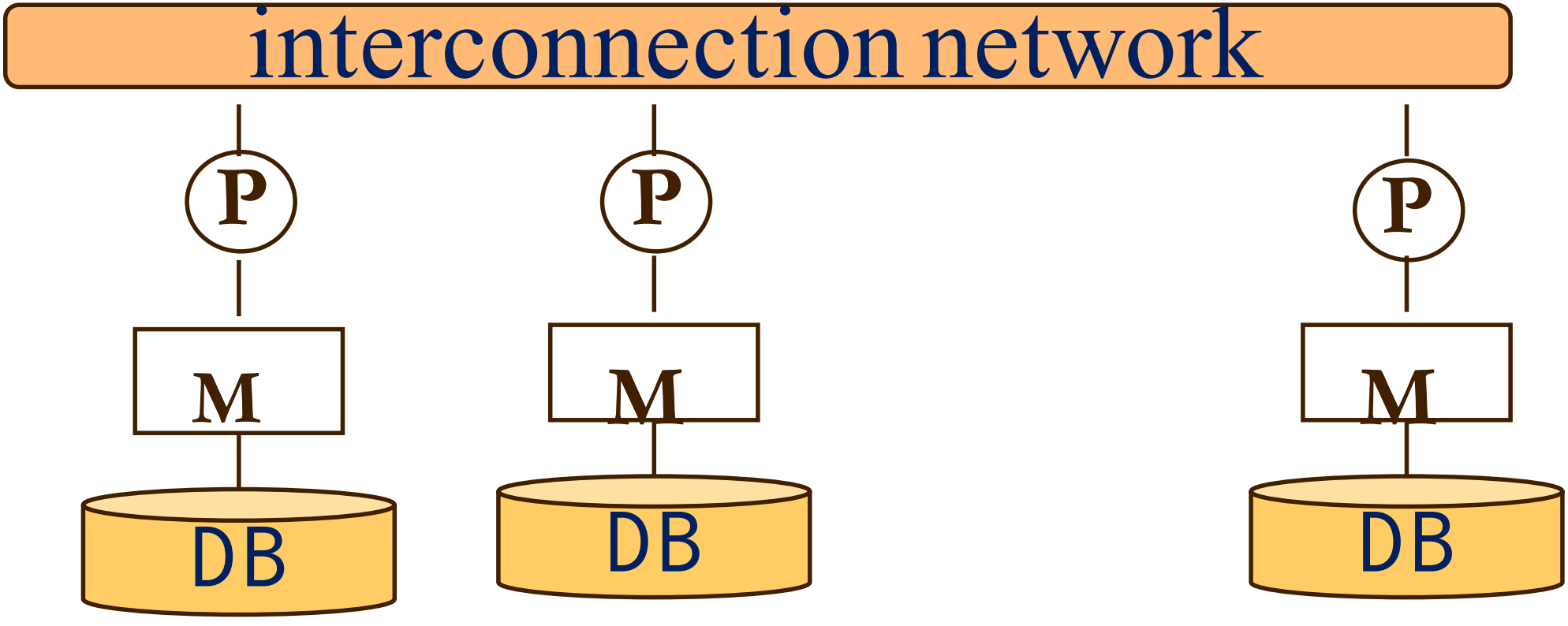
Common Wisdom

5

- **Parallel query processing**

Indexing


interconnection network
P
P
P
M
M
M
DB
DB
DB

Assuming *linear* scalability, using 50,000 processors

- 2,020 mins is reduced to 2.4s

the best we can hope for

1. a privilege of big companies (resource demanding)

2. typically using key-value systems for horizontal scalability
   - efficiency of SQL@KV is far from good *(much slower than DBMS)*
3. not every computation is parallel scalable
   - up to a point, adding more processors doesn't help

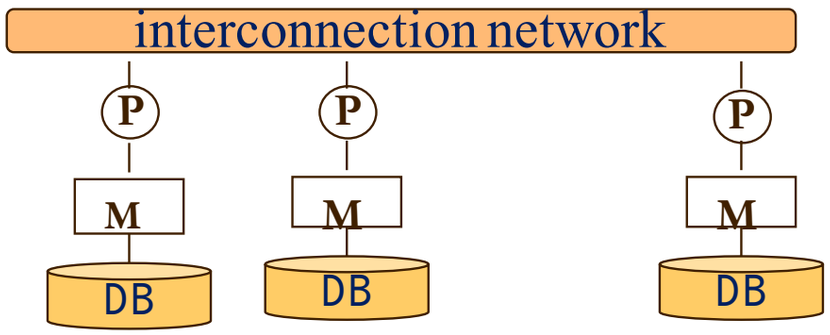
1. not always effective (and costly to maintain)
   - *e.g.*, index on `uid` or `country` of **`update`** doesn't help too much
2. cost fluctuates and is unknown before execution
3. not a first citizen of relational algebra (planning).

with constrained resources

We need a cost-bounded evaluation paradigm for small companies

Common Wisdom

- **Parallel query processing**



the best we can hope for

Assuming *linear* scalability, using 50,000 processors

- 2,020 mins is reduced to 2.4s

1. a privilege of big companies (resource demanding)
2. typically using key-value systems for horizontal scalability
   - efficiency of SQL@KV is far from good *(much slower than DBMS)*
3. not every computation is parallel scalable
   - up to a point, adding more processors doesn't help

- **Indexing**

1. not always effective (and costly to maintain)
   - *e.g.,* index on `uid` or `country` of **`update`** doesn't help too much

with constrained resources

We need a cost-bounded evaluation paradigm for small companies

3. not a first citizen of relational algebra (planning).

Bounded Query Processing