Centrum voor Wiskunde en Informatica

01/06/04

RAM

A multi dimensional array DBMS

Alex.van.Ballegooij@cwi.nl

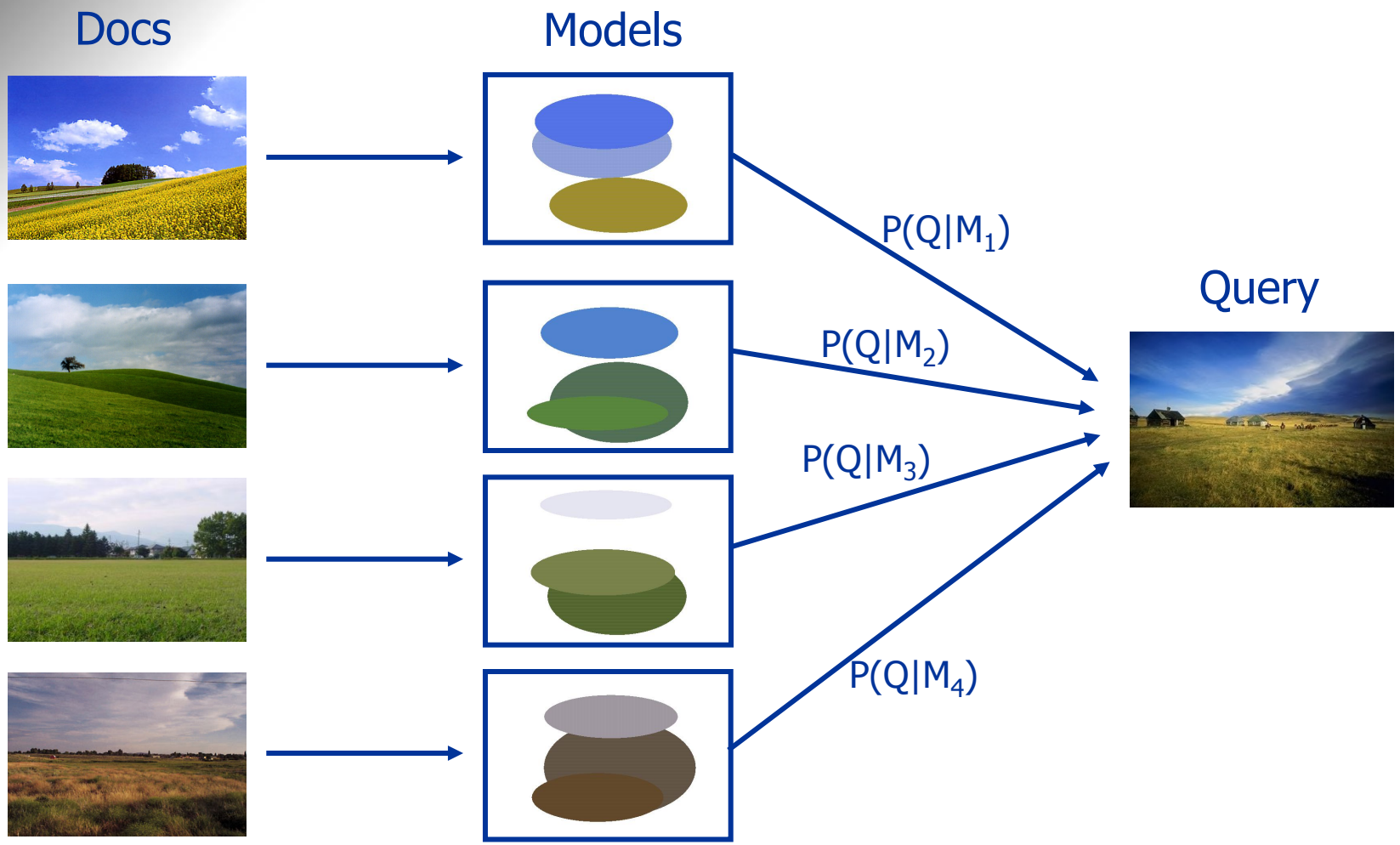
RAM Overview (2)

- RAM offers an array query language
  - Declarative, comprehension based, language

$$[\ f(A(x),x,c)\ |\ x < N1,\ y<N2,\ldots\ ]$$

    - Defines the **shape** of the array
    - Defines the **values** in terms of constants, functions and existing arrays

Example: Image retrieval with GMM



Example: GMM ranking with RAM

$$\forall M : P(Q \mid M) = \sum_{s=1}^{N_s} \log(P(Q_s \mid M))$$

```
Scores = [sum( [ log( p(s,m) ) | s<Ns ] ) | m<Nm ]
```

$$P(Q_s \mid M) = \sum_{c=1}^{N_c} P(C_c, M) \frac{1}{\sqrt{(2\pi)^{N_n} \prod_{n=1}^{N_n} \sigma_n^2}} e^{-\frac{1}{2}\sum_{n=1}^{N_n} \frac{(Q_{s,n}-\mu_n)^2}{\sigma_n^2}}$$

```
p(s,m) = sum([
         Pr(c,m) *
         (1/(sqrt(pow(2*PI,Nn))*prod([S2(n,c,m)|n<Nn]))) *
         exp(-0.5 *
         sum([pow(Q(n,s)-Mu(n,c,m),2)/S2(n,c,m)|n<Nn]))
                  | c<Nc ])
```

RAM Overview

Abstract query

$$\sum_{c=1}^{N_c} \frac{P(C_c, M)}{\sqrt{(2\pi)^{N_n} \prod_{n=1}^{N_n} \sigma_n^2}} e^{-\frac{1}{2}\sum_{n=1}^{N_n} \frac{(Q_{s,n}}{\sigma}}$$

RAM query

```
P(s,m) = sum([Pr(c,m) *
(1/(sqrt(pow(2*PI,Nn))*prod
([S2(n,c,m)|n<Nn])))
```

RAM Algebra

```
map(dbl,[const([8,1320,3500
0],"14")])]),apply(aggreg
ate(prod,apply(apply(#([350
```

Optimised RAM Algebra

```
map(dbl,[const([8,1320,3500
0],"14")])]),apply(aggreg
ate(prod,apply(apply(#([350
```

MIL query

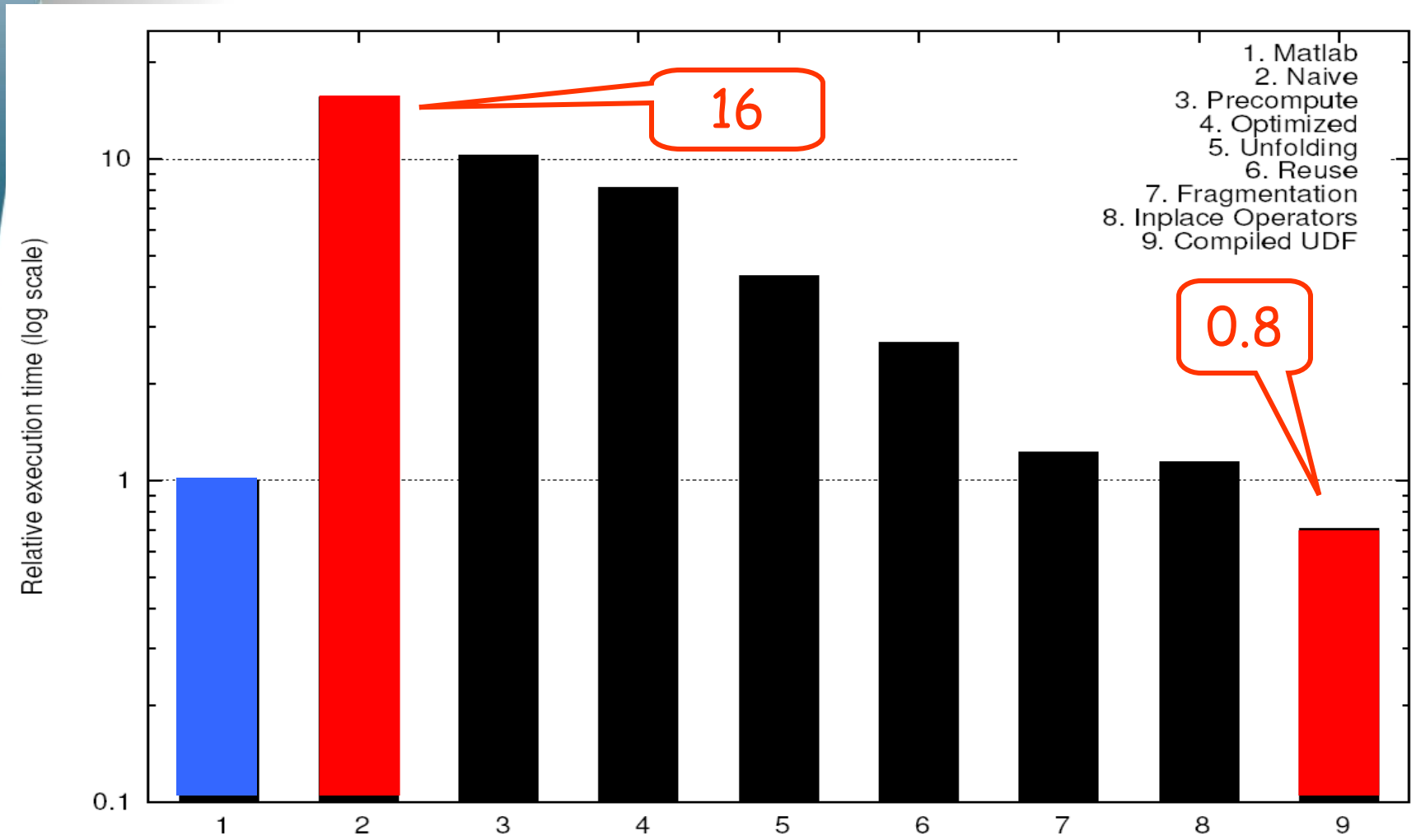
```
t26 := {prod}(t27,t51,t73);
t85 := join(t86,bat_Q);
t24 := nil;
```

SQL query

```
A0.i3 = A1.i3 ) AS A GROUP
BY i0 ) AS A1 WHERE A0.i0 =
A1.i0 AND A0.i1 = A1.i1 AND
```

RAM performance

- Experiment with GMM image ranking:



RAM Conclusions

- Powerful query language
  - seems to suit scientific data processing
- Opportunity for Optimizations
  - exploit array specific properties in data mapping and query translation
- Performance
  - on par with Matlab
  - may be improved further (X100?)