class 7

complex plans and hybrid layouts

prof. Stratos Idreos

HTTP://DASLAB.SEAS.HARVARD.EDU/CLASSES/CS165/
column-storage

| A | B | C | D |

row-storage

| A | B | C | D |

two extremes
**PAX**: store all data about a row in a single page but organize data in a column major way inside each page
column-groups: store data about a row in a single page but use any kind of column-group combination inside each page
column-groups on separate files: group columns in separate files - each row is spread in >1 pages
how do we decide?
say this is the disk storage

first think what do we want to do with the data: access patterns
say this is the disk storage

first think what do we want to do with the data: access patterns

query A, B,
query A, B, C, D
insert
delete
update A
say this is the disk storage

first think what do we want to do with the data: access patterns

query A,B, query A,B,C,D
insert delete update A

metric: data movement
offline
online
code generation
can we do any better?

fixed-width
dense
unordered
columns
zone1:
min value = m1
max value = k1

zone2:
min value = m2
max value = k2
zone1:
min value = m1
max value = k1

zone2:
min value = m2
max value = k2

what if data is uniformly distributed over the column
zone1: 
min value = m1 
max value = k1

zone2: 
min value = m2 
max value = k2

what if data is uniformly distributed over the column

adaptively reorganize column
to minimize # of zones a query has to scan
cs165/265 project: finalist ACM SIGMOD undergrad research competition 2016
it is all about the bits

```
column
value1= 1 0 1 0 1 0 0 1
value2= 0 0 1 1 0 0 0 0
value3= 0 0 0 0 0 1 1 0
```

Jignesh Patel, U of Wisconsin
**BitWeaving: fast scans for main memory data processing**
Yinan Li, Jignesh M. Patel

**ByteSlice: Processing with a New Storage Layout**
Ziqiang Feng, Eric Lo, Ben Kao, Wenjian Xu
give me all students enrolled in cs165

```sql
select student.name from student, enrolled, course where course.name="cs165" and enrolled.courseld=course.id and student.id=enrolled.studentId
```
Query and Query Plan (MAL Algebra)

select sum(R.a) from R, S where R.c = S.b and 5<R.a<20 and 40<R.b<50 and 49<S.a<65

1. inter1 = select(Ra,5,20)
2. inter2 = reconstruct(Rb,inter1)
3. inter3 = select(inter2,40,50)
4. join_input_R = reconstruct(Rc,inter3)
5. inter4 = select(Sa,49,65)
6. inter5 = reconstruct(Sb,inter4)
7. join_input_S = reverse(inter5)
8. join_res_R_S = join(join_input_R,join_input_S)
9. inter6 = voidTail(join_res_R_S)
10. inter7 = reconstruct(Ra,inter6)
11. result = sum(inter7)
### Initial Status

<table>
<thead>
<tr>
<th>Relation R</th>
<th>Relation S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ra</td>
<td>Rb</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>16</td>
<td>34</td>
</tr>
<tr>
<td>56</td>
<td>75</td>
</tr>
<tr>
<td>9</td>
<td>45</td>
</tr>
<tr>
<td>11</td>
<td>49</td>
</tr>
<tr>
<td>27</td>
<td>58</td>
</tr>
<tr>
<td>8</td>
<td>97</td>
</tr>
<tr>
<td>41</td>
<td>75</td>
</tr>
<tr>
<td>19</td>
<td>42</td>
</tr>
<tr>
<td>35</td>
<td>55</td>
</tr>
</tbody>
</table>

### Query and Query Plan (MAL Algebra)

```sql
select sum(R.a) from R, S where R.c = S.b and
5 < R.a < 20 and 40 < R.b < 50 and 30 < S.a < 40
```

1. inter1 = select(Ra, 5, 20)
2. inter2 = reconstruct(Rb, inter1)
3. inter3 = select(inter2, 40, 50)
4. join_input_R = reconstruct(Rc, inter3)
5. inter4 = select(Sa, 49, 65)
6. inter5 = reconstruct(Sb, inter4)
7. join_input_S = reverse(inter5)
8. join_res_R_S = join(join_input_R, join_input_S)
9. inter6 = voidTail(join_res_R_S)
10. inter7 = reconstruct(Ra, inter6)
11. result = sum(inter7)
### Initial Status

<table>
<thead>
<tr>
<th>Relation R</th>
<th>Relation S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ra</td>
<td>Rb</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>16</td>
<td>34</td>
</tr>
<tr>
<td>56</td>
<td>75</td>
</tr>
<tr>
<td>9</td>
<td>45</td>
</tr>
<tr>
<td>11</td>
<td>49</td>
</tr>
<tr>
<td>27</td>
<td>58</td>
</tr>
<tr>
<td>8</td>
<td>97</td>
</tr>
<tr>
<td>41</td>
<td>75</td>
</tr>
<tr>
<td>19</td>
<td>42</td>
</tr>
<tr>
<td>35</td>
<td>55</td>
</tr>
</tbody>
</table>

### Query and Query Plan (MAL Algebra)

```sql
select sum(R.a) from R, S where R.c = S.b and 5 < R.a < 20 and 40 < R.b < 50 and 49 < S.a < 65
```

1. inter1 = `select(Ra, 5, 20)`
2. inter2 = `reconstruct(Rb, inter1)`
3. inter3 = `select(inter2, 40, 50)`
4. `join_input_R = reconstruct(Rc, inter3)`
5. inter4 = `select(Sa, 49, 65)`
6. inter5 = `reconstruct(Sb, inter4)`
7. `join_input_S = reverse(inter5)`
8. `join_res_R_S = join(join_input_R, join_input_S)`
9. inter6 = `voidTail(join_res_R_S)`
10. inter7 = `reconstruct(Ra, inter6)`
11. result = `sum(inter7)`
Query and Query Plan (MAL Algebra)

select sum(R.a) from R, S where R.c = S.b and 5<R.a<20 and 40<R.b<50 and 30<S.a<40

1. inter1 = select(Ra,5,20)
2. inter2 = reconstruct(Rb,inter1)
3. inter3 = select(inter2,40,50)
4. join_input_R = reconstruct(Rc,inter3)
5. inter4 = select(Sa,49,65)
6. inter5 = reconstruct(Sb,inter4)
7. join_input_S = reverse(inter5)
8. join_res_R_S = join(join_input_R,join_input_S)
9. inter6 = voidTail(join_res_R_S)
10. inter7 = reconstruct(Ra,inter6)
11. result = sum(inter7)
<table>
<thead>
<tr>
<th>Ra</th>
<th>Rb</th>
<th>Rc</th>
<th>Sa</th>
<th>Sb</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>12</td>
<td>12</td>
<td>17</td>
<td>11</td>
</tr>
<tr>
<td>16</td>
<td>34</td>
<td>34</td>
<td>49</td>
<td>35</td>
</tr>
<tr>
<td>56</td>
<td>75</td>
<td>53</td>
<td>58</td>
<td>62</td>
</tr>
<tr>
<td>9</td>
<td>45</td>
<td>23</td>
<td>99</td>
<td>44</td>
</tr>
<tr>
<td>11</td>
<td>49</td>
<td>78</td>
<td>64</td>
<td>29</td>
</tr>
<tr>
<td>27</td>
<td>58</td>
<td>65</td>
<td>37</td>
<td>78</td>
</tr>
<tr>
<td>8</td>
<td>97</td>
<td>33</td>
<td>53</td>
<td>19</td>
</tr>
<tr>
<td>41</td>
<td>75</td>
<td>21</td>
<td>61</td>
<td>81</td>
</tr>
<tr>
<td>19</td>
<td>42</td>
<td>29</td>
<td>32</td>
<td>26</td>
</tr>
<tr>
<td>35</td>
<td>55</td>
<td>0</td>
<td>50</td>
<td>23</td>
</tr>
</tbody>
</table>

Query and Query Plan (MAL Algebra)

select sum(R.a) from R, S where R.c = S.b and
5 < R.a < 20 and 40 < R.b < 50 and 49 < S.a < 65

1. inter1 = select(Ra,5,20)
2. inter2 = reconstruct(Rb,inter1)
3. inter3 = select(inter2,40,50)
4. join_input_R = reconstruct(Rc,inter3)
5. inter4 = select(Sa,49,65)
6. inter5 = reconstruct(Sb,inter4)
7. join_input_S = reverse(inter5)
8. join_res_R_S = join(join_input_R,join_input_S)
9. inter6 = voidTail(join_res_R_S)
10. inter7 = reconstruct(Ra,inter6)
11. result = sum(inter7)
Query and Query Plan (MAL Algebra)

select sum(R.a) from R, S where R.c = S.b and 5<R.a<20 and 40<R.b<50 and 49<S.a<65

1. inter1 = select(Ra,5,20)
2. inter2 = reconstruct(Rb,inter1)
3. inter3 = select(inter2,40,50)
4. join_input_R = reconstruct(Rc,inter3)
5. inter4 = select(Sa,49,65)
6. inter5 = reconstruct(Sb,inter4)
7. join_input_S = reverse(inter5)
8. join_res_R_S = join(join_input_R,join_input_S)
9. inter6 = voidTail(join_res_R_S)
10. inter7 = reconstruct(Ra,inter6)
11. result = sum(inter7)
### Initial Status

<table>
<thead>
<tr>
<th>Relation R</th>
<th>Relation S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ra</td>
<td>Rb</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>16</td>
<td>34</td>
</tr>
<tr>
<td>56</td>
<td>75</td>
</tr>
<tr>
<td>9</td>
<td>45</td>
</tr>
<tr>
<td>11</td>
<td>49</td>
</tr>
<tr>
<td>27</td>
<td>58</td>
</tr>
<tr>
<td>8</td>
<td>97</td>
</tr>
<tr>
<td>41</td>
<td>75</td>
</tr>
<tr>
<td>19</td>
<td>42</td>
</tr>
<tr>
<td>35</td>
<td>55</td>
</tr>
</tbody>
</table>

### Query and Query Plan (MAL Algebra)

select sum(R.a) from R, S where R.c = S.b and 5 < R.a < 20 and 40 < R.b < 50 and 49 < S.a < 65

1. inter1 = select(Ra,5,20)
2. inter2 = reconstruct(Rb,inter1)
3. inter3 = select(inter2,40,50)
4. join_input_R = reconstruct(Rc,inter3)
5. inter4 = select(Sa,49,65)
6. inter5 = reconstruct(Sb,inter4)
7. join_input_S = reverse(inter5)
8. join_res_R_S = join(join_input_R,join_input_S)
9. inter6 = voidTail(join_res_R_S)
10. inter7 = reconstruct(Ra,inter6)
11. result = sum(inter7)
**Query and Query Plan (MAL Algebra)**

Given the query:

```
select sum(R.a) from R, S where R.c = S.b and 5<R.a<20 and 40<R.b<50 and 30<S.a<40
```

1. **inter1 = select(Ra,5,20)**
2. **inter2 = reconstruct(Rb,inter1)**
3. **inter3 = select(inter2,40,50)**
4. **join_input_R = reconstruct(Rc,inter3)**
5. **inter4 = select(Sa,49,65)**
6. **inter5 = reconstruct(Sb,inter4)**
7. **join_input_S = reverse(inter5)**
8. **join_res_R_S = join(join_input_R,join_input_S)**
9. **inter6 = voidTail(join_res_R_S)**
10. **inter7 = reconstruct(Ra,inter6)**
11. **result = sum(inter7)**

---

### Initial Status

<table>
<thead>
<tr>
<th>Relation R</th>
<th>Relation S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ra</td>
<td>Rb</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>16</td>
<td>34</td>
</tr>
<tr>
<td>56</td>
<td>75</td>
</tr>
<tr>
<td>9</td>
<td>45</td>
</tr>
<tr>
<td>11</td>
<td>49</td>
</tr>
<tr>
<td>27</td>
<td>58</td>
</tr>
<tr>
<td>8</td>
<td>97</td>
</tr>
<tr>
<td>41</td>
<td>75</td>
</tr>
<tr>
<td>19</td>
<td>42</td>
</tr>
<tr>
<td>35</td>
<td>55</td>
</tr>
</tbody>
</table>

---

### Reconstruct (Ra,inter6)

<table>
<thead>
<tr>
<th>inter6</th>
<th>Ra</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>35</td>
</tr>
</tbody>
</table>

**Result:** 28

---

### Sum (inter7)

<table>
<thead>
<tr>
<th>inter7</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 19</td>
<td>28</td>
</tr>
</tbody>
</table>
select \( \text{max}(R.D), \text{min}(S.G) \)
from R,S
where R.A=S.A and R.C<10 and S.F>30

what happens after the join?

access patterns

block operator
select R.A, R.B, R.C, S.A, S.B, S.C
from R, S
where R.J=S.J and ...

preparing the R join input

we need the original positions so we can fetch other R columns after the join

same for the S join input
\textbf{select} R.A, R.B, R.C, S.A, S.B, S.C
\textbf{from} R, S
\textbf{where} R.J = S.J and …
\begin{verbatim}
select R.A, R.B, R.C, S.A, S.B, S.C
from R, S
where R.J = S.J and ...
\end{verbatim}
first part done: basic concepts in modern systems

coming up: indexing and fast scans
Cache-Conscious Radix Decluster Projections
By S. Manegold, P. Boncz, N. Nes, and M. Kersten
Very Large Databases Conference, 2004

H2O: A Hands-free Adaptive Store
Ioannis Alagiannis, Stratos Idreos, and Anastassia Ailamaki
complex plans & hybrid layouts

DATA SYSTEMS

prof. Stratos Idreos