, s.readings from sensors s

Out of line representation - Header

- First Binding value at Project

array_header:0x05

Tuple_header:0x04 is_open:0 num_of_attributes:4

attribute_header: attribute_name:sid header:tuple_header:0x04 is_open:0 num_of_attributes:3

attribute_header: attribute_name:sid header:number_header:0x03

attribute_header: attribute_name:type header:string_header:0x09

attribute_header: attribute_name:reading header:pointer_header:0x07

attribute_header: attribute_name:v1 header:number_header:0x03:

attribute_header: attribute_name:v2 header:string_header:0x09

attribute_header: attribute_name:v3 header:pointer_header0x07

Out of line representation - Data

- First Binding value at Project

header_ref:xx num_of_items:1 item_id:1
static_header_item:tuple_item:dense_tuple_item:0x01 item_bytes:xx

static_header_item:tuple_item:dense_tuple_item:0x01 item_bytes:xx

item:number_item:1

item:string_item:temperature

item:pointer_item:block_ref:xx item_ref:xx

static_header_item:number_item:1

static_header_item:string_item:temperature

static_header_item:pointer_item:block_ref:xx item_ref:xx

Pointer Data

- Out of line stored Pointer Data

array_item:overflow:no_overflow:0x00 item_bytes:xx num_of_items:10

tuple_item:dense_tuple_item:0x01 item_bytes:xx

timestamp_item:2014-01-01:00:00:00 number_item:50

.

.

10 items

Problem

- Problem: Works for complex values and strings. Other scalar values are still copied to the tuple because of *Navs*

- Idea: Use pointers (or indexes) into tuple *Nav* values refer to

Nav Header Based Representation-Query

- Select s.sid, s.type, s.readings from sensors s

Nav Header Based Representation - Header

- First Binding value at Project

array_header:0x05

Tuple_header:0x04 is_open:0 num_of_attributes:4

attribute_header: attribute_name:s header:tuple_header:0x04 is_open:0 num_of_attributes:3

attribute_header: attribute_name:sid header:number_header:0x03

attribute_header: attribute_name:type header:string_header:0x09

attribute_header: attribute_name:reading header:pointer_header:0x07

attribute_header: attribute_name:v1 header:nav_header:0x08

attribute_header: attribute_name:v2 header:nav_header:0x08

attribute_header: attribute_name:v3 header:nav_header:0x08

Nav Header Based Representation - Data

- First Binding value at Project

header_ref:xx num_of_items:1 item_id:1
static_header_item:tuple_item:dense_tuple_item:0x01 item_bytes:xx

static_header_item:tuple_item:dense_tuple_item:0x01 item_bytes:xx

item:number_item:1 item:string_item:temperature
item:pointer_item:block_ref:xx item_ref:xx

static_header_item:nav_item:sibling_index:0 num_of_steps:1
step:attribute_index1

static_header_item:nav_item:sibling_index:0 num_of_steps:1
step:attribute_index2

static_header_item:nav_item:sibling_index:0 num_of_steps:1
step:attribute_index3

Problem

- Problem: How does this scale to hetrogenous types?

- Idea: Use dynamic header type

Dynamic Header - Query

Select s.sid, s.type, s.readings from sensors s

Schema

```
Sensors:[{ sid: number, type: string, reading:
    [{ time: timestamp, value: any}]
}]
```

Data

```
Data: [{sid: 1, type: 'temperature', readings:
    [{time: '2014-01-01:00:00:00', value: 50
    }...]
},{sid: 2, type: 'co' ,readings:
    [{time: '2014-01-01:00:00:00', value: 'low'}...]
}]
```

Naive append of binding values - Header

- First Binding value at Project

array_header:0x05

Tuple_header:0x04 is_open:0 num_of_attributes:3

attribute_header: attribute_name:sid header:number_header:0x03

attribute_header: attribute_name:type header:string_header:0x09

attribute_header: attribute_name:reading header:array_header:0x05

Tuple_header:0x04 is_open:0 num_of_attributes:2

attribute_header: attribute_name:time header:time: timestamp_header:0x10

attribute_header: attribute_name:value header:dynamic_type_header:0x00

attribute_header: attribute_name:v1 header:number_header:0x03:

attribute_header: attribute_name:v2 header:string_header:0x09

attribute_header: attribute_name:v3 header:array_header0x05

Tuple_header:0x04 is_open:0 num_of_attributes:2

attribute_header: attribute_name:time header:time: timestamp_header:0x10

attribute_header: attribute_name:value header:dynamic_type_header:0x00

Naive append of binding values – Data – 'temperature'

- First Binding value at Project

header_ref:xx num_of_items:1 item_id:1 static_header_item:tuple_item:dense_tuple_item:0x01 item_bytes:xx

static_header_item:tuple_item:dense_tuple_item:0x01 item_bytes:xx

item:number_item:1 item:string_item:temperature item:array_item:overflow:no_overflow:0x00 item_bytes:xx num_of_items:10

tuple_item:dense_tuple_item:0x01 item_bytes:xx

timestamp_item:2014-01-01:00:00:00 dyanmic_header_item:0x01 number_item:50

.

.

10 items

static_header_item:number_item:1

static_header_item:string_item:temperature

static_header_item: array_item:overflow:no_overflow:0x00 item_bytes:xx num_of_items:10

tuple_item:dense_tuple_item:0x01 item_bytes:xx

timestamp_item:2014-01-01:00:00:00 dyanmic_header_item:0x01 number_item:50

.

.

10 items

Naive append of binding values – Data – 'co'

- First Binding value at Project

header_ref num_of_items:1 item_id:1 static_header_item:tuple_item:dense_tuple_item:0x01 item_bytes:xx

static_header_item:tuple_item:dense_tuple_item:0x01 item_bytes:xx

item:number_item:1 item:string_item:temperature item:array_item:overflow:no_overflow:0x00 item_bytes:xx num_of_items:10

tuple_item:dense_tuple_item:0x01 item_bytes:xx

timestamp_item:2014-01-01:00:00:00 dyanmic_header_item: string_item:low

.

.

10 items

static_header_item:number_item:1

static_header_item:string_item:temperature

static_header_item: array_item:overflow:no_overflow:0x00 item_bytes:xx num_of_items:10

tuple_item:dense_tuple_item:0x01 item_bytes:xx

timestamp_item:2014-01-01:00:00:00 dyanmic_header_item:0x01 string_item:low

.

.

10 items

Open Questions

- Every tuple is forced to use out of line storage because it is declared in header, even though its data may be small
- Schema changes at each operator and therefore, a mechanism is required to pass create output schema from input schema at each operator
- Pointer header doesn't store schema of data pointer item may refer to
- Cannot have dynamic_header and pointer_header for the same item even though they are orthogonal to each other