SciDB:
Massively Parallel Array Data Storage, Processing and Analysis

Paul G. Brown
Sequencing Machine Identifier
1. Flow Cell Lane
2. Tile Number
3. X Index on Tile
4. Y Index on Tile

Something complex and biological.

@PHUSCA-W21172_91027:1:1:0:1521#0/1
NAGCCCTGCCCCCTTCTGAGAGTCCCTTGTTAAGCAA
+PHUSCA-W21172_91027:1:1:0:1521#0/1
DNWWWWWVVUWWWWWVVVUWUWWUWWWWVUTSTSB
@PHUSCA-W21172_91027:1:1:0:1784#0/1
NACCAGGACTATTTGCACTCTTTGGGGAAGGCTCGA
+PHUSCA-W21172_91027:1:1:0:1784#0/1
DOWYUVVUVYUYYWWWYYWWWUTVYRPWUWYWWT

X 85 Million
X 10 per day
'Short Reads' have fixed length, 50 offsets in this case.

A - Adenine  
G - Guanine  
T – Thymine (Uracil)  
C – Cytosine  
N - “ERROR”

Quality control—how many 'N' values appear, and where—is an important process management question.
Quality Control Metrics

- **Sequences with a “N”**
  - Y-axis: 0 to 60000
  - X-axis: 2 to 50

- **Position of the “N”**
  - Y-axis: 0 to 500000
  - X-axis: 1 to 49
### Why another DBMS?

```
CREATE TABLE Short_Reads (  
  Machine_ID    CHAR(25) NOT NULL,  
  Flow_Cell_Lane INTEGER  NOT NULL,  
  Tile_Number   INTEGER  NOT NULL,  
  X             INTEGER  NOT NULL,  
  Y             INTEGER  NOT NULL,  
  Offset        INTEGER  NOT NULL,  
  Base_Pair     CHAR(1)  NOT NULL,  
  Supp          FLOAT
    PRIMARY KEY ( Machine_ID,  
                  Flow_Cell_Lane,  
                  Tile_Number, X, Y, Offset )
);
```

- Explicit storage of key information multiplies data storage requirements.
- Order of key (or index) specification biases performance.
- Many problems where 'previous' and 'next' have semantic heft.
Introducing SciDB

CREATE ARRAY Reads
  ( Read::CHAR(1), Supp::FLOAT )
  [ Seq_ID:0,*, Offset:0,50 ];

( AGGREGATE ( COUNT(), R.Offset, FILTER ( R.Read = 'N', READS AS R ) ) );
Architecture Overview 1

1. Vertical Partitioning into multiple single-attribute arrays.

<table>
<thead>
<tr>
<th>G 0.5</th>
<th>C 0.3</th>
<th>A 0.1</th>
<th>A' 0.9</th>
<th>C 0.2</th>
<th>A 0.9</th>
<th>T 0.1</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>T 0.2</th>
<th>T 0.1</th>
<th>A 0.6</th>
<th>A 0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>C 0.2</td>
<td>T 0.9</td>
<td>T 0.7</td>
<td>A 0.8</td>
<td>C 0.5</td>
<td>N?</td>
<td>G 0.5</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>C 0.8</td>
<td>C 0.3</td>
<td>A 0.5</td>
<td>T 0.2</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>T 0.3</td>
<td>A 0.6</td>
<td>C 0.4</td>
<td>N?</td>
<td>G 0.8</td>
<td>C 0.5</td>
<td>C 0.7</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>G 0.3</td>
<td>C 0.9</td>
<td>A 0.7</td>
<td>A 0.9</td>
</tr>
</tbody>
</table>

| G C A A C A T . . . . T T A A |
| C T T A C N G . . . . C C A T |
| . . . . . . . . . . . . . . . . |
| T A C N G C C . . . . G C A A |

<table>
<thead>
<tr>
<th>0.5</th>
<th>0.3</th>
<th>0.1</th>
<th>0.9</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>0.6</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.9</td>
<td>0.7</td>
<td>0.8</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>0.5</td>
<td>0.2</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>0.3</td>
<td>0.6</td>
<td>0.4</td>
<td>?</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>0.7</td>
<td>0.9</td>
</tr>
</tbody>
</table>
1. Divide the single-value array into overlapping partitions, or chunks.
3. Assign *chunks* to physical nodes in a massively parallel architecture.
Architecture 4: Query Processing

- Query expressions parsed into physical plans.
  - Minimal heuristics for coarse plan.
  - Run time re-organization of plan based on progress.

- Parallelizable segments of plans distributed.
  - Governing query manager coordinates segments.

- Local segment execution.
  - 'Chunk at a time' processing.
  - Extensible type system and Array-Value Operators.
Project Status

• **Pick-up team of developers.**
  - Science project developers.
  - Experienced DBMS server types.
  - SciDB is an Open-Source (GPL-3) project.

• **Crash-or-Crash-Through development.**
  - Demonstration system at VLDB-09.
  - Currently developing V1, due Q2 2010.

• **Help!?**
  - Fun research questions to answer.
  - Lots of work to do.
  - http://www.scidb.org/