Wave Computing in the Cloud

Bingsheng He

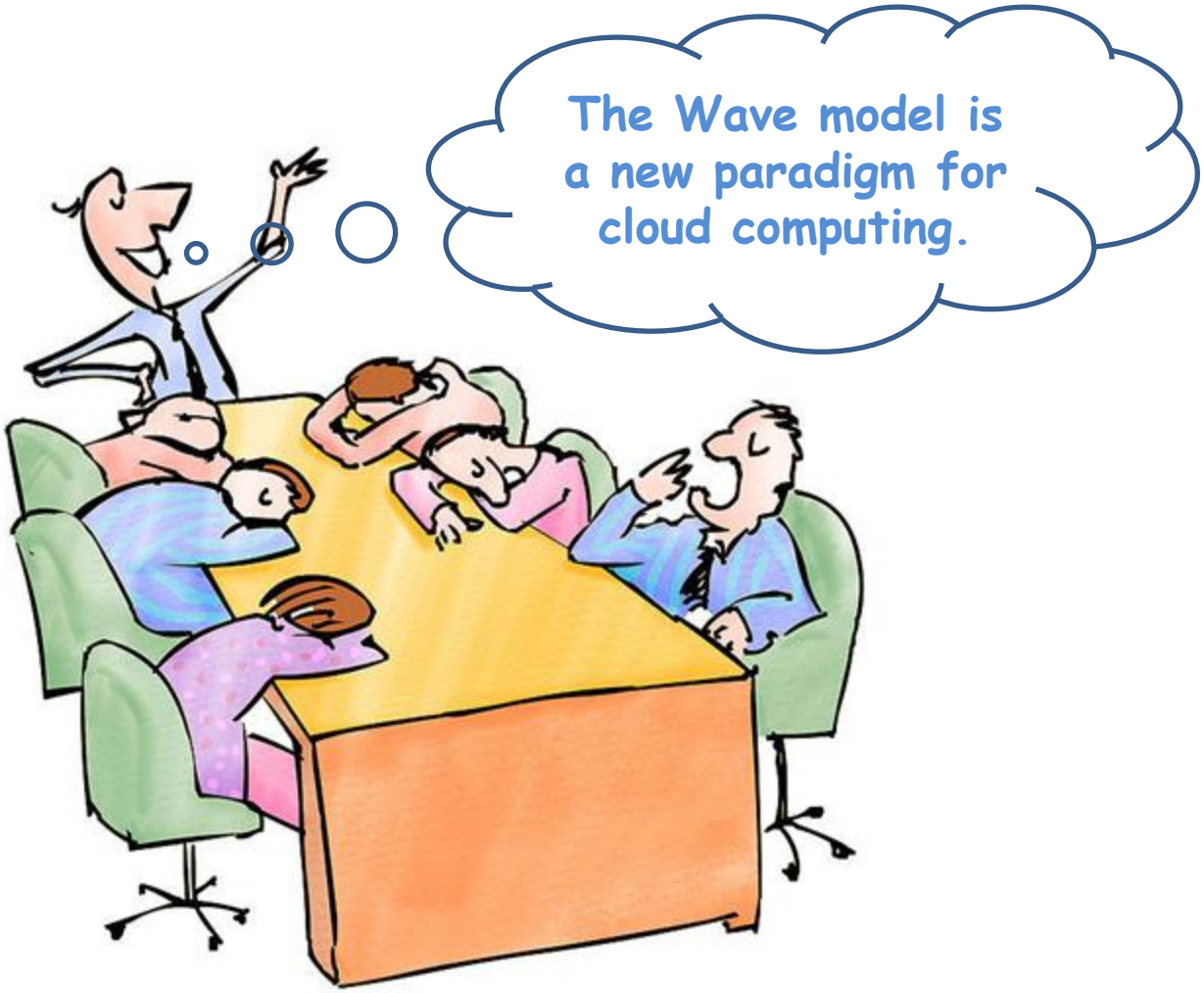
Microsoft Research Asia

Joint work with Mao Yang, Zhenyu Guo, Rishan Chen,

Wei Lin, Bing Su, Hongyi Wang, Lidong Zhou

My Dream Wave Computing

But, Today, Wave Computing is Actually…



State-of-the-art in the Cloud

- We provide scalability and fault-tolerance on thousands of machines.
- We provide the query interference using high level languages.

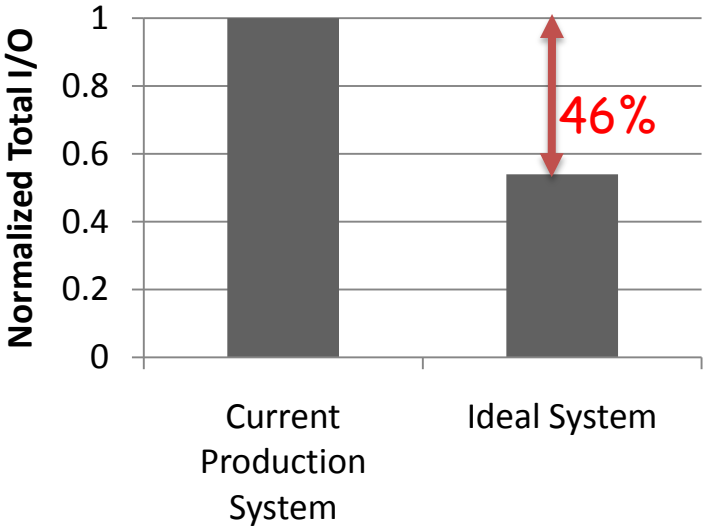
(MapReduce and its brothers: **G**. **Y**. **M**.)

Are G.Y.M.'s Executions Optimal?

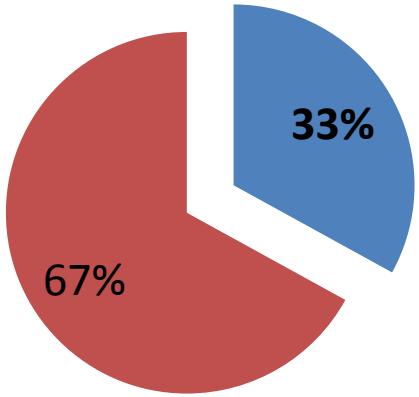
(Mr. Leopard)

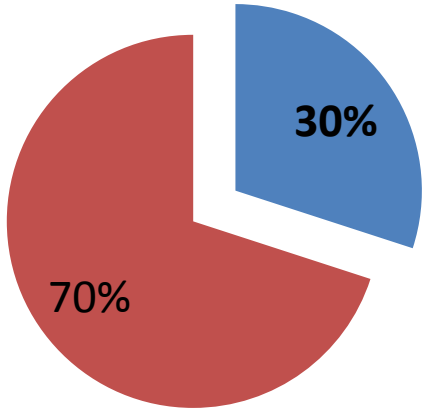
- We looked at a query trace from a production system (20 thousand queries, 29 million machine hours).
- We focused on the I/O and computation efficiency.

Our Finding: "Far From Ideal"



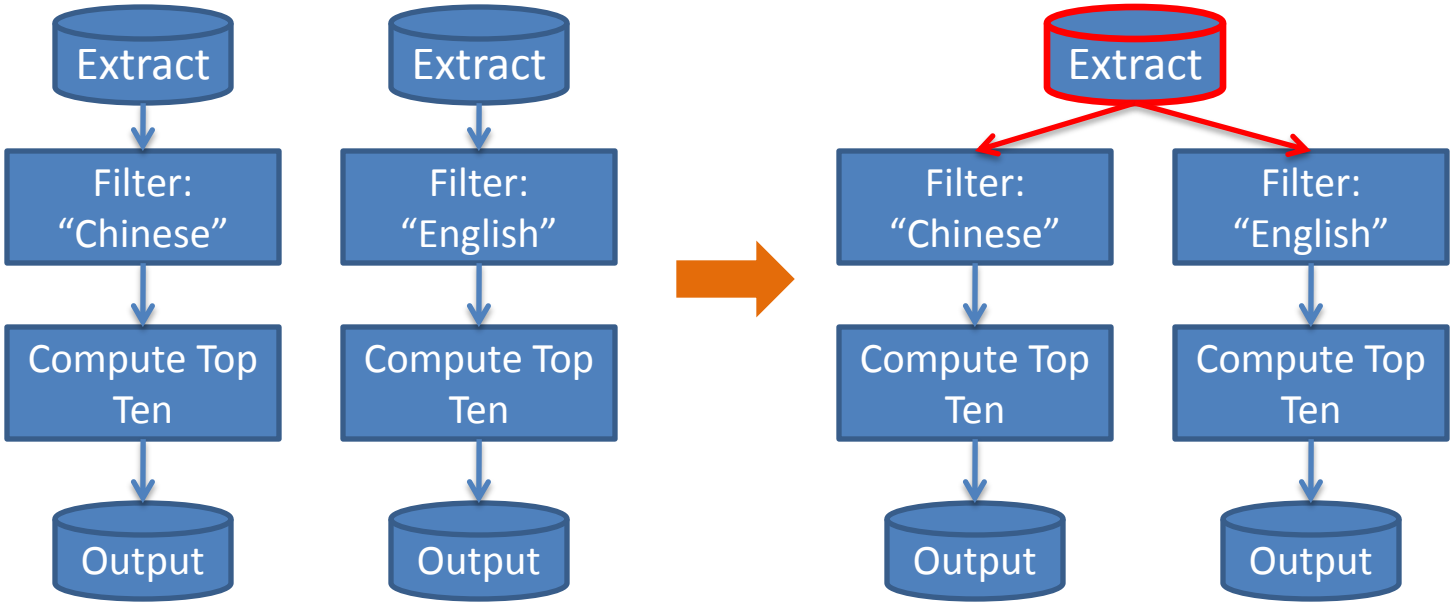
(Results from simulation)





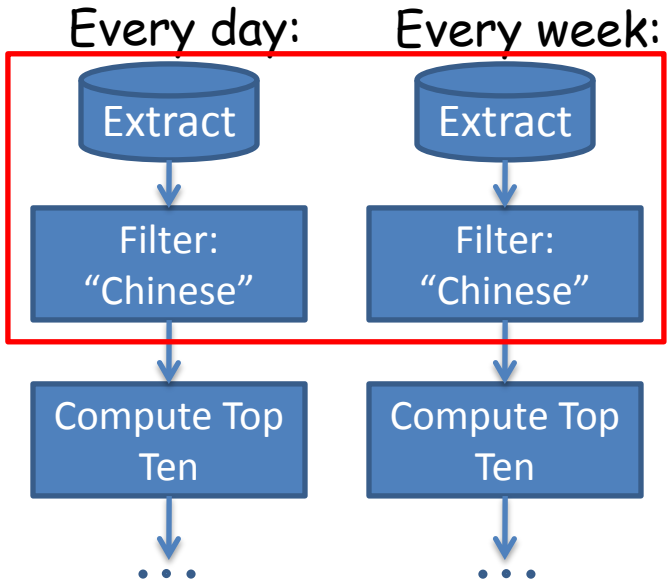
I/O Redundancy

- Two sample workloads
  - Obtaining the top ten hottest Chinese pages daily
  - Obtaining the top ten hottest English pages daily



Computation Redundancy

- Two sample workloads
  - Obtaining the top ten hottest Chinese pages daily
  - Obtaining the top ten hottest Chinese pages weekly



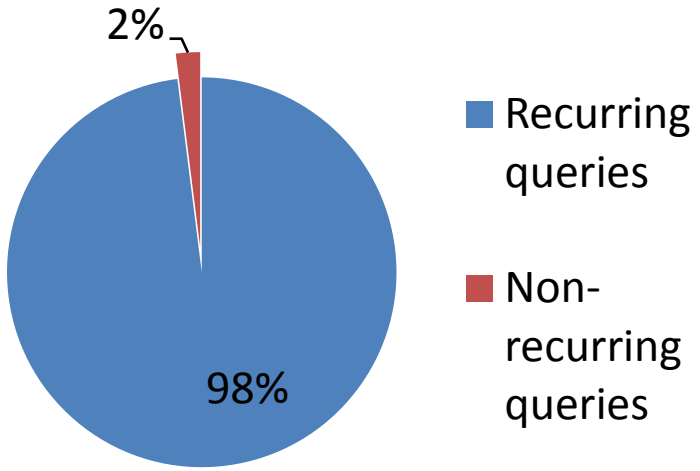
Common computation on per-day log (Ideally)

Why?

Correlations among queries

– Temporal correlations among queries

(A series of queries with recurrent computation)

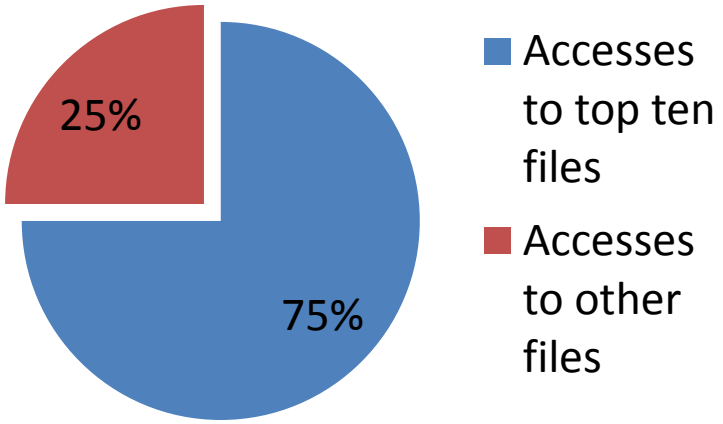


Why?

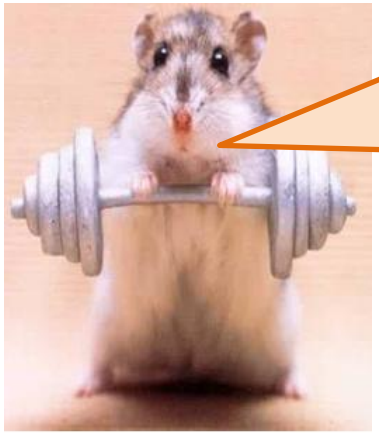
Correlations among queries

– Spatial correlations among queries

(Input data are targeted by multiple individual queries)



How To Exploit the Correlations?



(G.Y.M.)

Err… This is a little tricky. What about developing these?
- a probabilistic model on scheduling the input data access
- a predictive cache server
- a speculative query decomposer.

(Mr. Leopard)

No… Let's K.I.S.S.:
- Since correlations are inherent, we need a notion to capture them.
- Our solution is the Wave model to capture the correlation for both the user and the system.

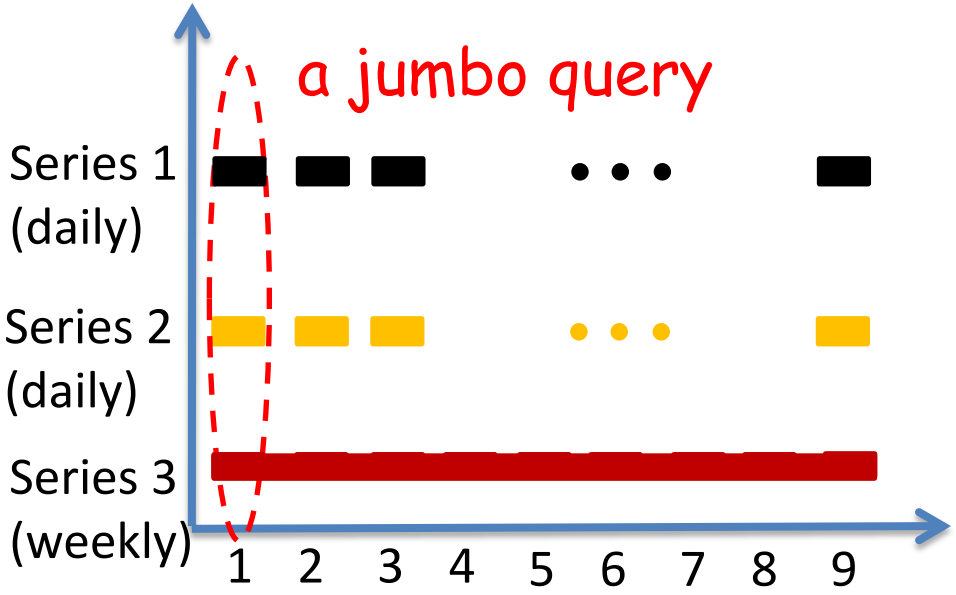
The Wave Model

- Key concepts capturing the correlation among queries
  - Data: not a static file, but a stream with periodically updated (*append-only*)
  - Query: computation on the input stream
  - Query series: recurrent computation on the stream

Optimization Opportunities in Waves

- Shared scan
  - Identifies the same input stream accesses among queries
- Shared computation
  - Identifies common computation steps among queries
- Query decomposition
  - Decomposes a query into a series of smaller queries
  - Uncovers more opportunities for shared scan and computation

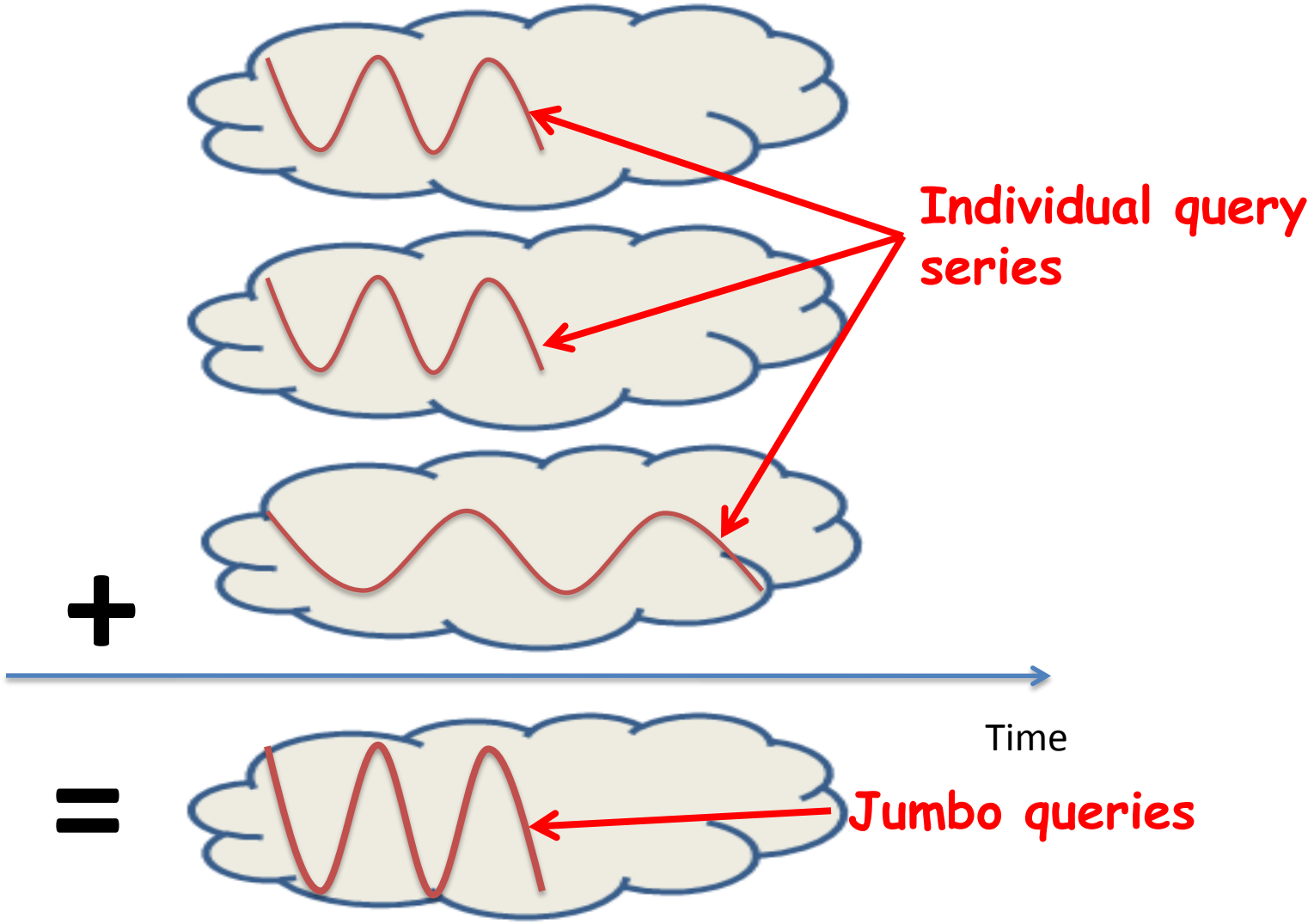
Query Optimizations in Wave Computing



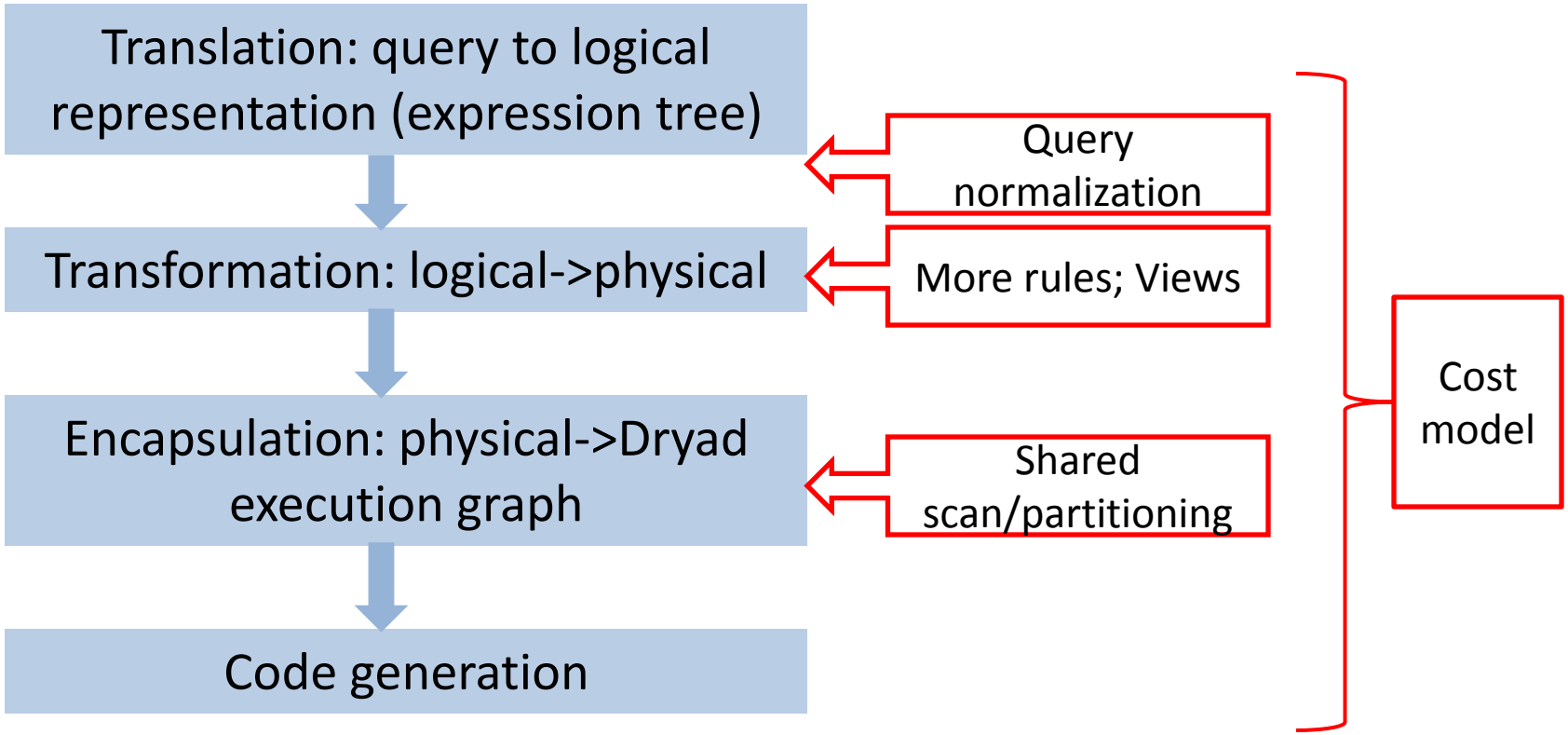
- Decomposition
- Form jumbo queries
- Optimizations on jumbo queries
  - Shared scan and computation

Query series 1: Obtaining the top ten hottest Chinese pages daily;
Query series 2: Obtaining the top ten hottest English pages daily;
Query series 3: Obtaining the top ten hottest Chinese pages weekly;

Ultimate (Wave+Cloud)



Comet: Integration into DryadLINQ



An Example of Query Decomposition in DryadLINQ

```
// Q2: weekly histogram aggregation grouping on (A,B)
q2 = env.Extractor("log?today-6...today")
  .Select(x => new {x.A, x.B})
  .Where(x => x.A != "gb")
  .GroupBy(x => x) //grouping on (A,B)
  .Select(x => new {x.Key, a = x.Count()});
```

Decompose an operator

Q → seven daily queries + one combining query

Daily query

```
= env.Extractor("log?today")
  .Select(x => new { x.A, x.B })
  .Where(x => x.A != "gb")
  .GroupBy(x => x) //grouping on (A,B)
  .Select(x => new { x.Key, c = x.Count() });
  .ToDryadPartitionedTableLazy("q2dview?today");
```

Views (Cost estimation)

Combining

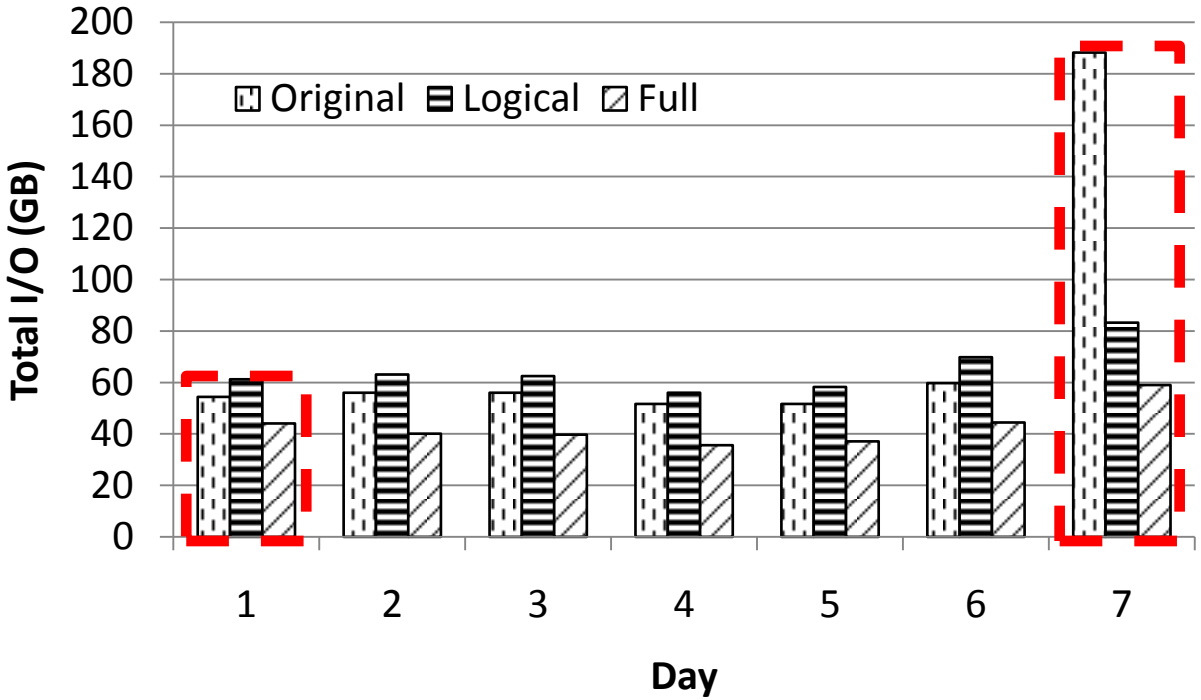
```
= env.Extractor2("q2dview?today-6...today")
  .AssumeHashPartition(x => x)
  .GroupBy(x => x)
  .Select(x => new {x.Key, c = x.Sum(y => y.c)});
```

Combine all the views

Automatic query decomposition is challenging.

Micro Benchmark

- Overall effectiveness
  - Logical optimization of Comet reduces 12.3% of total I/O.
  - Full (Logical + Physical optimizations) of Comet reduces 42.3% of total I/O.



(Running three sample queries on one week data of around 120 GB; A cluster of 40 machine)

Summary

- The Wave model is a new paradigm for capturing the query correlations in the cloud.
- The Wave model enables significant opportunities in improving performance and resource utilization.
- Comet: our ongoing project integrating Wave computing into DryadLINQ.