LOAD

- Input is assumed to be a bag (sequence of tuples)
- Assumes that every dataset is a sequence of tuples
- Specify a parsing function with “USING”
- Specify a schema with “AS”

```
A = LOAD 'myfile.txt' USING PigStorage('\t') AS (f1,f2,f3);
```

<1, 2, 3>
<4, 2, 1>
<8, 3, 4>
<4, 3, 3>
<7, 2, 5>
<8, 4, 3>

FILTER: Getting rid of data

- Arbitrary boolean conditions
- Regular expressions allowed

  $0 matches apache

```
Y = FILTER A BY f1 == '8';
```

A =
<1, 2, 3>
<4, 2, 1>
<8, 3, 4>
<4, 3, 3>
<7, 2, 5>
<8, 4, 3>

Y =
<8, 3, 4>
<8, 4, 3>

GROUP: Getting data together

X = GROUP A BY f1;

A =
<1, 2, 3>
<4, 2, 1>
<8, 3, 4>
<4, 3, 3>
<7, 2, 5>
<8, 4, 3>

X =
<1, {<1, 2, 3>}>
<4, {<4, 2, 1>, <4, 3, 3>}>
<7, {<7, 2, 5>}>
<8, {<8, 3, 4>, <8, 4, 3>}>

- first field will be named “group”
- second field has name “A”

DISTINCT: Getting rid of duplicates

Y = DISTINCT A

A = <1, 2, 3>
<1, 2, 3>
<8, 3, 4>

Y = <1, 2, 3>
<8, 3, 4>

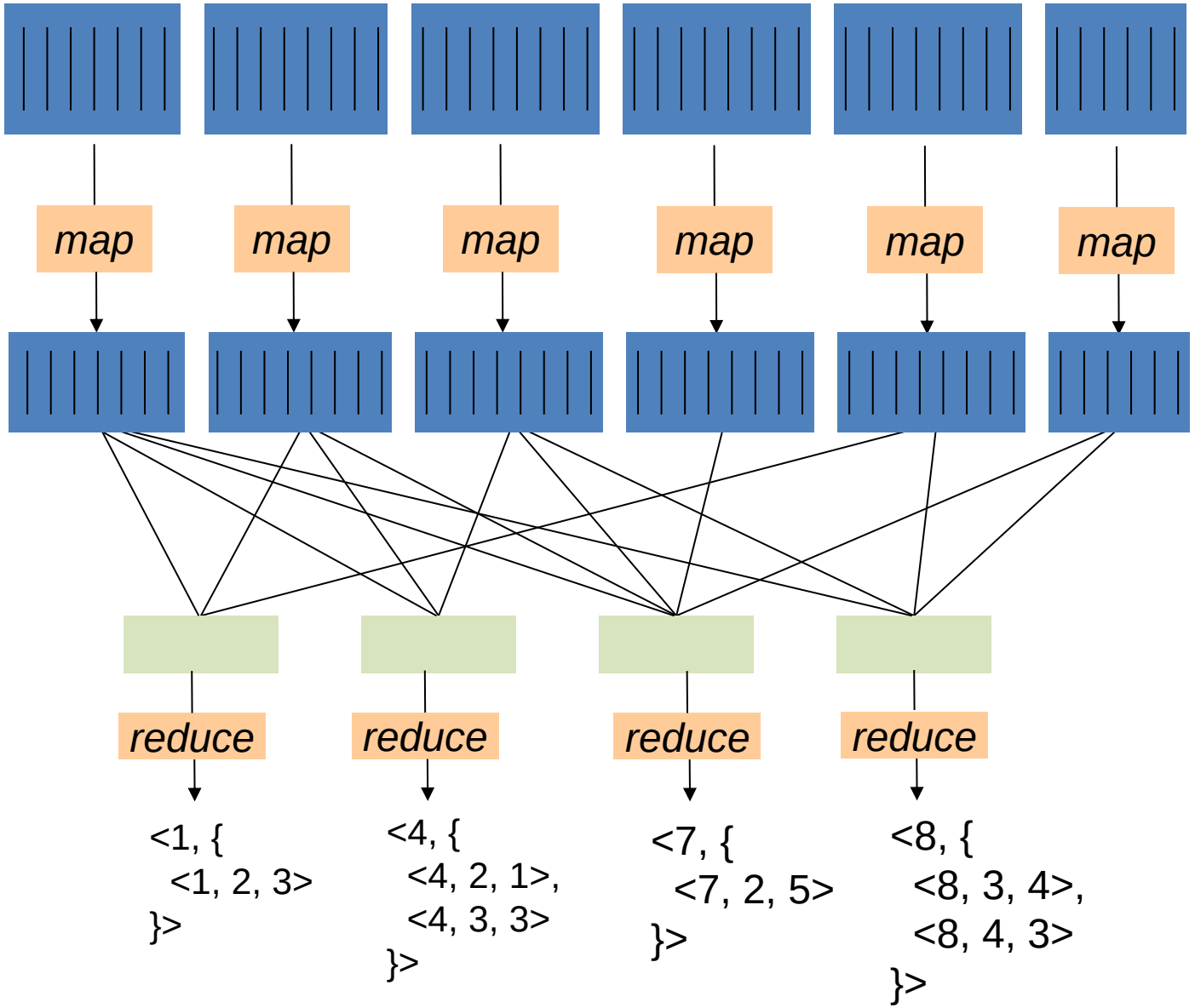
Claim:
DISTINCT A == GROUP A BY (f1, f2, f3)

Not quite:

< <1, 2, 3>, {<1,2,3>} >
< <8, 3, 4>, {<8,3,4>} >

GROUP

GROUP A BY f1;



key: f1
value: (f1,f2,f3)

FOREACH: Manipulate each tuple

```
X = FOREACH A GENERATE f0, f1+f2;
```

You can call UDFs

```
Y = GROUP A BY f1;
Z = FOREACH X GENERATE group, X.($1, $2);
```

You can manipulate nested objects

A =
<1, 2, 3>
<4, 2, 1>
<8, 3, 4>
<4, 3, 3>
<7, 2, 5>
<8, 4, 3>

X =
<1, 5>
<4, 3>
<8, 7>
<4, 6>
<7, 7>
<8, 7>

Z =
<1, {<2, 3>}>
<4, {<2, 1>, <3, 3>}>
<7, {<2, 5>}>
<8, {<3, 4>, <4, 3>}>

using the FLATTEN keyword

```
Y = GROUP A BY f1;
Z = FOREACH X GENERATE group, FLATTEN(X);
```

I don't like this, because FLATTEN has no well defined type. It's "magic"

A =
<1, 2, 3>
<4, 2, 1>
<8, 3, 4>
<4, 3, 3>
<7, 2, 5>
<8, 4, 3>

X =
<1, 5>
<4, 3>
<8, 7>
<4, 6>
<7, 7>
<8, 7>

Z =
<1, 2, 3>
<4, 2, 1>
<4, 3, 3>
<7, 2, 5>
<8, 3, 4>
<8, 4, 3>