Bancos de Dados Geográficos

Sistemas de Bancos de Dados Matriciais

Gilberto Ribeiro de Queiroz

06 de Agosto de 2018

Como armazenar, gerenciar e facilitar a análise de grandes volumes de dados de Observação da Terra?

Utilizar conjunto de arquivos (GeoTIFF, HDF, NetCDF)?

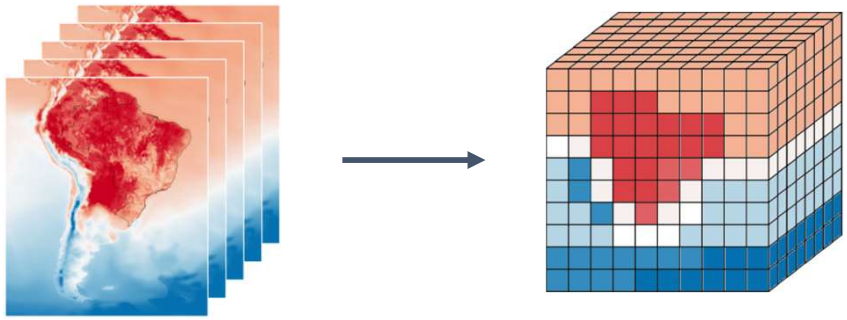
Utilizar um SGBD Relacional (Oracle GeoRaster, PostGIS Raster)?

Utilizar um NoSQL? Qual?

Sistemas de Bancos de Dados Matriciais

SGBD-M

- **Sistemas Gerenciadores de Banco de Dados Matriciais** (SGBD-M) – *Array Databases*
- Modelo de dados trata de "*Arrays*"



Modelo mais apropriado para dados intrinsecamente ordenados de forma espacial e/ou temporal

Array Databases

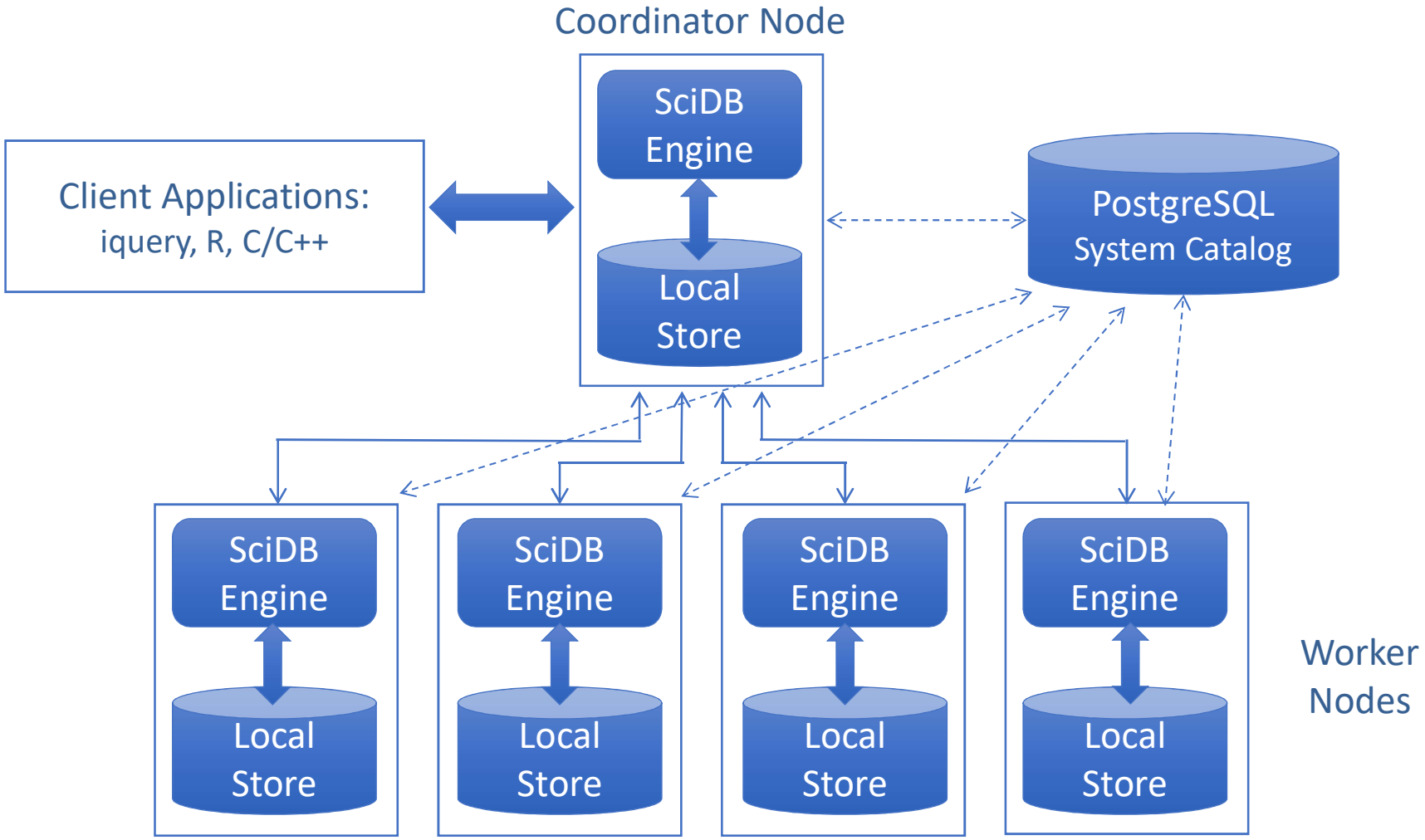
SciDB

Foto: David.Monniaux, CC BY-SA 4.0

"SciDB is an open-source analytical database oriented toward the data management needs of scientists."

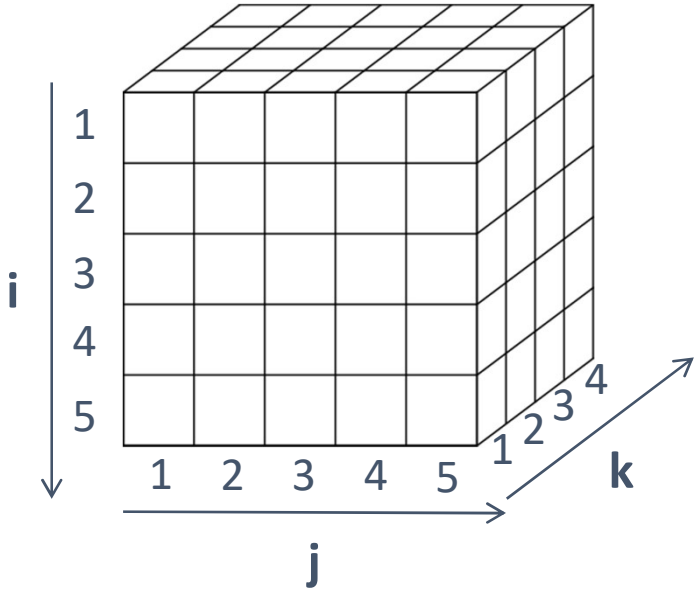
(Stonebraker et al., 2011)

SciDB: Arquitetura



Fote: Adaptado de Paradigm4 (2016)

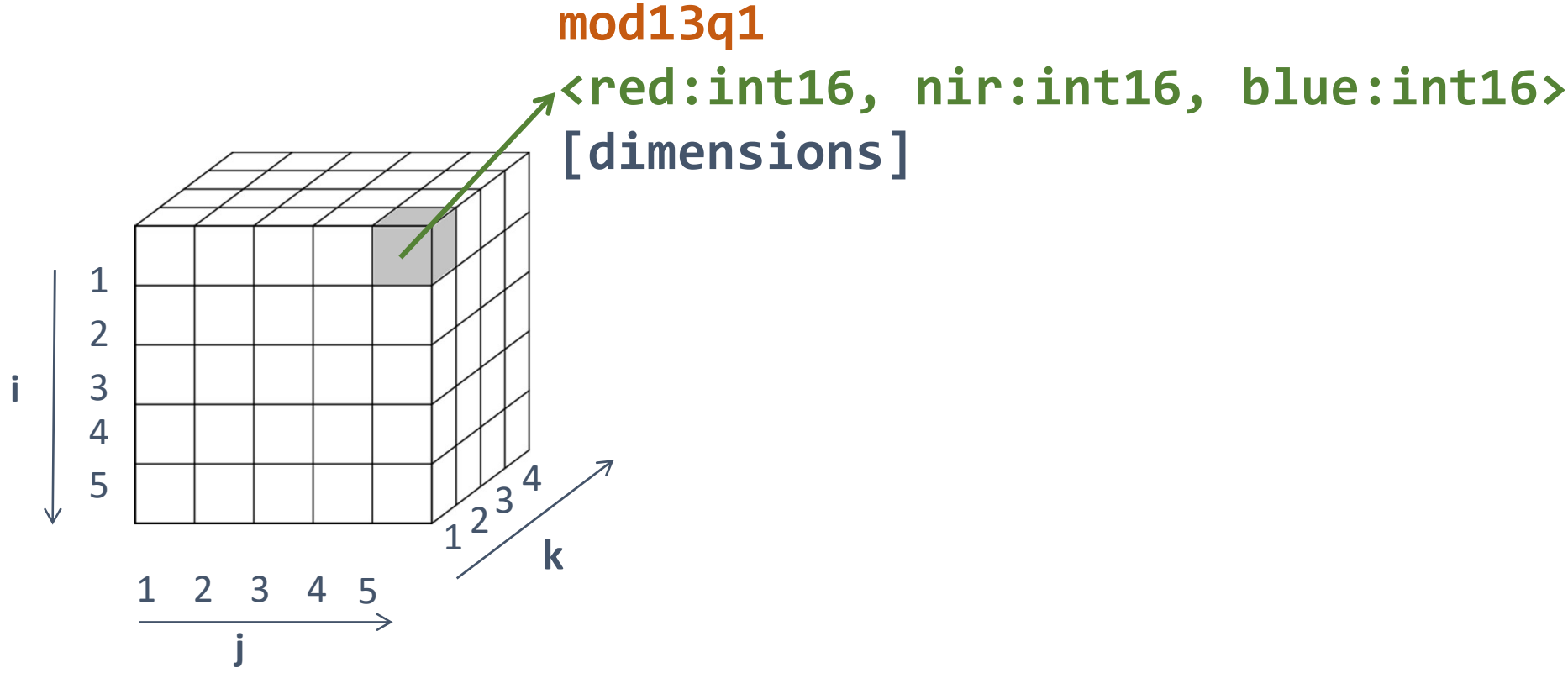
Arrays



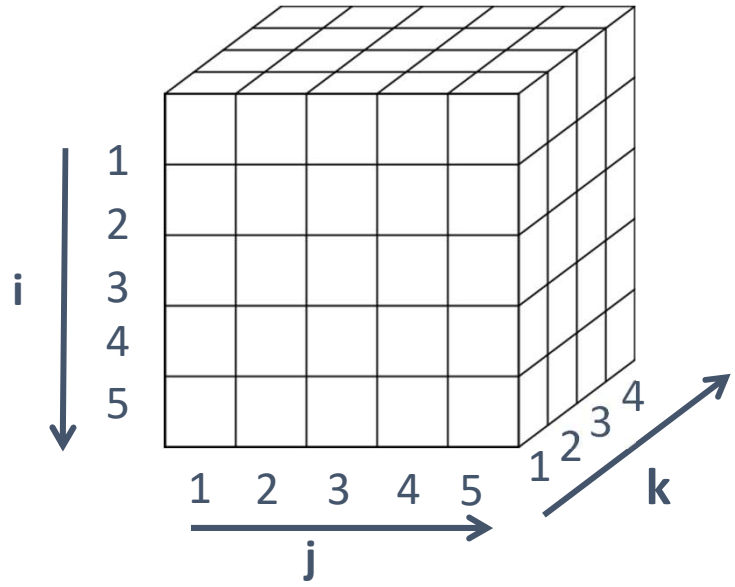
nome <atributos> [dimensões]

SciDB: Atributos

- Cada célula pode ser associada a múltiplos valores.
- Tipos de dados: `bool, char, datetime, datetimez, double, float, int8, int16, int32, int64, string, uint8, uint16, uint32, uint64`

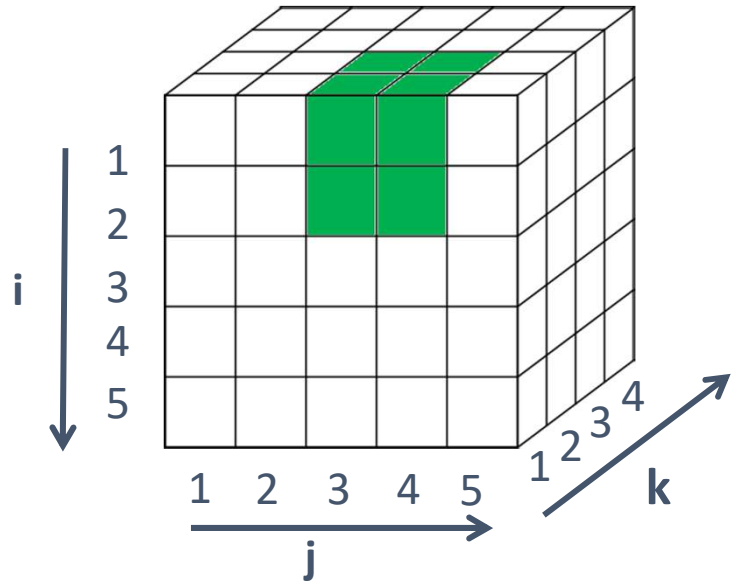


SciDB: Dimensões




```
mod13q1 <red:int16, nir:int16, blue:int16>
[j=1:5,2,1, i=1:5,2,1, k=1:4,2,1]
```

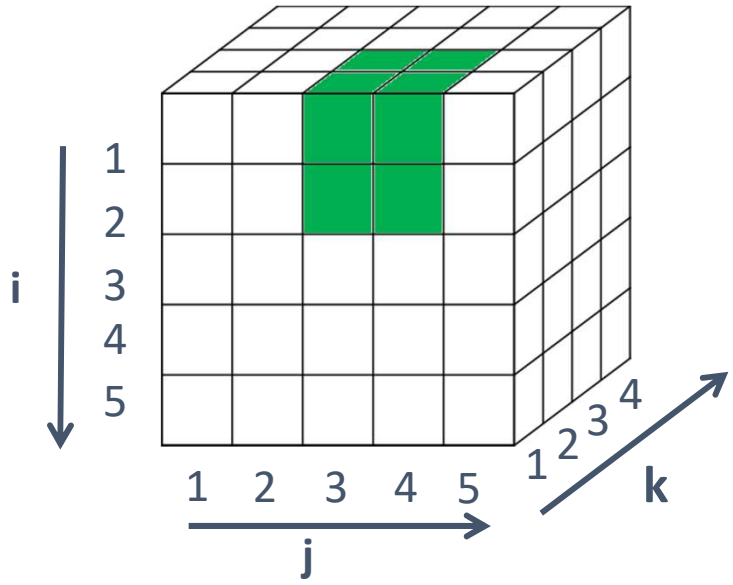
SciDB: Chunks

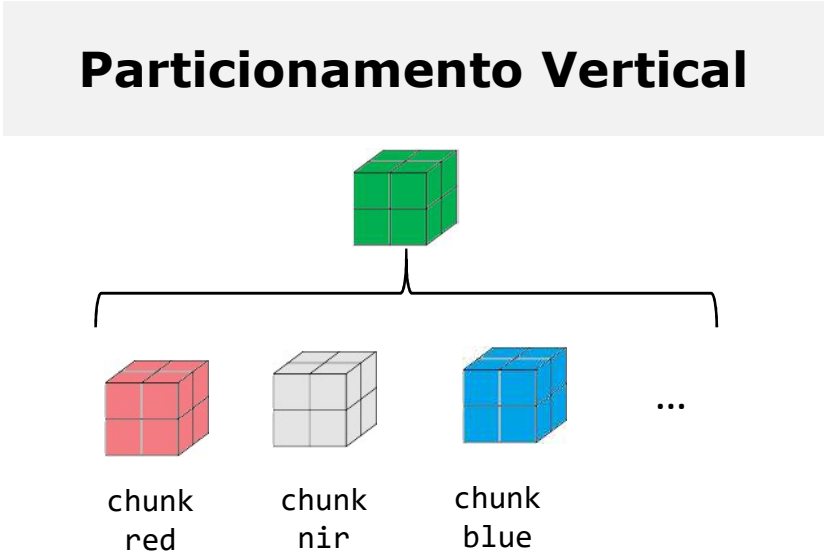



```
mod13q1 <red:int16, nir:int16, blue:int16>
[j=1:5,2,1, i=1:5,2,1, k=1:4,2,1]
```

Chunk size: **2** x **2** x **2**

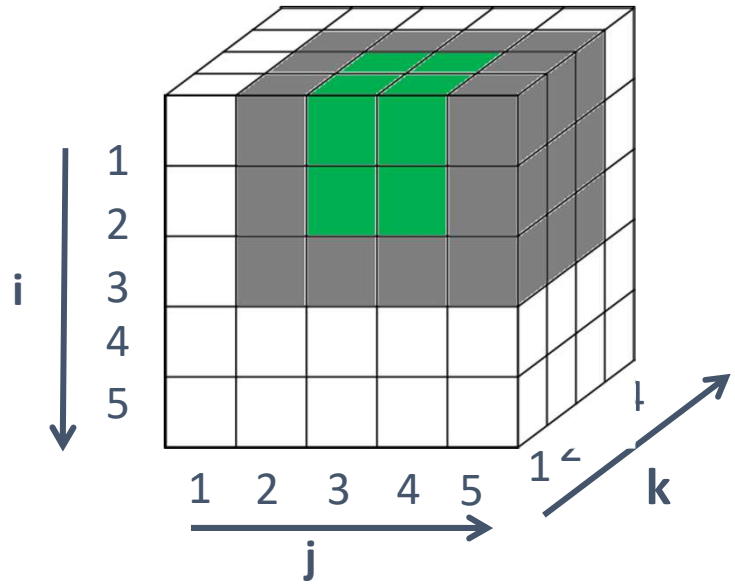
SciDB: Particionamento Vertical






```
mod13q1 <red:int16, nir:int16, blue:int16>
[j=1:5,2,1, i=1:5,2,1, k=1:4,2,1]
```

SciDB: Overlap (Replicação)




```
mod13q1 <red:int16, nir:int16, blue:int16>
[j=1:5,2,1, i=1:5,2,1, k=1:4,2,1]
```

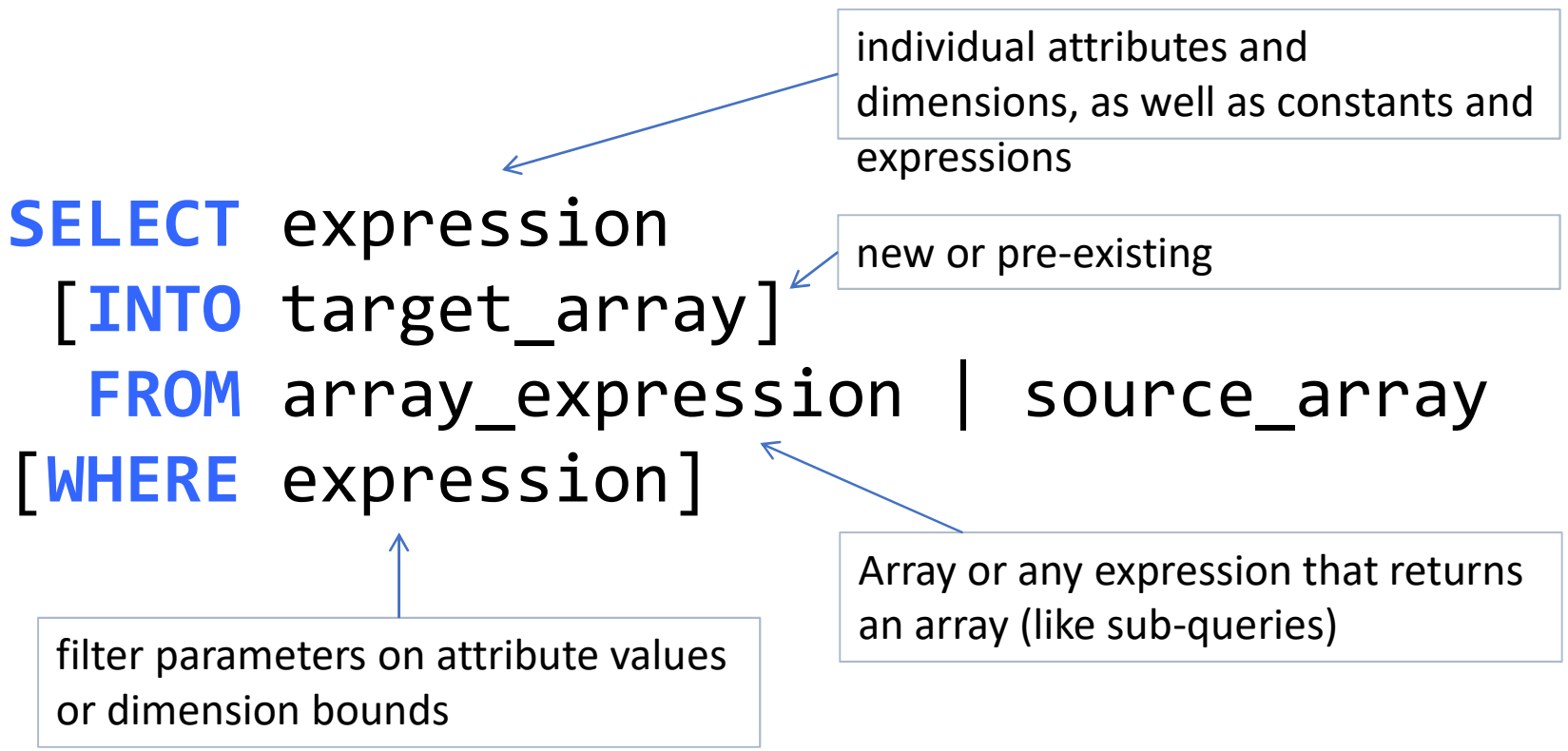
Chunk overlap: 1 x 1 x 1

Linguagens de Consulta

Array **Q**uery **L**anguage (AQL)

Array **F**unctional **L**anguage (AFL)

Array Query Language: AQL



There are DML and DDL clauses

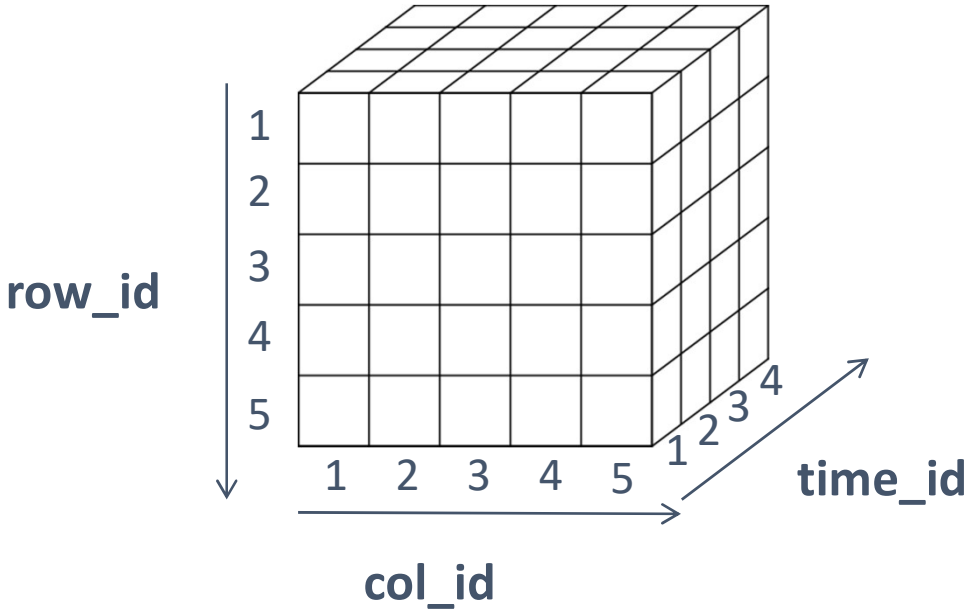
Array Functional Language (AFL)

```
store(
  build(<num:double>
        [x=0:8,1,0, y=0:9,1,0],
        random()
  ),
  random_numbers);
```

Consultas em AFL

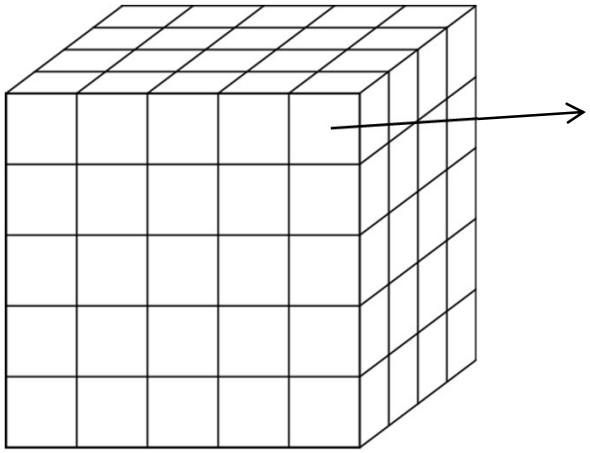
Definindo um Array: `mod13q1`

```
CREATE ARRAY mod13q1
<nir:double, red:double, blue:double>
[col_id=0:4,1,0, row_id=0:4,1,0,
 time_id=0:3,4,0]
```



Creating Array: mod13q1

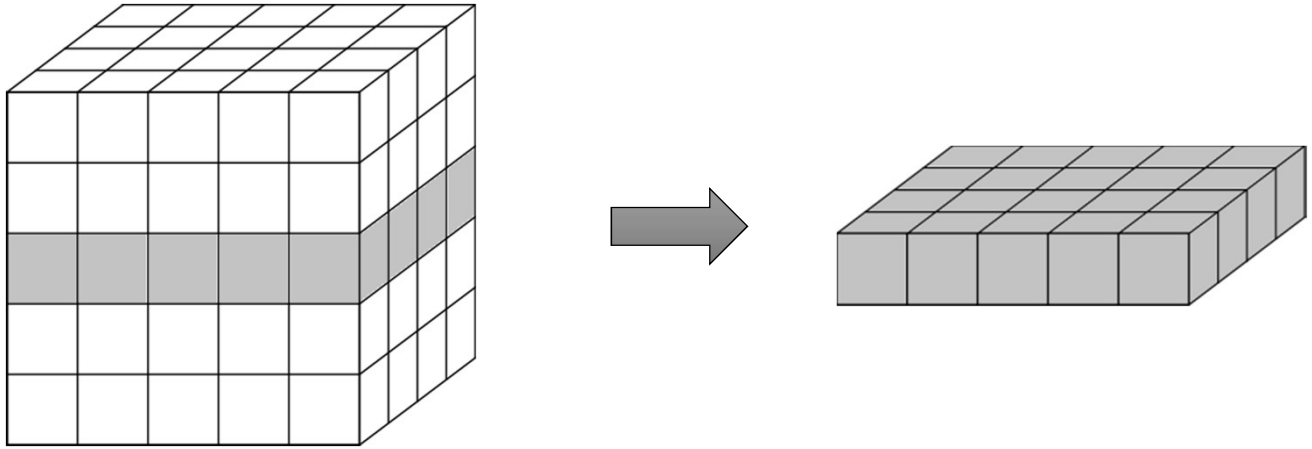
```
store(join(join(
build(<val:double>[col_id=0:4,1,0,row_id=0:4,1,0,time_id=0:3,4,0]
,(col_id+(row_id*5)+time_id*(5*5))/1.0),
build(<val:double>[col_id=0:4,1,0,row_id=0:4,1,0,time_id=0:3,4,0]
,(col_id+(row_id*5)+time_id*(5*5))/10.0)),
build(<val:double>[col_id=0:4,1,0,row_id=0:4,1,0,time_id=0:3,4,0]
,(col_id+(row_id*5)+time_id*(5*5))/100.0) ), mod13q1)
```



```
nir   = val/1.0
red   = val/10.0
blue = val/100.0
```

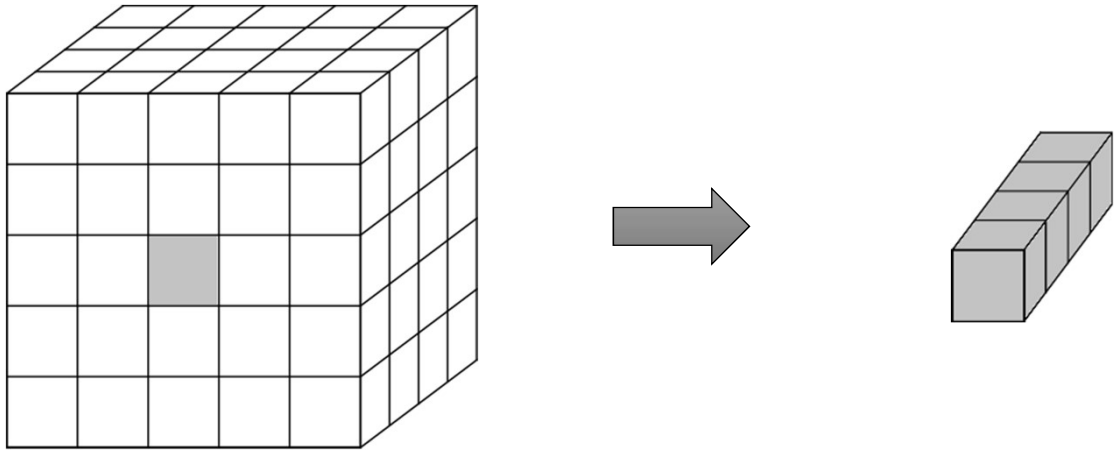
Horizontal Slice

```
subarray(mod13q1,
         0, 2, 0,
         4, 2, 3)
```



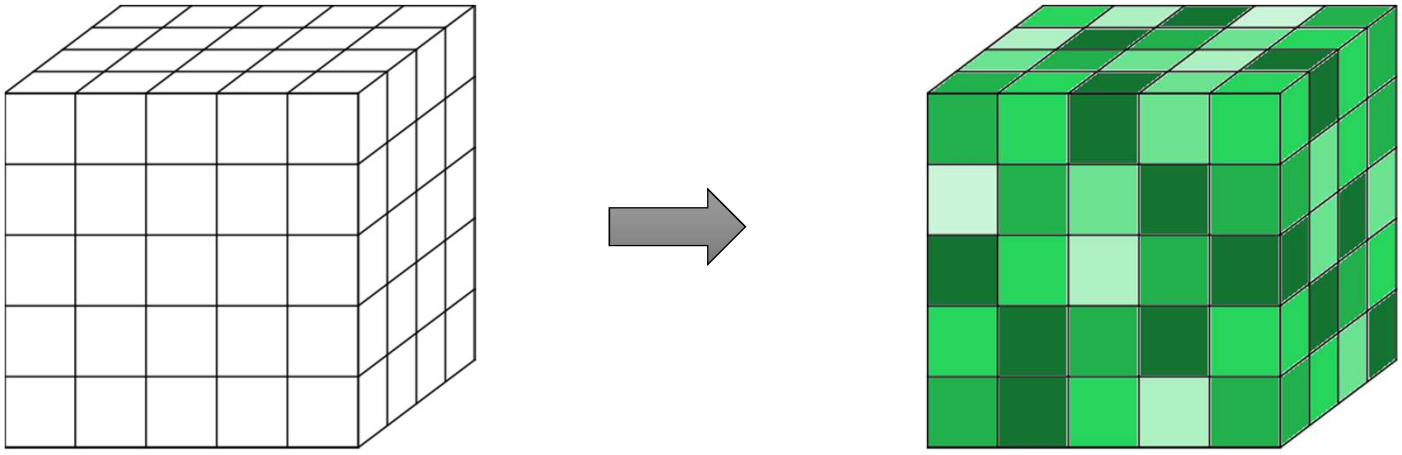
Time Series

```
subarray(mod13q1,
         2, 2, 0,
         2, 2, 3)
```



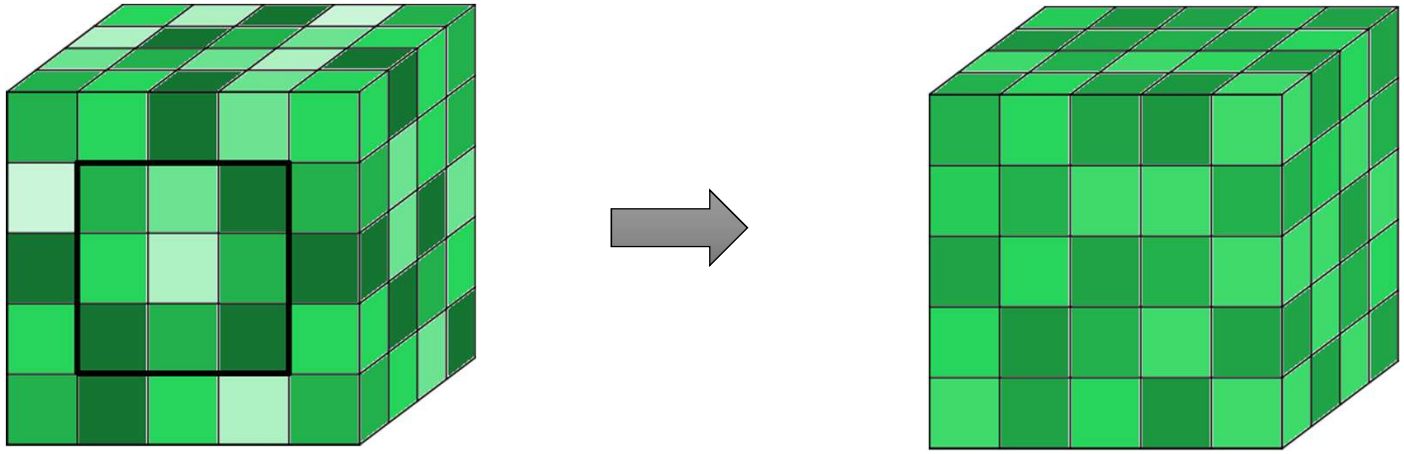
NDVI

```
store (
  project (
    apply (
      mod13q1, new_evi,
      2.5*(nir-red) /(nir+6.0*red-7.5*blue+1.)
    ),
    new_evi ), evi_array);
```



Window Queries

```
store(
    window(evi_array,
             0, 2,
            0, 2,
        0, 0,
        avg(new_evi)
    ),
   evi_avg);
```

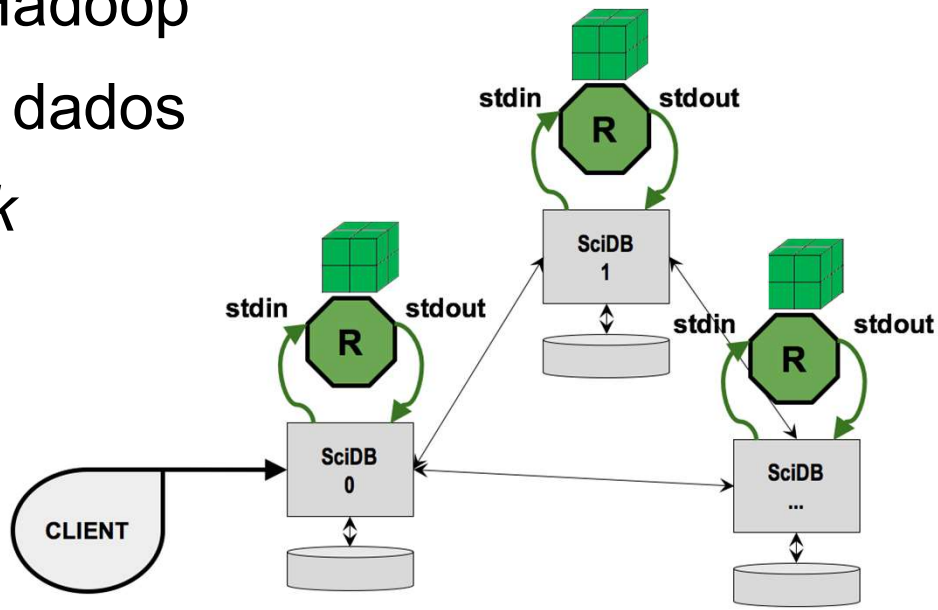


Stream

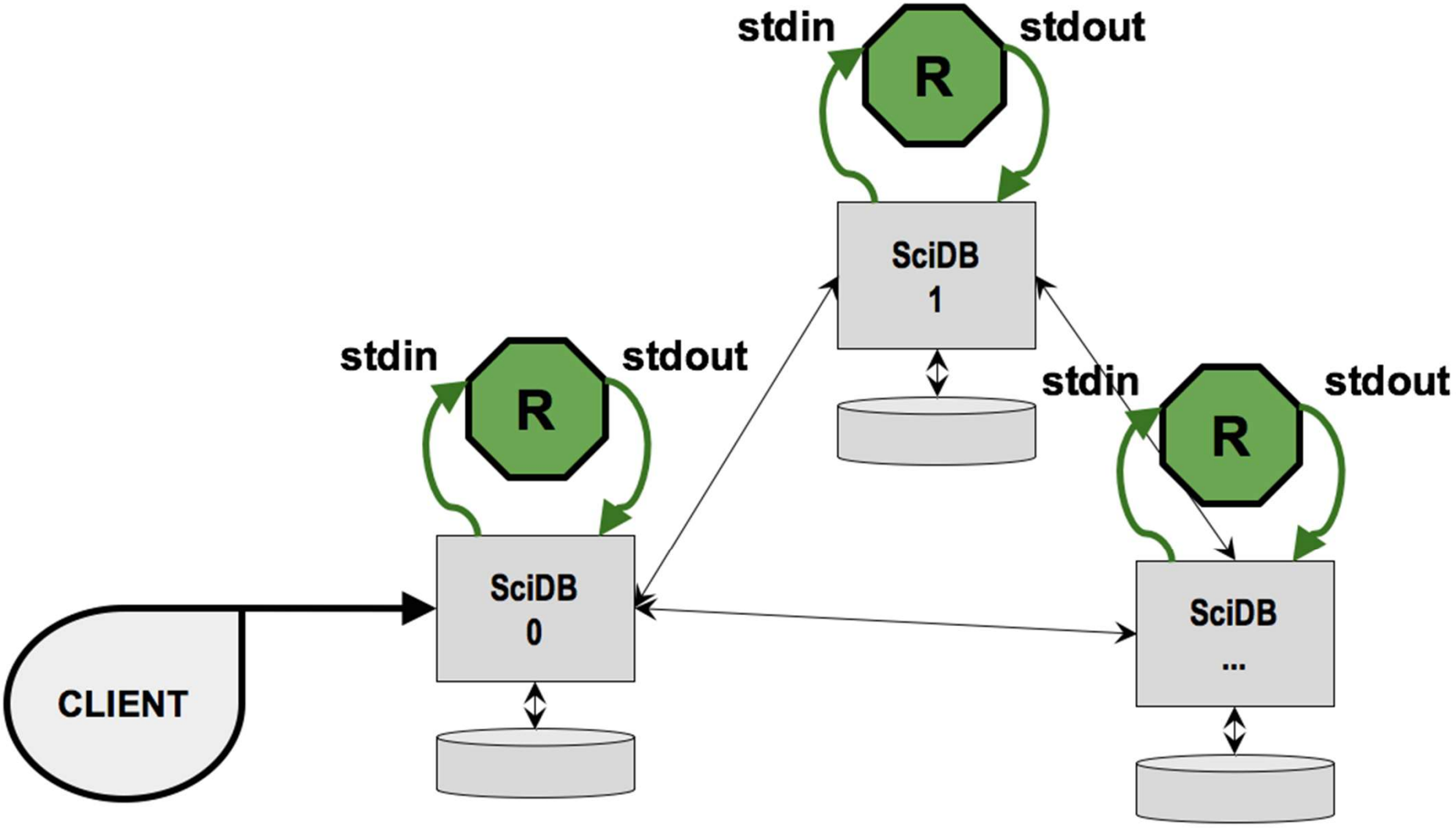
SciDB: Stream

- Plugin Stream
- Abordagem equivalente Hadoop
- Processamento junto aos dados
- Processamento por *chunk*

```
stream(array,'/path/to/app');
```



Fonte: Paradigm4 (2018)

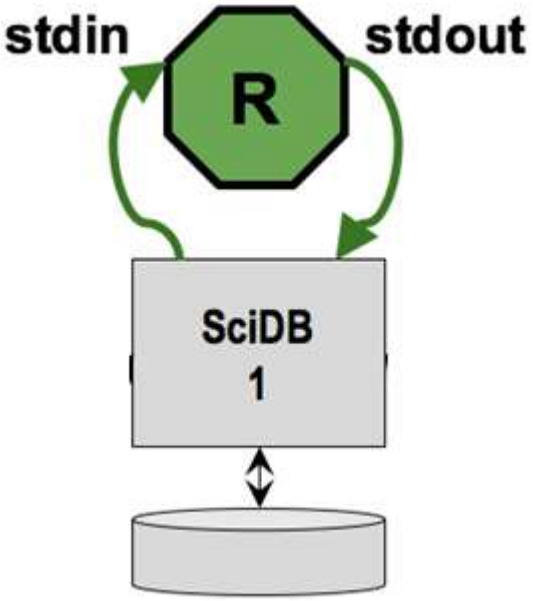


SciDB – Stream

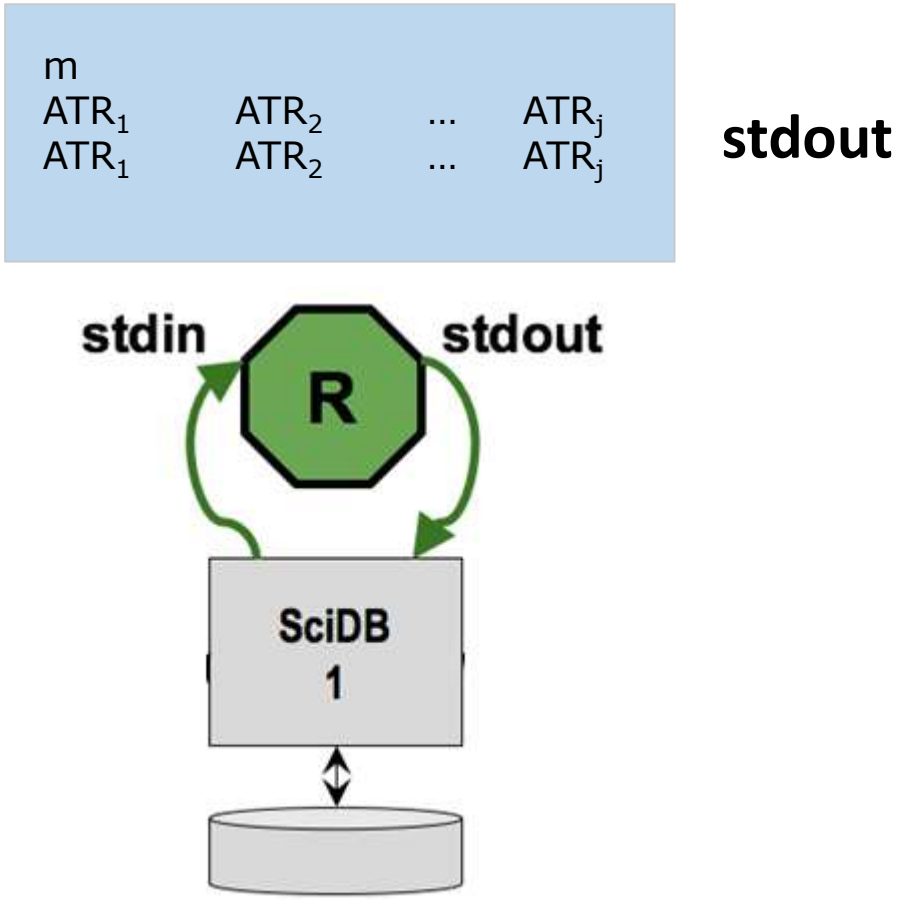
SciDB -> Application

stdin

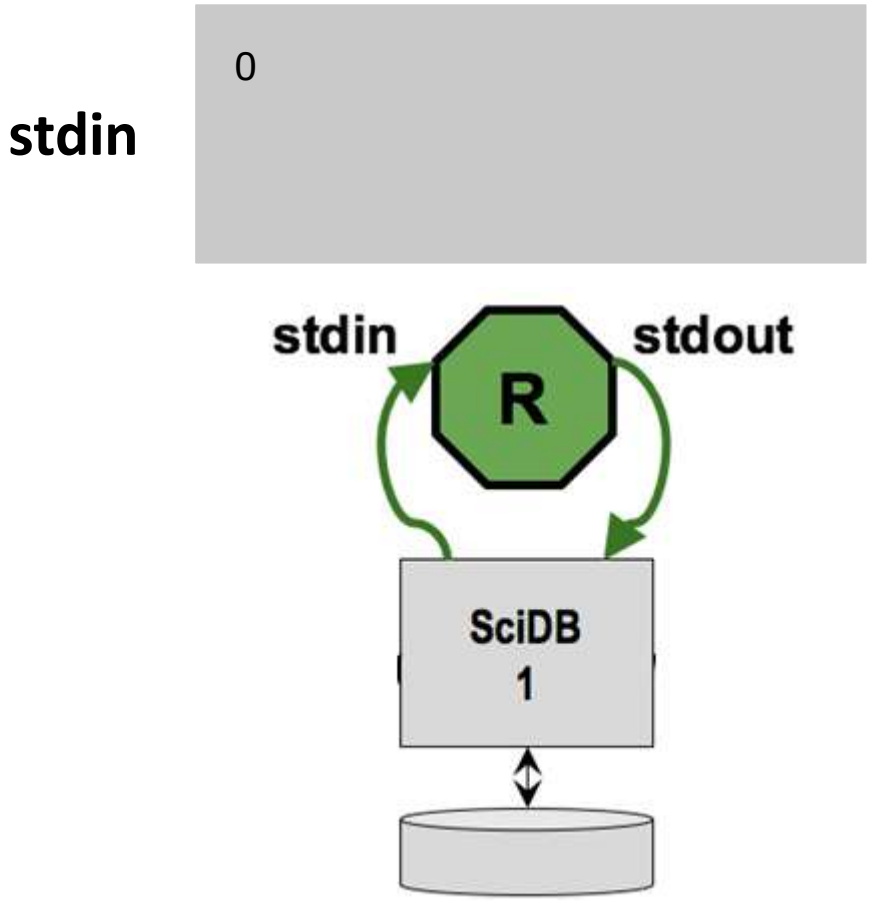
n
ATR_1 ATR_2 ... ATR_k
ATR_1 ATR_2 ... ATR_k
ATR_1 ATR_2 ... ATR_k



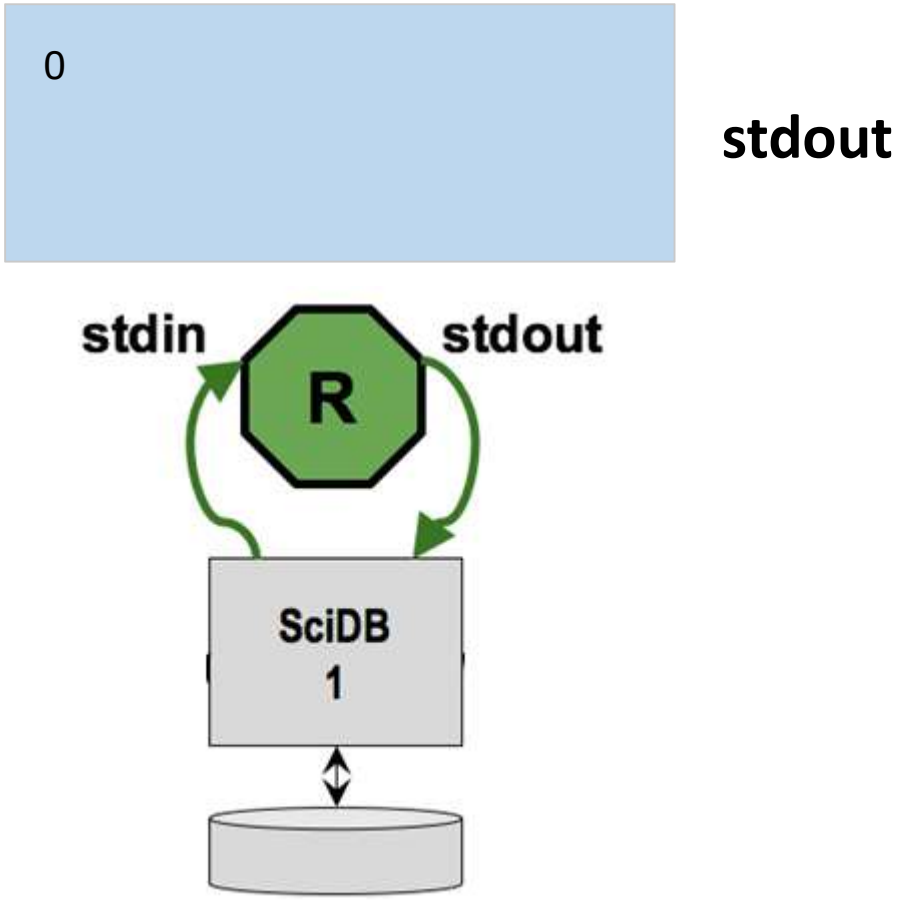
Application -> SciDB



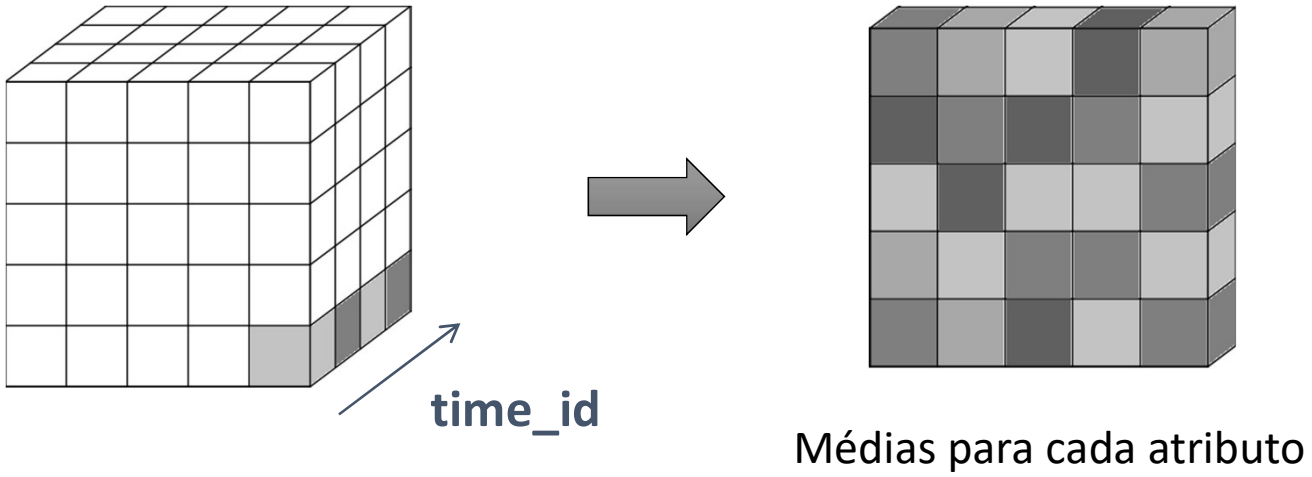
SciDB: Finished



Application: Finished



Computing the Average of Time Series

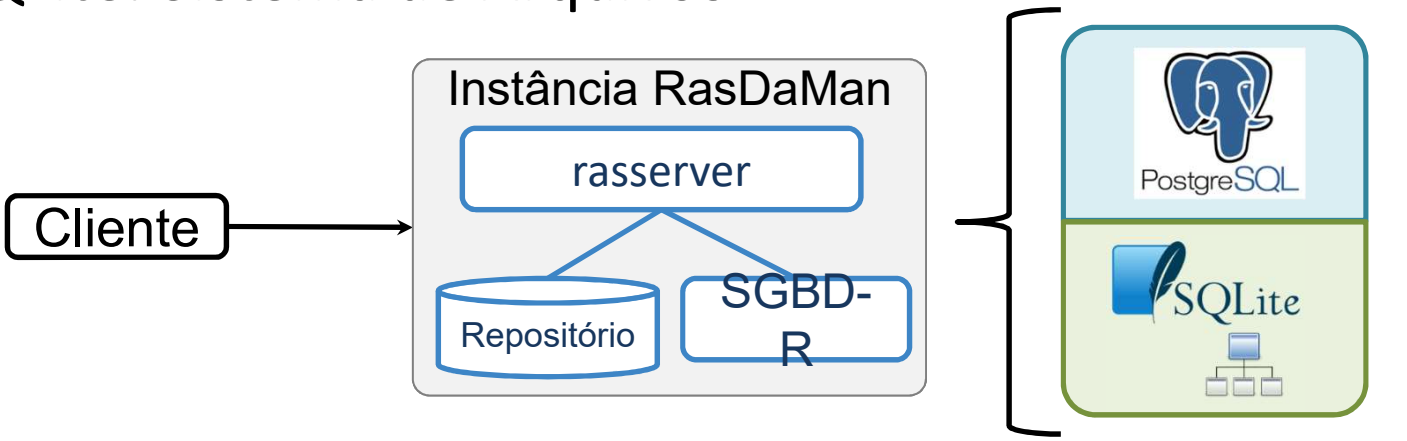



```
stream(mod13q1, '/GeoData/aqua/scidb/avg_time.py')
```

Considerações Finais

RasDaMan

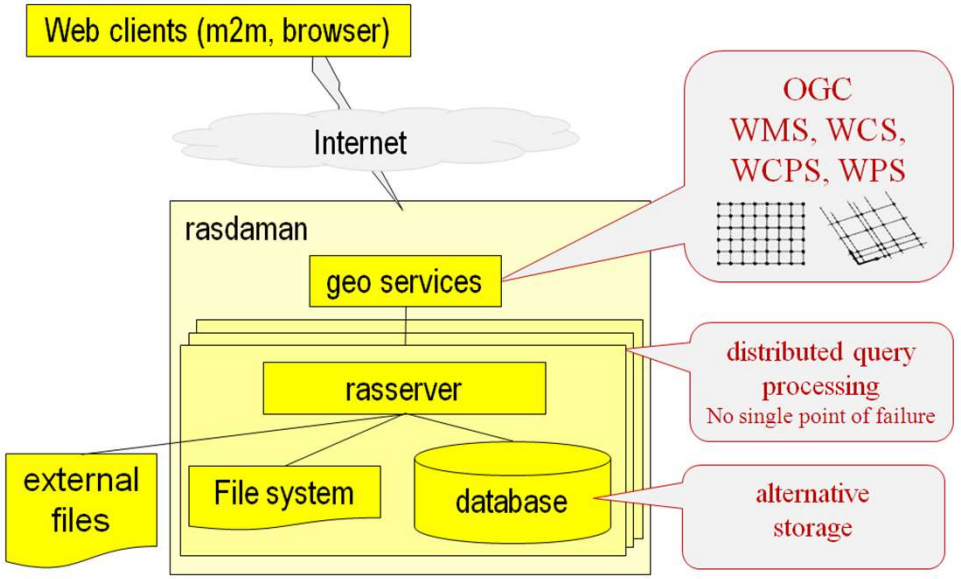
- **Ra**ster **Da**ta **Man**ager
- Clientes: R, Python, Java, C++
- PostgreSQL: campos *blob*
- SQLite: Sistema de Arquivos



Cluster de computadores somente na versão Enterprise

RasDaMan

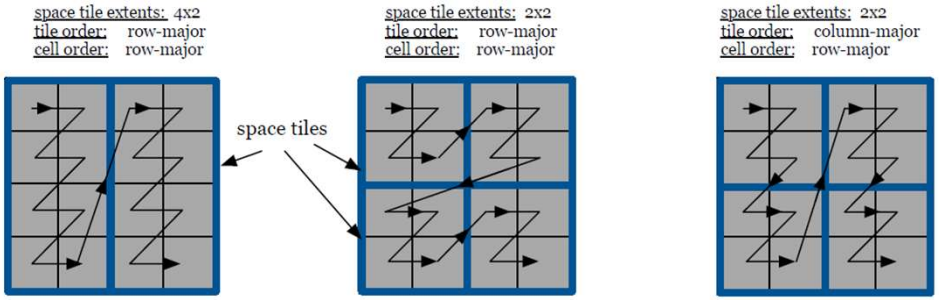
- Projeto EarthServer
- Acesso aos dados através de WMS, WCS, WCPS, WPS

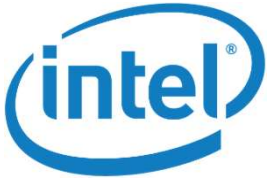


Fonte: RasDaMan (2018)

TileDB

- Biblioteca em C
- Sem distribuição de dados
- Pasta representa o *Array* no Sist. Arq.
- Baixa abstração para acesso aos dados

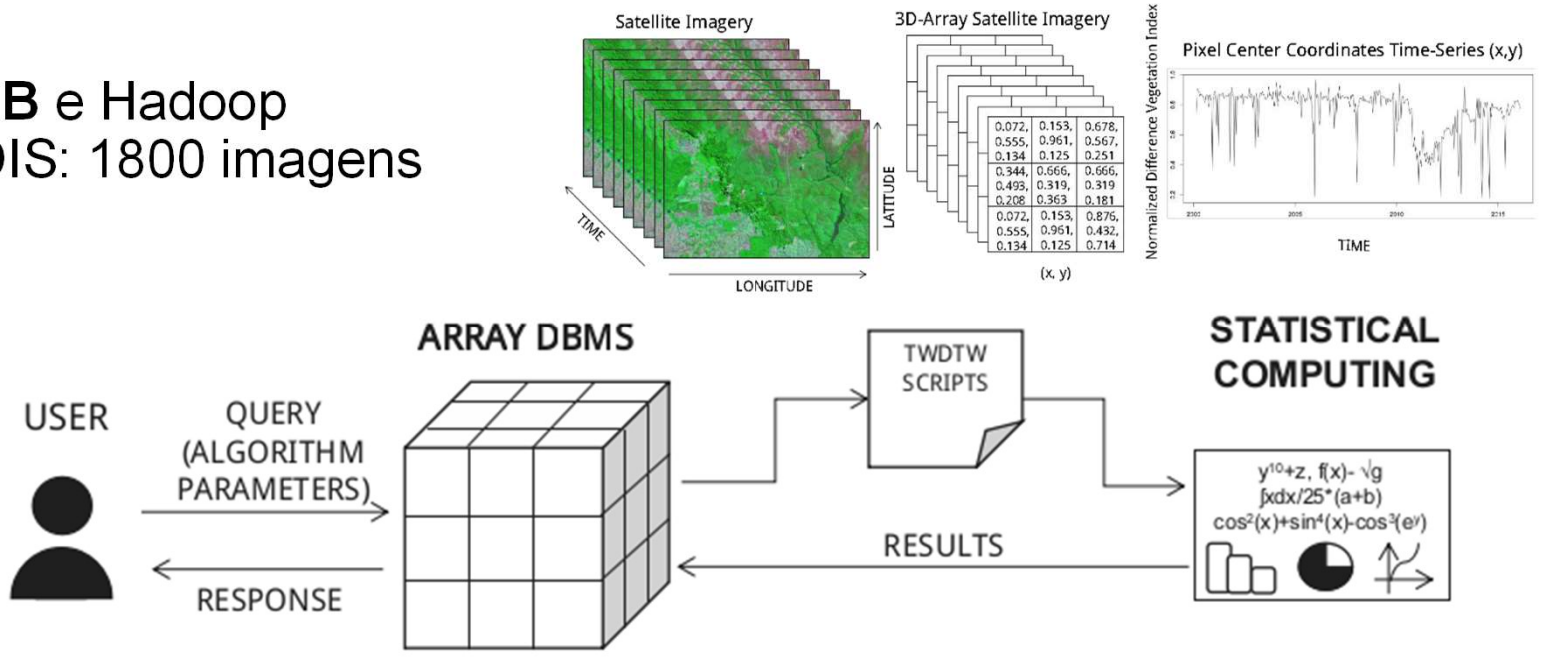




Architecture for Big EO Data Analytics

Avaliação da arquitetura: análise de série temporal

- **SciDB** e Hadoop
- MODIS: 1800 imagens



Fonte: Câmara et al. (2016)

Considerações Finais

- Nesta aula apresentamos uma classe de sistemas de bancos de dados baseado em um modelo orientado a arrays.
- Álgebra + Linguagem de Consulta matricial.
- Particionamento de Dados e Replicação.
- Escalabilidade.

Referências Bibliográficas

Referências Bibliográficas

- Stonebraker, M.; Brown, P.; Poliakov, A.; Raman, S. **The Architecture of SciDB**. Proceedings of the 23rd International Conference on Scientific and Statistical Database Management, SSDBM'11, 2011.

- Paradigm4. SciDB Documentation. Acesso: Agosto de 2016.

Exercícios