ELF♪

Efficient Lightweight Fast Stream Processing at Scale*

Liting Hu Karsten Schwan Hrishikesh Amur

Xin Chen

Georgia Institute of Technology

*Funded in part with support from the Intel Cloud ISTC

cercs

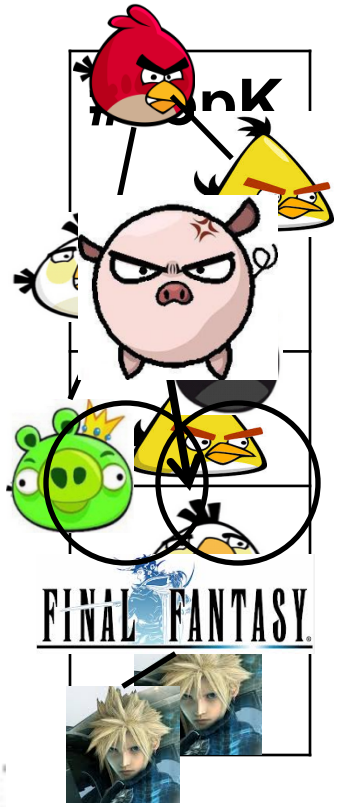
Georgia Institute of Technology

Outline

Motivation

ELF Design

Evaluation



Applications

Micro-promotions

Product-bundling

Sales-prediction

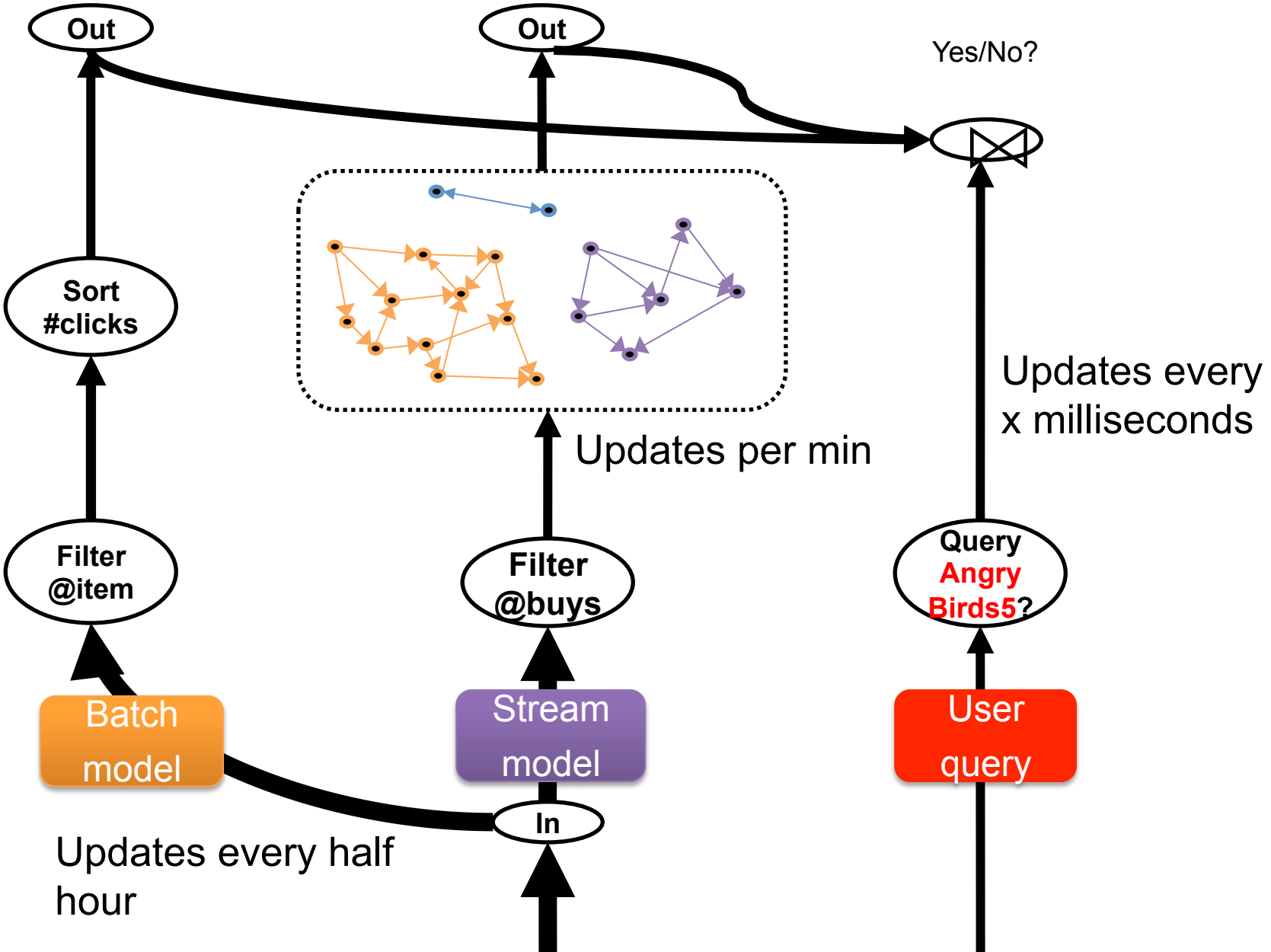
… … …

Coupon

Recommendations

Predictions

Concurrent, diverse jobs with varied delays



System Needs

Flexible

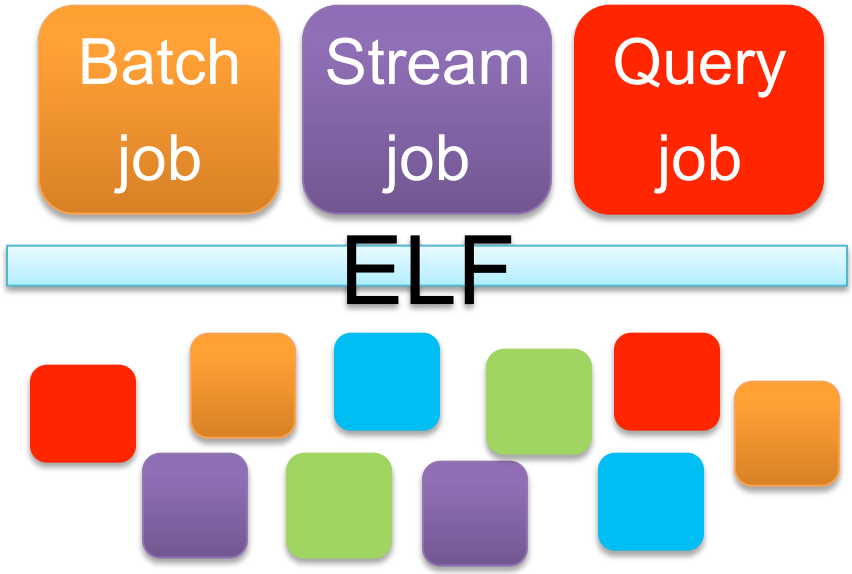
- Diverse jobs
- Cross-job coordination
- Runtime job change

Scalable

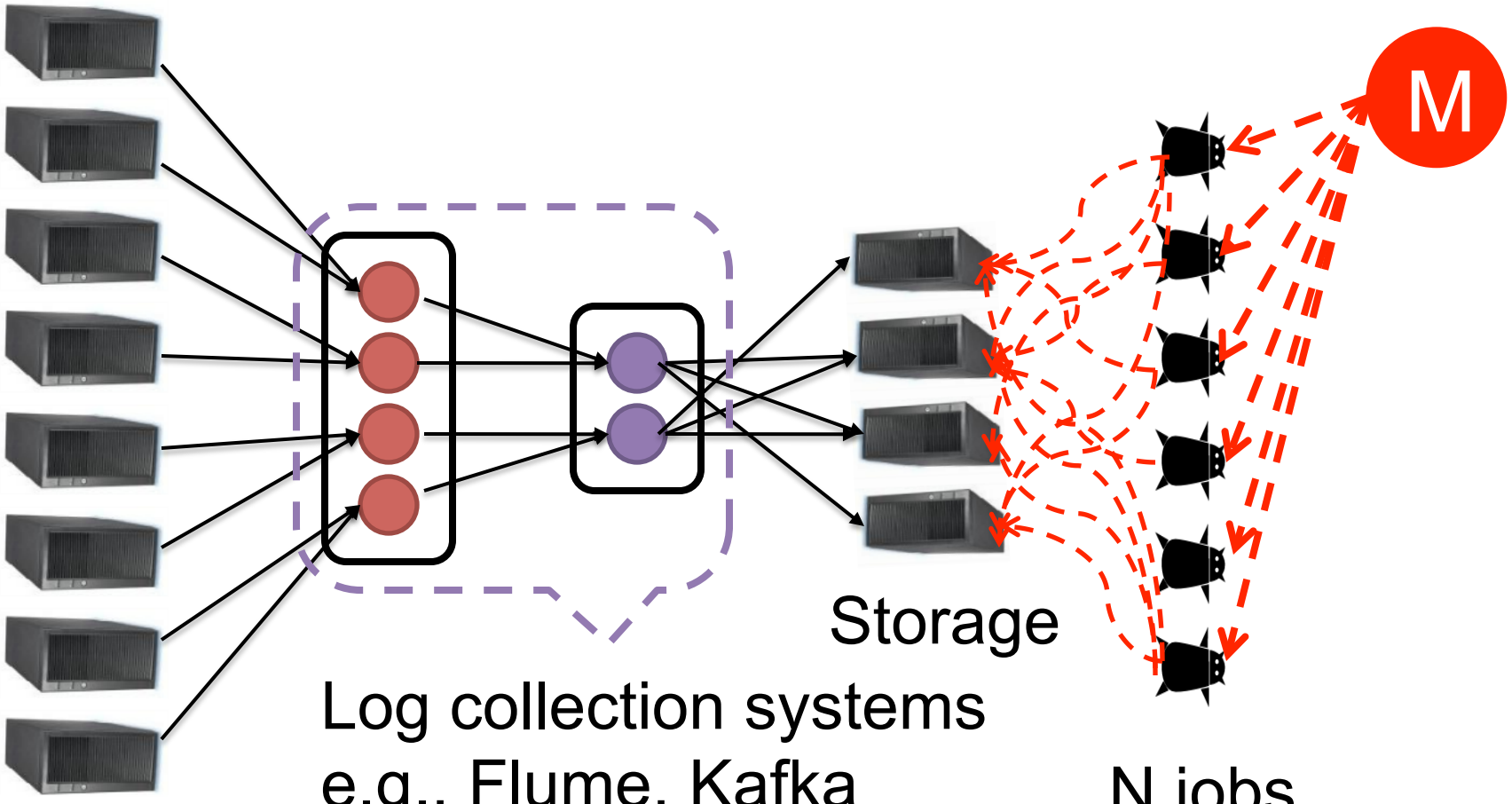
- Hundreds of concurrent jobs with shared inputs

Performance

- High throughput
- Low latency



Existing Solutions

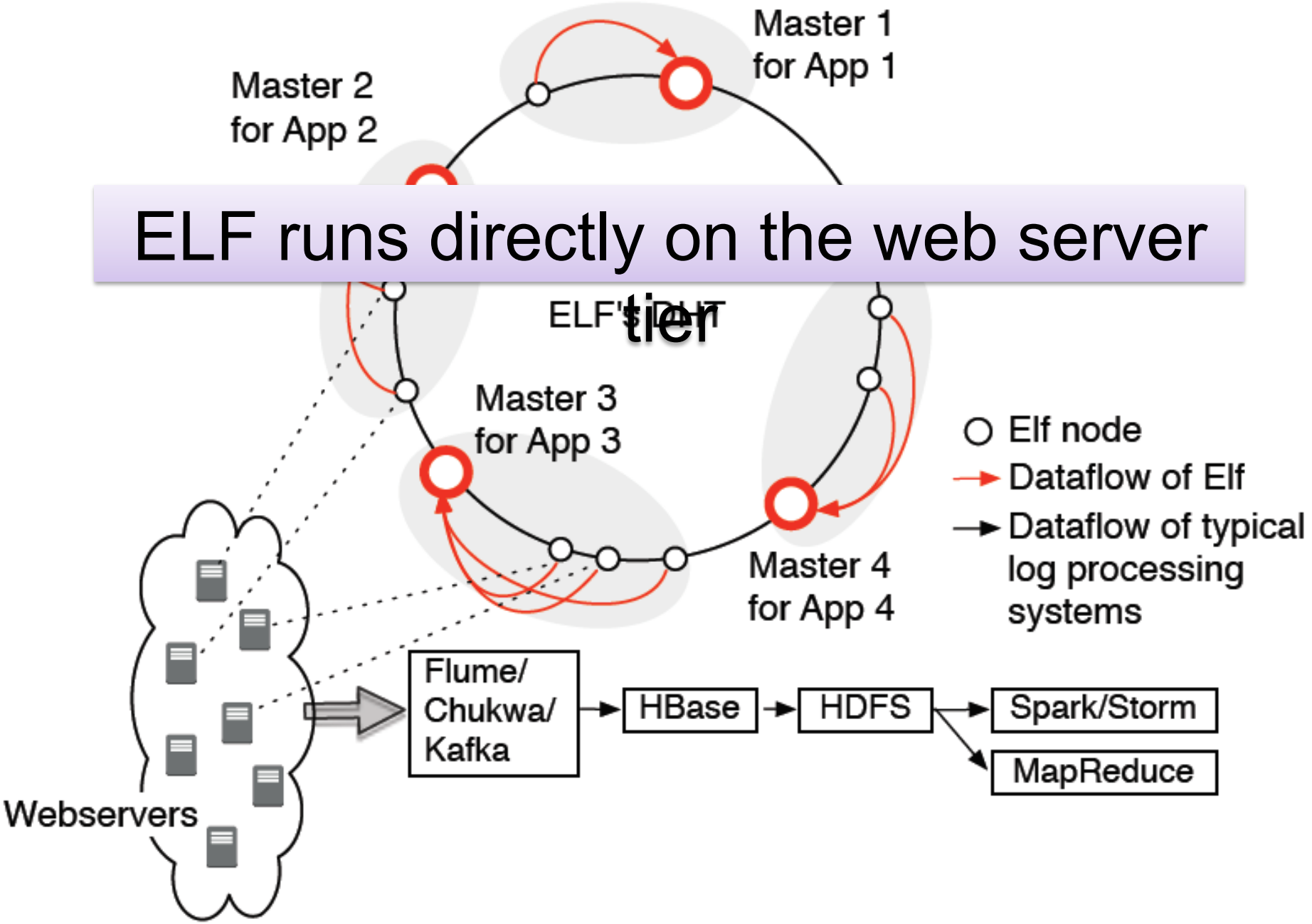


Outline

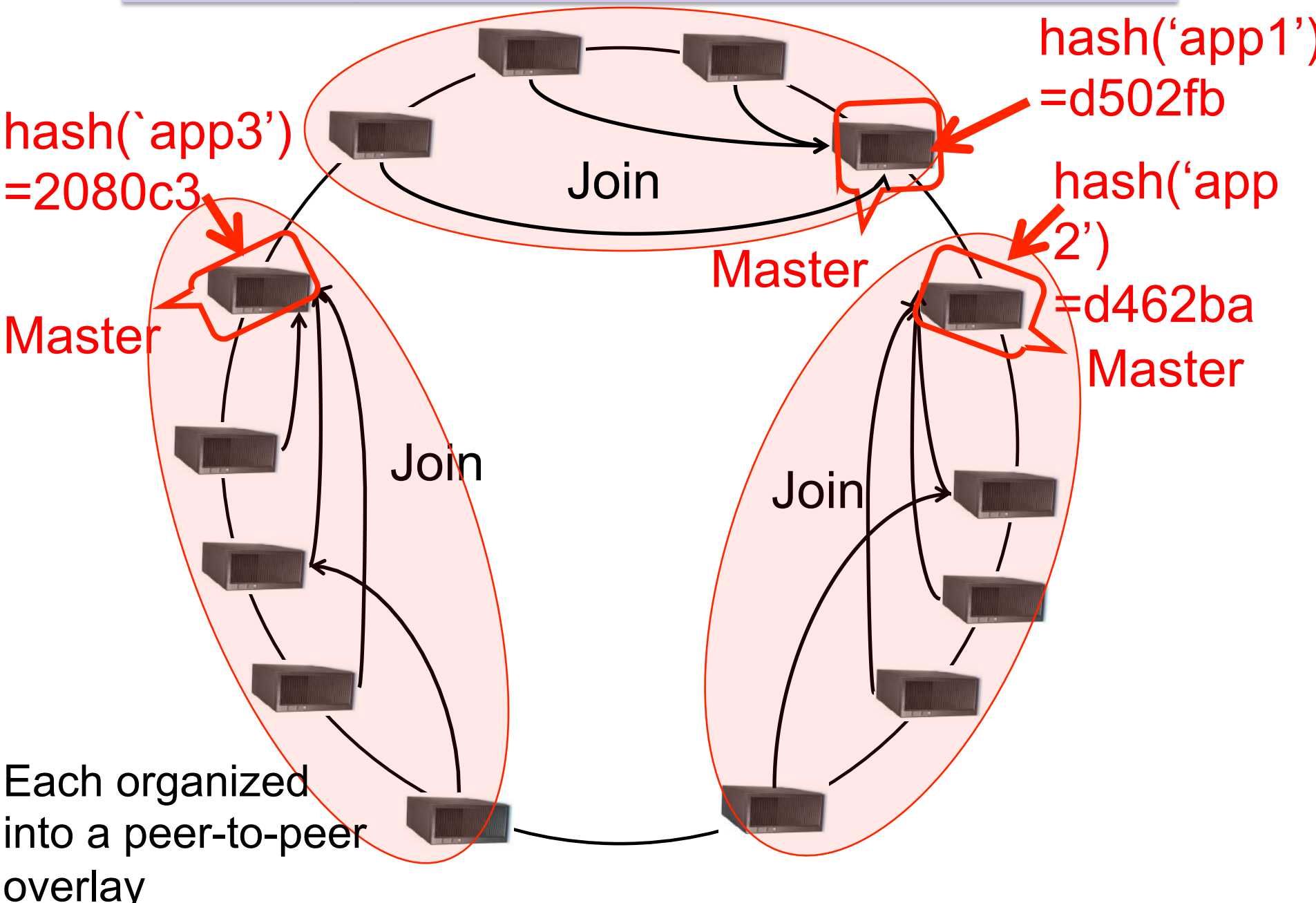
Motivation

ELF Design

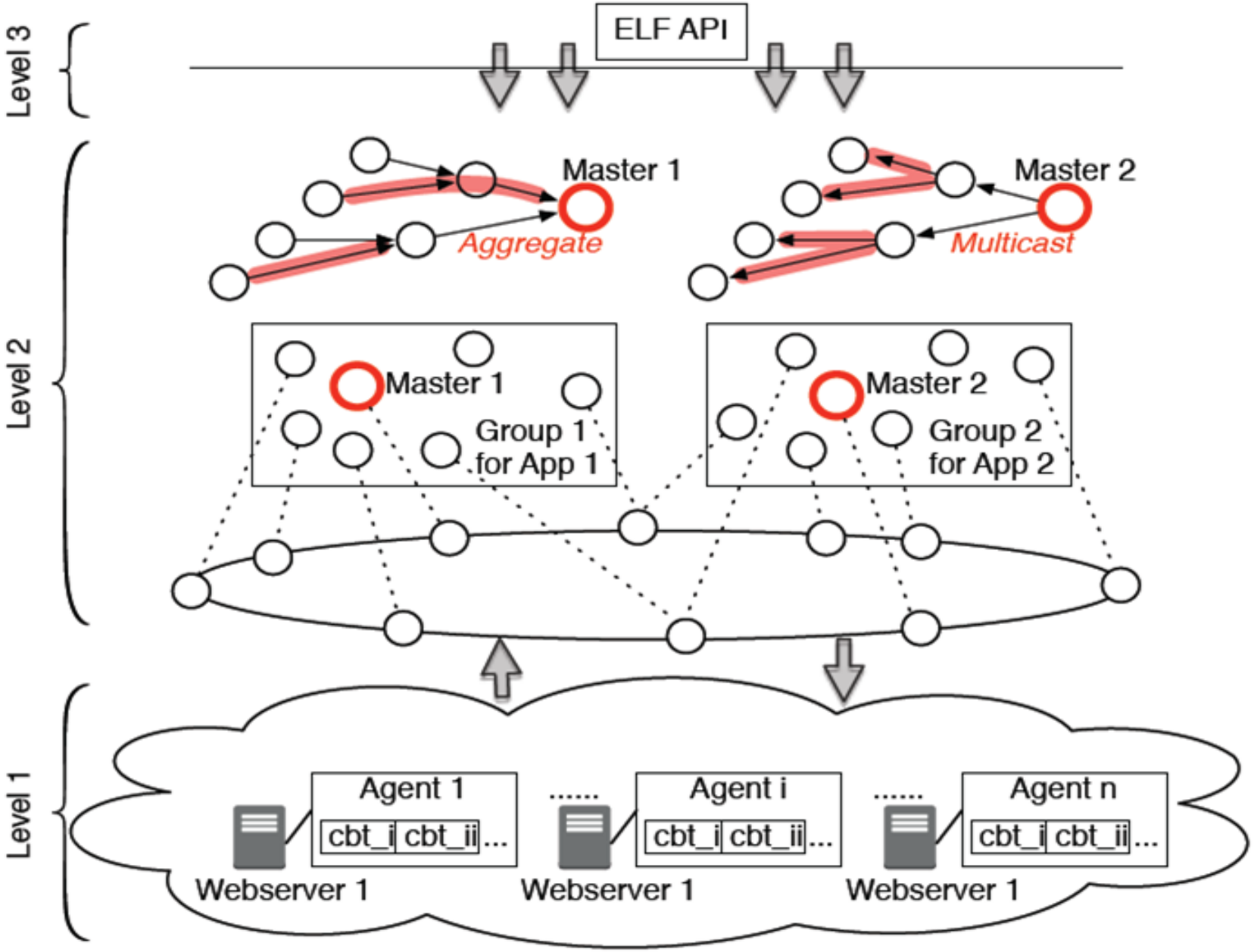
Evaluation


Master 1
for App 1
Master 2
for App 2
ELF runs directly on the web server
tier
Master 3
for App 3
Master 4
for App 4
Elf node
Dataflow of Elf
Dataflow of typical
log processing
systems
Webservers
Flume/
Chukwa/
Kafka
HBase
HDFS
Spark/Storm
MapReduce

Many Masters/Many Workers



ELF Software Architecture



ELF Design

CBT-based agent `pre-reduces' local key-value pairs

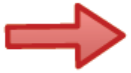
SRT globally `shuffles' and `reduces' key-value pairs

ELF APIs for programmers to implement diverse jobs

Sample ELF Query

Example log event

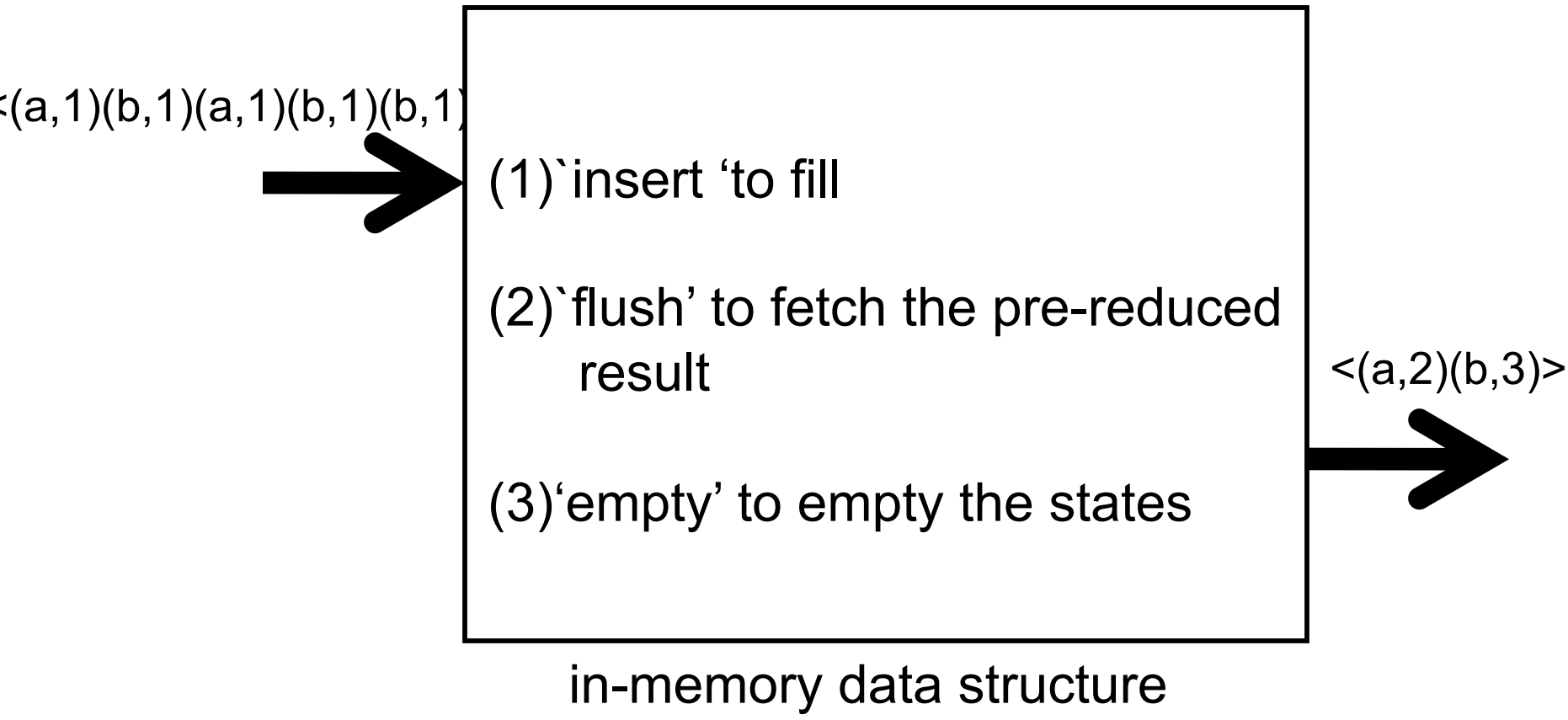
{"created_at":"23:48:22 +0000 2013",
"id":29966594182495027 3,
"product":"Angry Birds Season",
"clicks_count":2,
"buys_count":0,
"user":{"id":343284040,
"name":"@Curry",
"location":"Ohio", ...} ...}



ELF QL ->

```
SELECT product,SUM(clicks_count)
FROM *
WHERE store == `video_games'
GROUP BY product
SORT BY SUM(clicks_count) DESC
LIMIT 10
WINDOWING 30 SECONDS;
```

Compressed Buffer Tree (CBT)



ELF Design

CBT-based agent `pre-reduces' local key-value pairs

Stateful, asynchronous, and synchronous execution

SRT globally `shuffles' and `reduces' key-value pairs

ELF APIs for programmers to implement diverse jobs

ELF Design

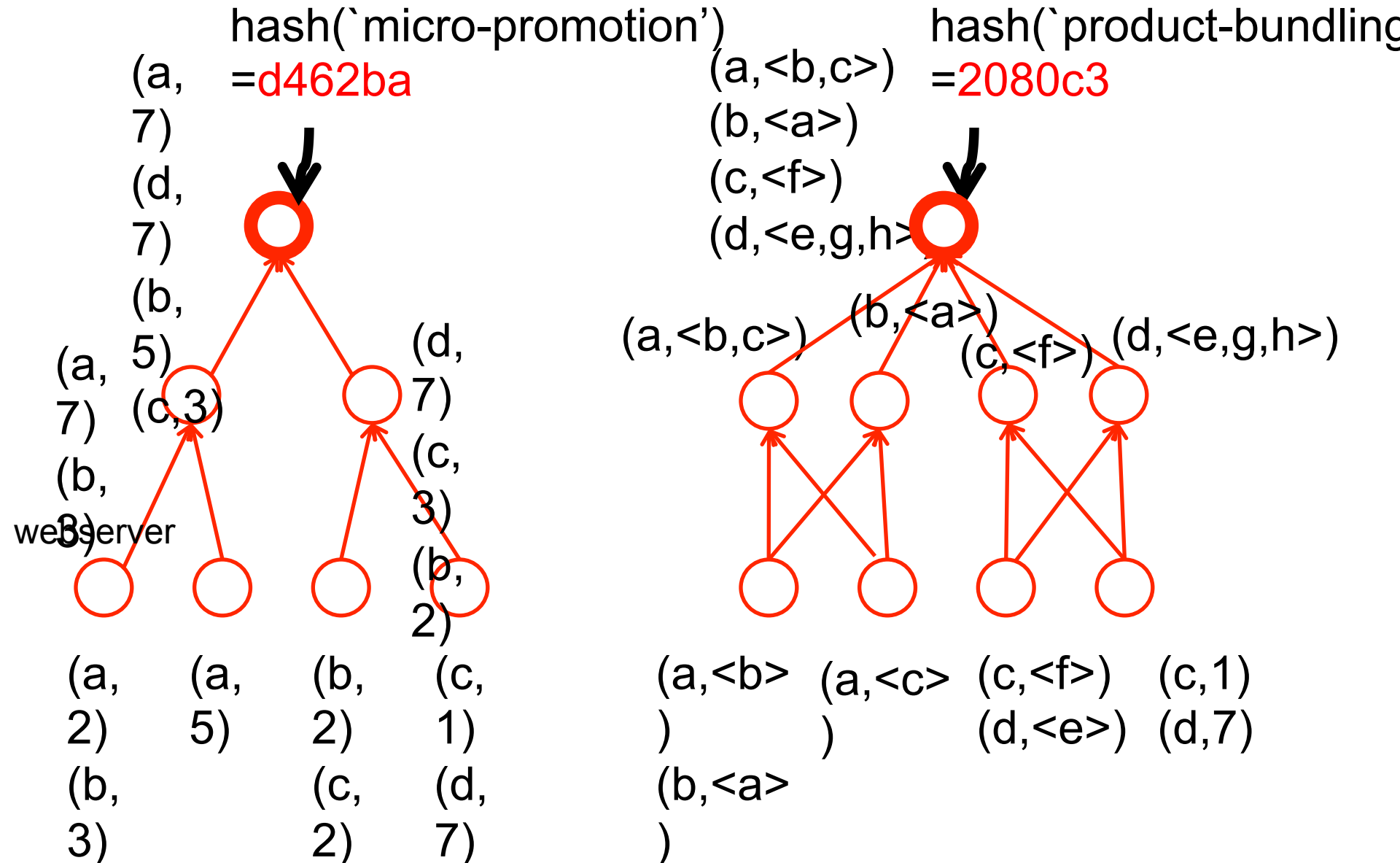
CBT-based agent `pre-reduces' local key-value pairs

Stateful, asynchronous, and synchronous execution

SRT globally `shuffles' and `reduces' key-value pairs

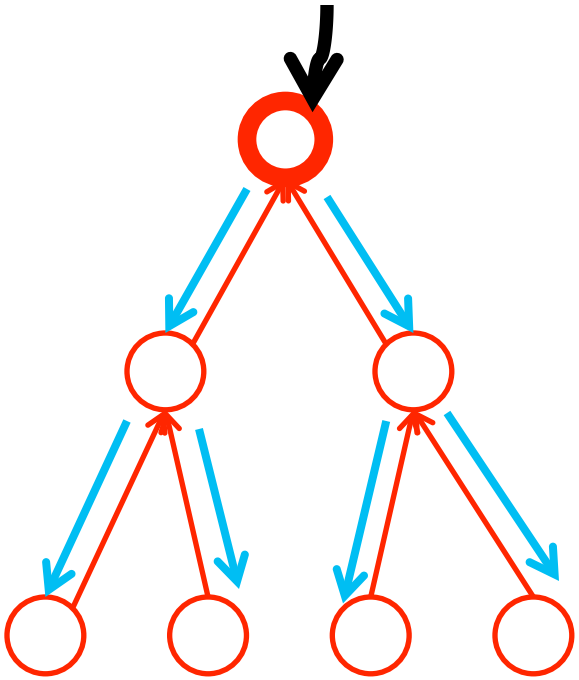
ELF APIs for programmers to implement diverse jobs

Per-job dataflow

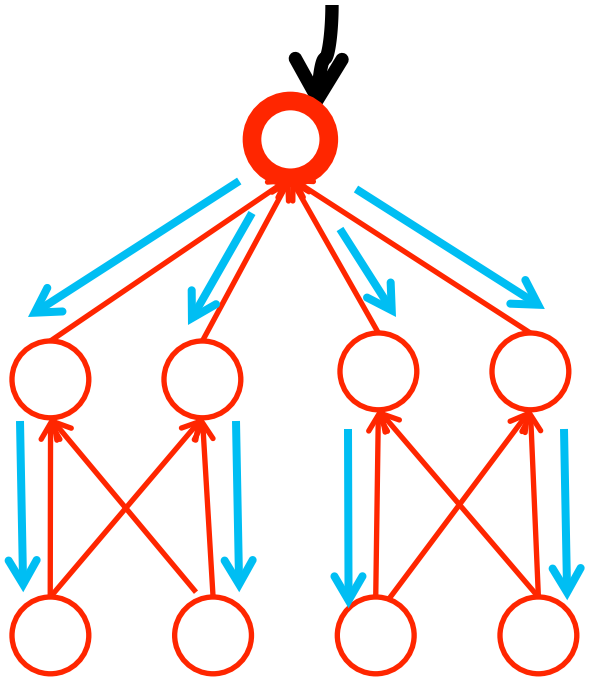


Per-job dataflow

hash(`micro-promotion’)
=d462ba

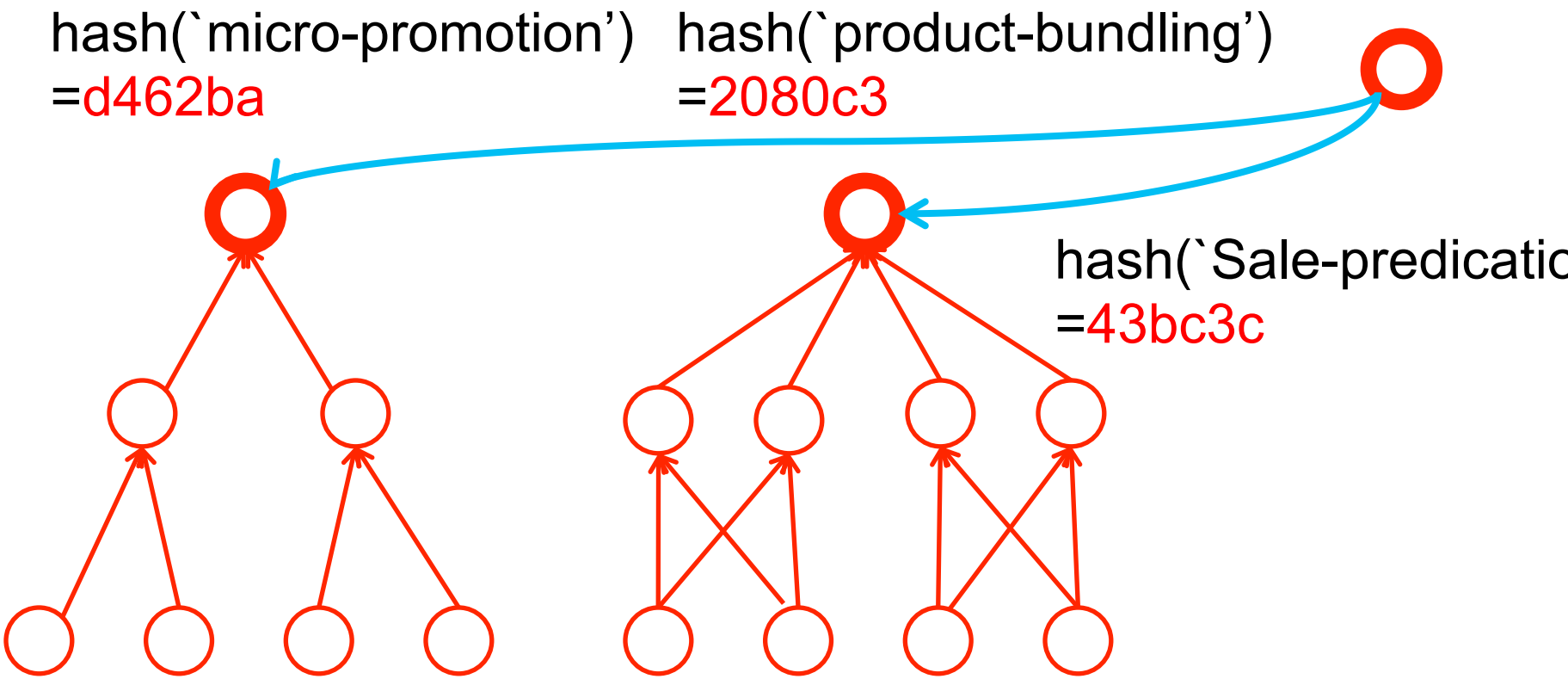


hash(`product-bundling
=2080c3



Synchronizing CBT to be emptied
Last iteration’s result
New job function or parameter

Cross-job Coordination



Limited to O(Log(N)) hops

ELF Design

CBT-based agent `pre-reduces' local key-value pairs

Stateful, asynchronous, and synchronous execution

SRT globally `shuffles' and `reduces' key-value pairs

Per-job dataflow, and cross-job coordination

ELF APIs for programmers to implement diverse jobs

ELF Design

CBT-based agent `pre-reduces' local key-value pairs

Stateful, asynchronous, and synchronous execution

SRT globally `shuffles' and `reduces' key-value pairs

Per-job dataflow, and cross-job coordination

ELF APIs for programmers to implement diverse jobs

Using ELF

```
ArrayList<String> topk;
void OnTimer () {
if (this.isRoot()) {
   this.Multicast(hash("micro-promotion"), new topk(topk));
   this.Multicast(hash("micro-promotion"), new update()); }
   }
void OnMulticast(Id appid, Message message) {
if (message instanceof topk) {
   for(String product: message.topk) {
      if(this.hasProduct(product))
      //if it is an topk message, appear discount ... }
      }
//if it is an update message, start a new batch
else if (message instanceof update) {
   //if leaves, flush CBT and update to the parent vertex
   if (!this.containsChild(appid)) {
      PAO paos = cbt.get(appid).flush();
      this.SendTo (this.getParent(appid), paos);
      cbt.get(appid).empty(); }
      }
   }
```

Outline

Motivation

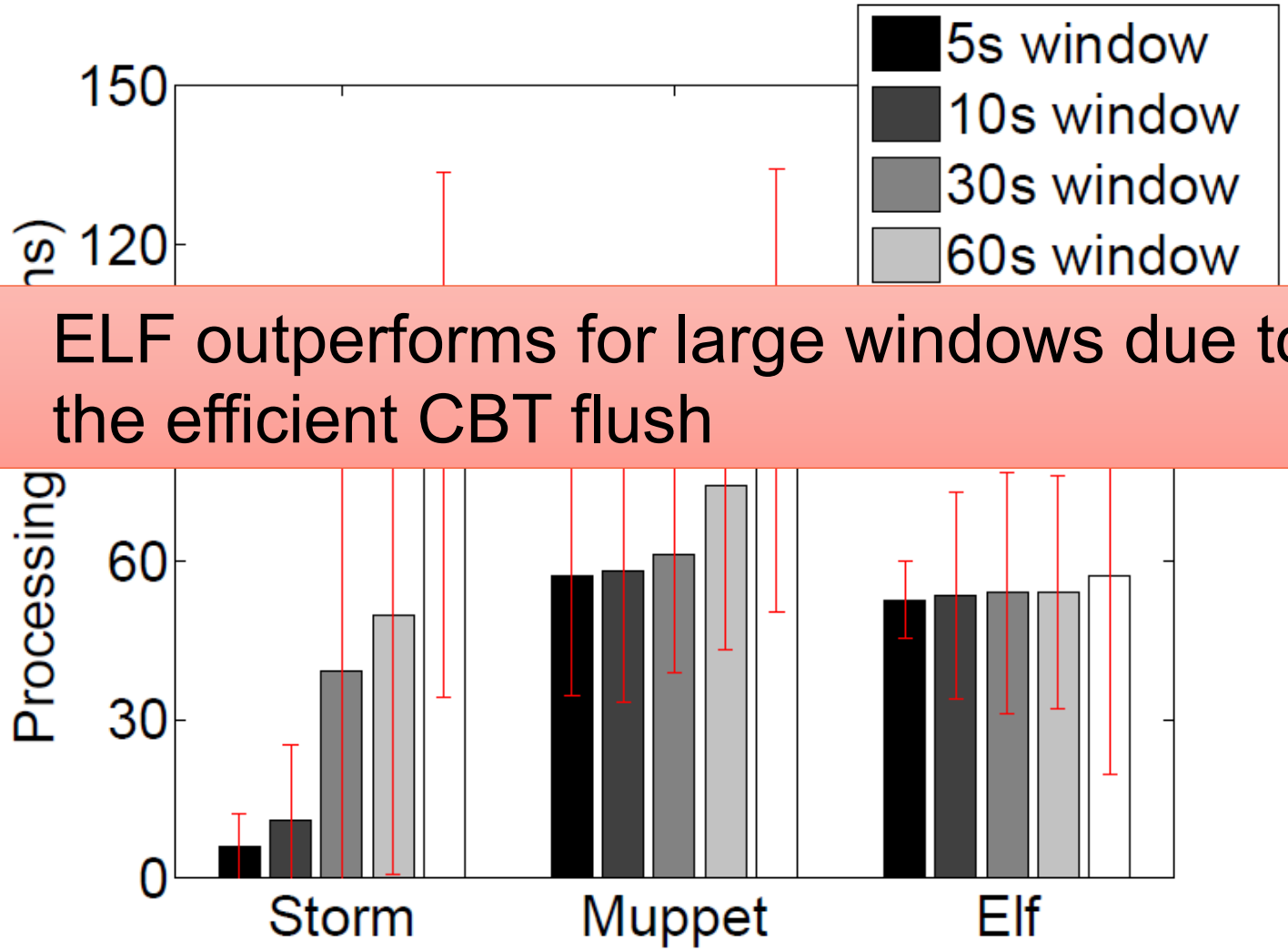
ELF design

Evaluation

Latency

Replay Twitter's stream
1280 agents
50 events/second

60 × 12-core 2.66 GHz AMD Opteron
48 GB RAM per server
Gigabit Ethernet

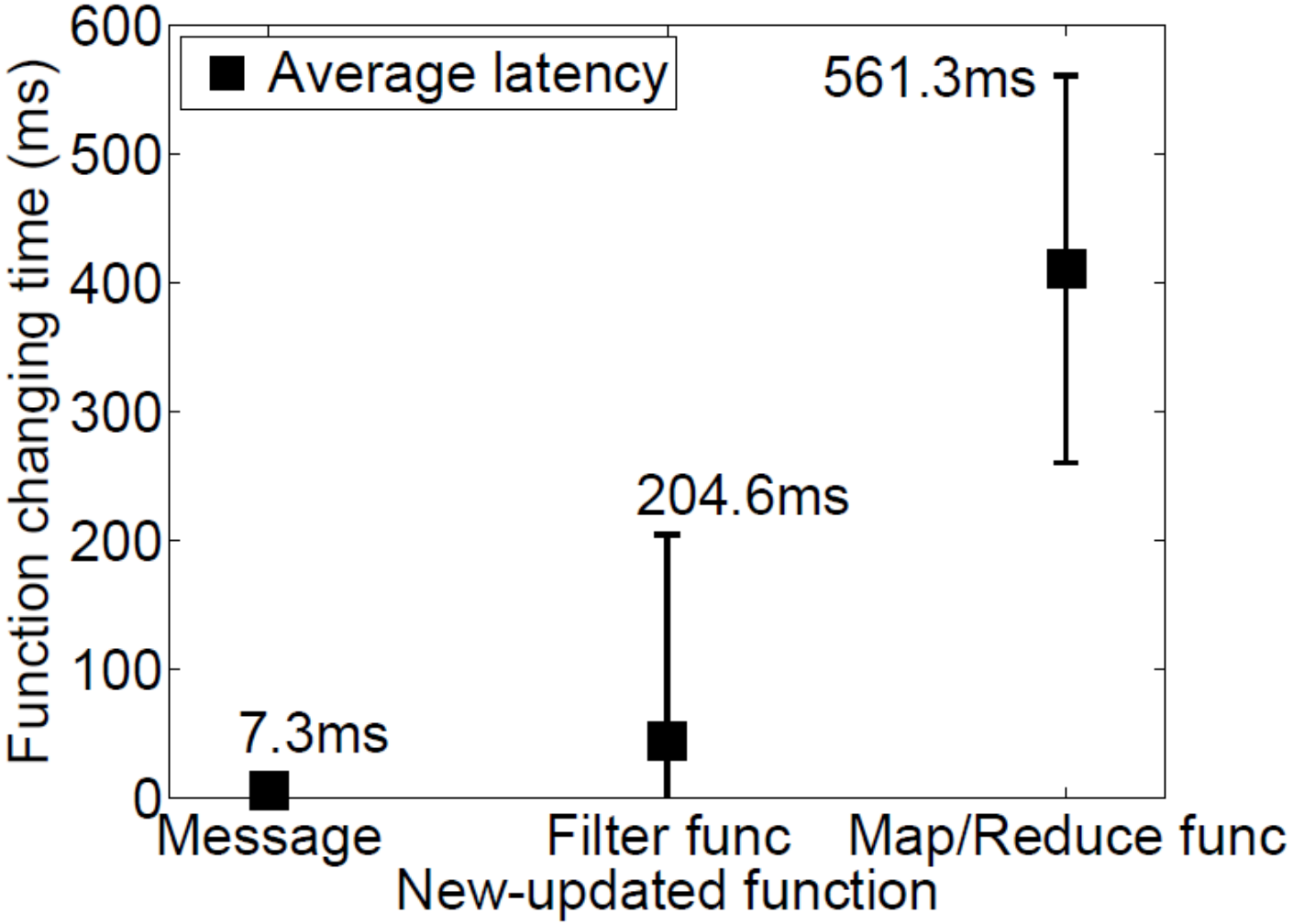


✓ ELF outperforms for large windows due to the efficient CBT flush

New Functionality

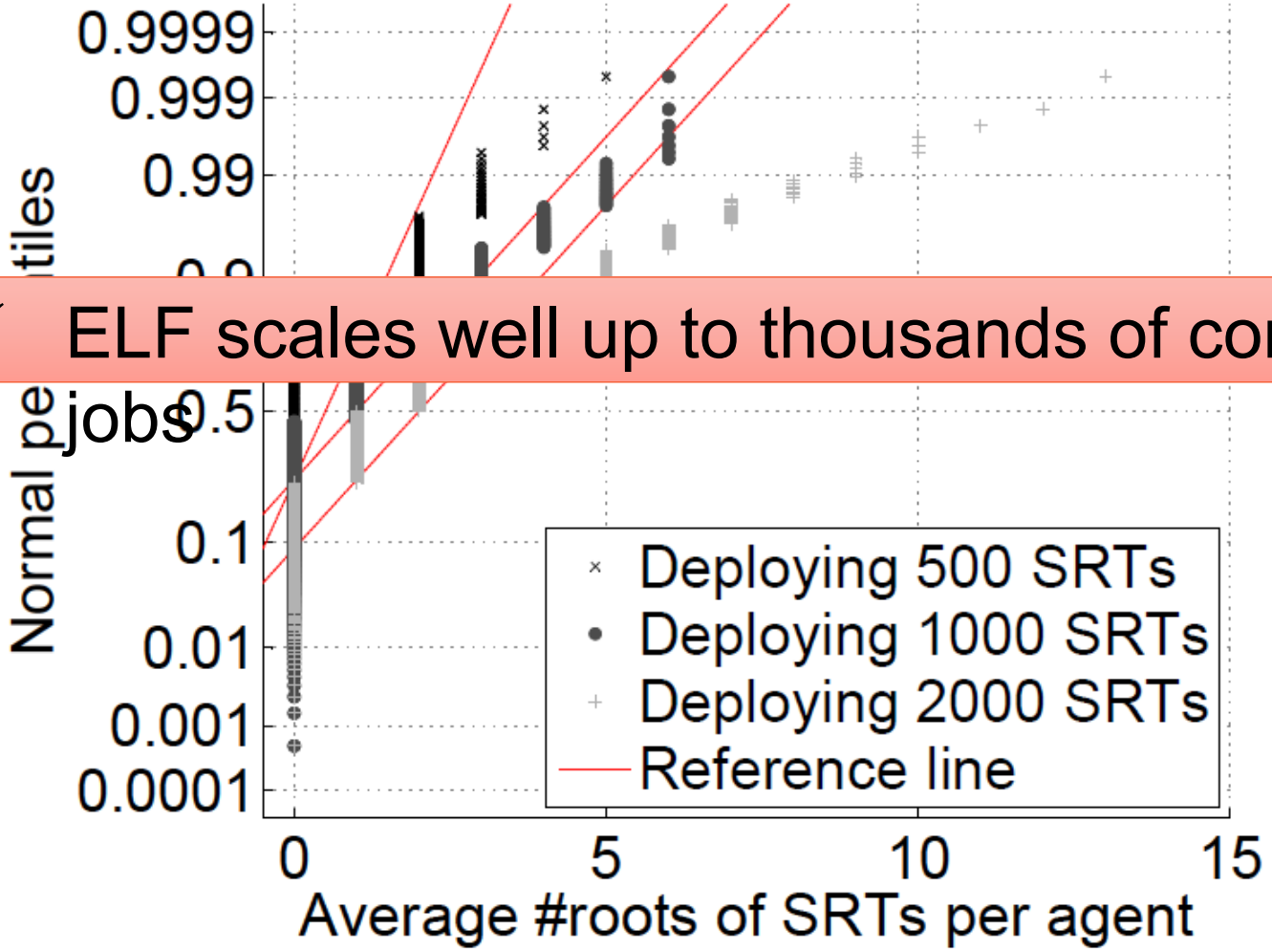
10s millisecond for multicast messages

100s millisecond to update functions



Scalability

60 × 12-core 2.66 GHz AMD Opteron
48 GB RAM per server
Gigabit Ethernet



✓ ELF scales well up to thousands of concurrent jobs

Conclusions and Future Work

ELF introduces a `many masters/many workers' framework:

- To achieve **good performance**: low latency and high throughput
- scalability: for diverse, concurrent jobs, and
- new functionality: job flexibility feedback and coordination.

Future work: understand the limitations of Elf; Explore more complex web-tier processing; look at task isolation.

Thank you!

Liting Hu
foxting@gatech.edu