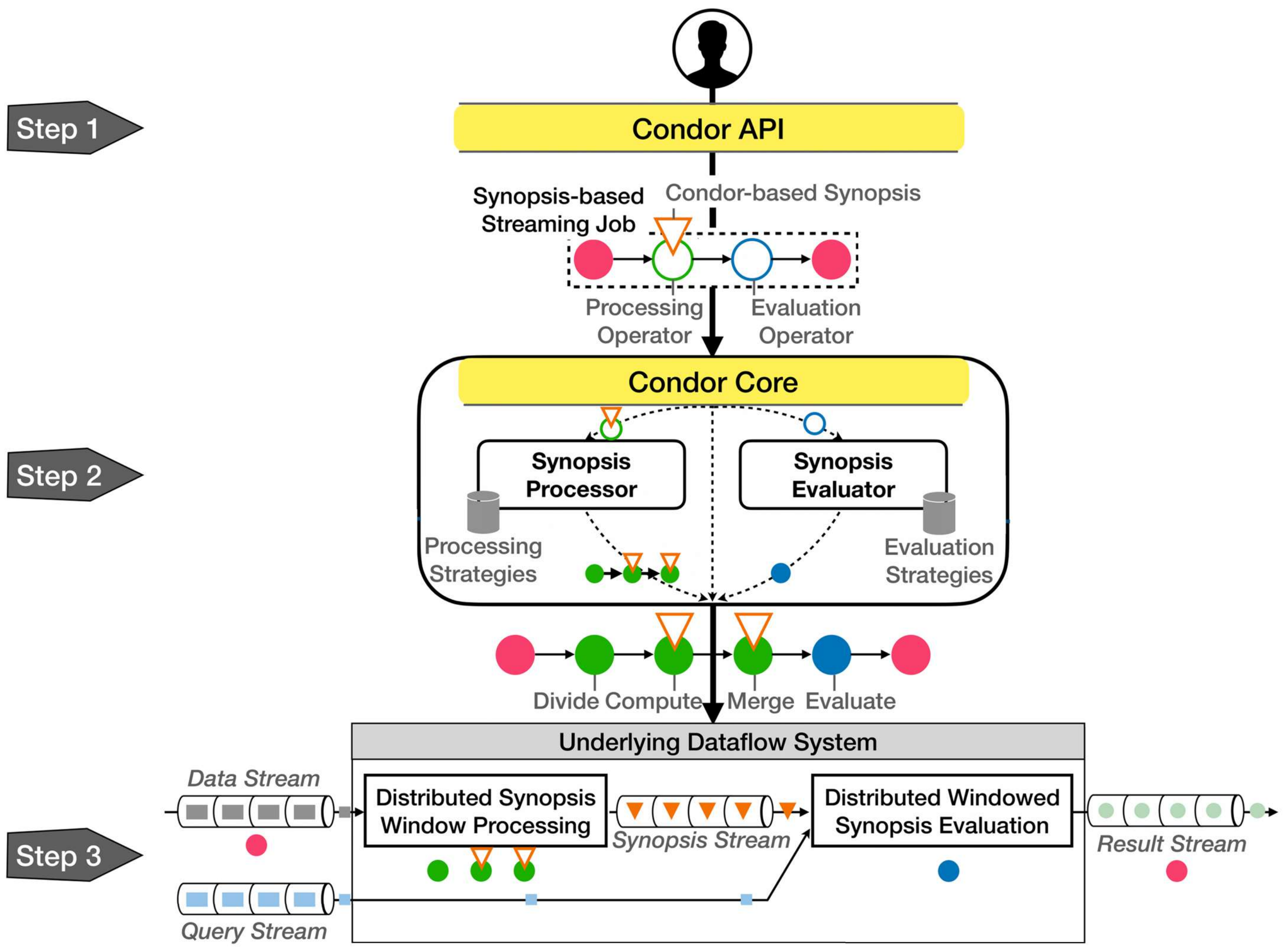
CONDOR

In the Land of Data Streams where Synopses are Missing, One Framework to Bring Them All

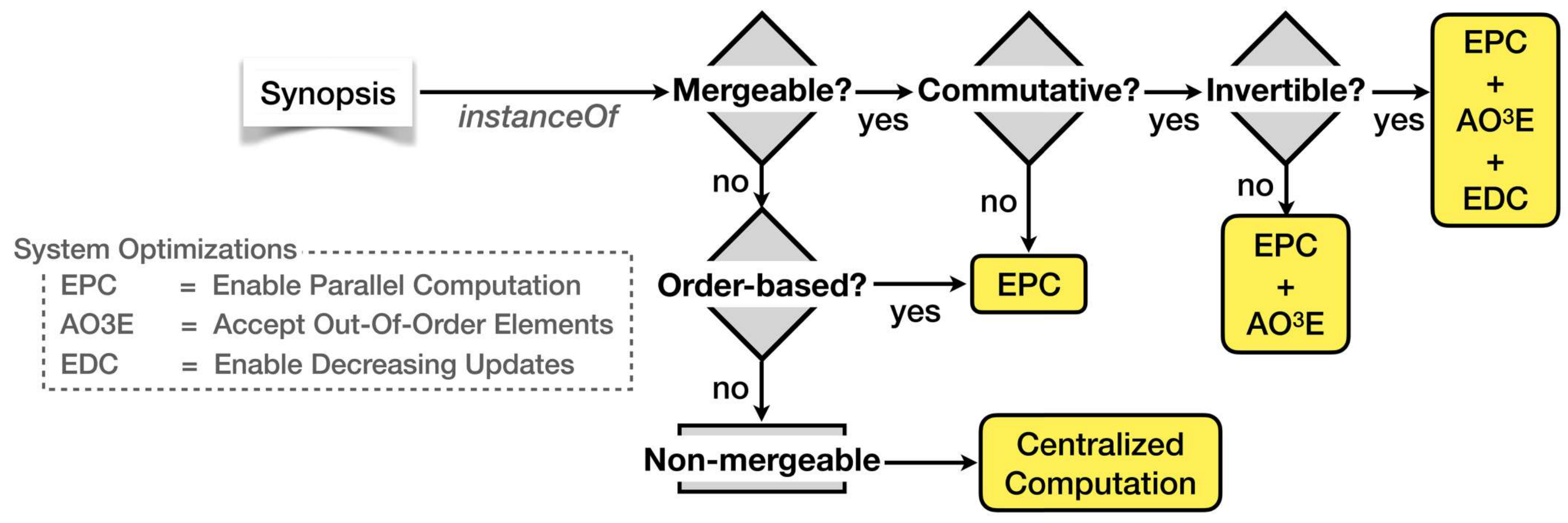
Rudi Poepsel-Lemaitre   Martin Kiefer   Joscha von Hein   Jorge-Arnulfo Quiané-Ruiz   Volker Markl

[r.poepsellemaitre, martin.kiefer, j.vonhein, jorge.quiane, volker.markl]@tu-berlin.de

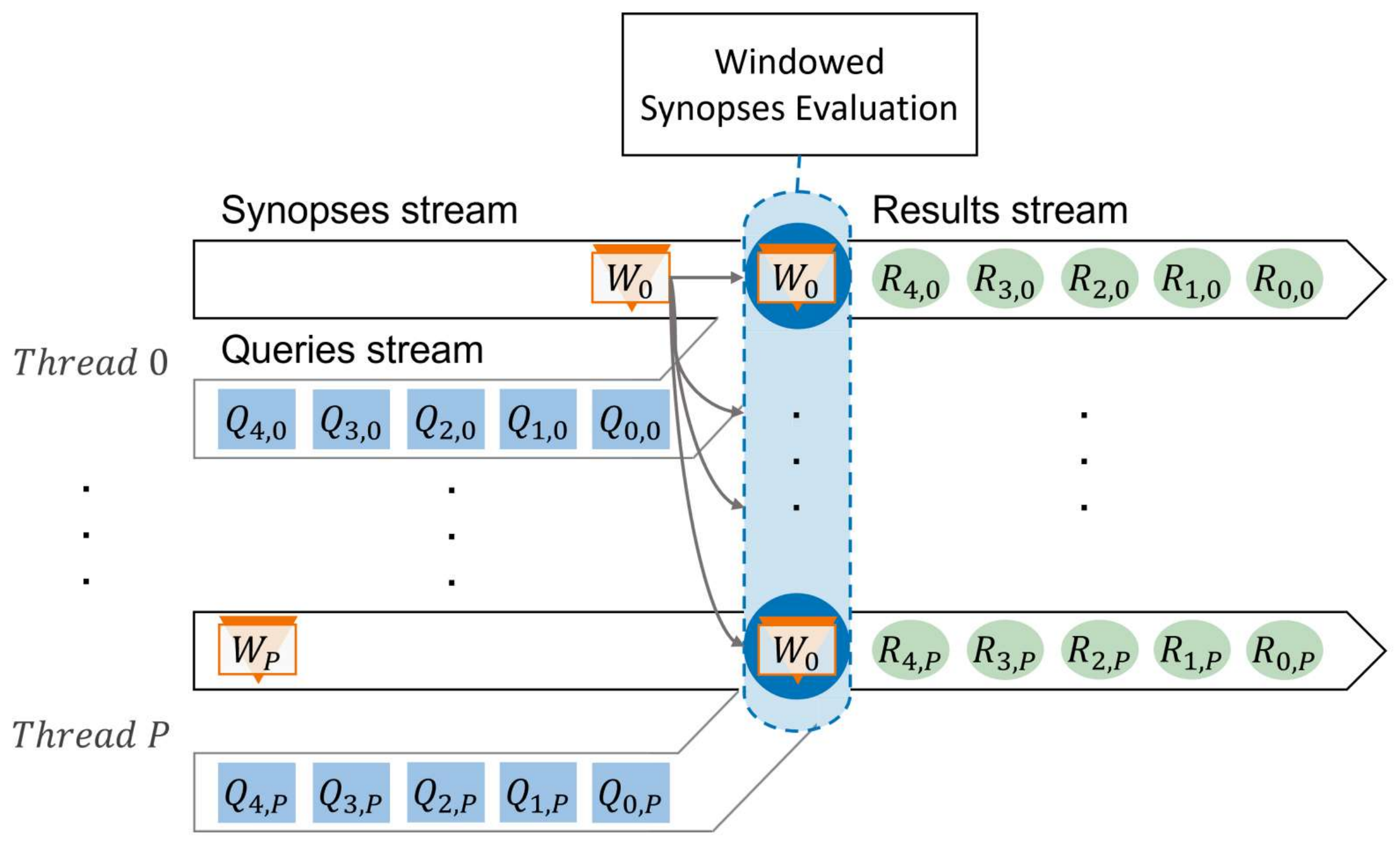
Architecture



Condor allows for the specification of synopsis-based streaming jobs on top of general dataflow engines. It provides a collection of twelve synopses and an integration to Apache Flink, however our techniques can be integrated into any dataflow engine that supports window processing. Users can integrate any new user defined synopsis into our framework using our API and Condor will provide a scalable processing environment.

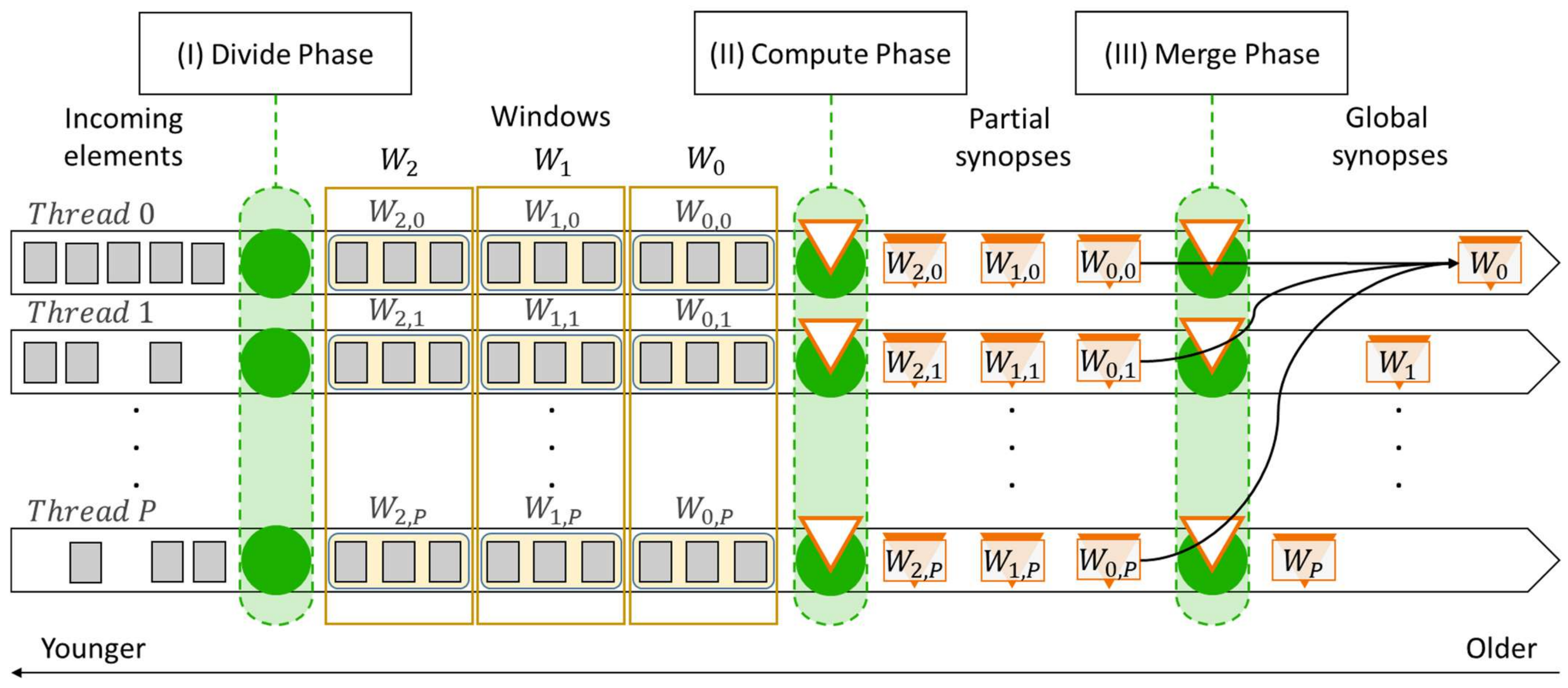


Synopses Evaluation


Windowed Synopses Evaluation
Synopses stream
Results stream
Thread 0
Queries stream
W_0
$R_{4,0}$ $R_{3,0}$ $R_{2,0}$ $R_{1,0}$ $R_{0,0}$
$Q_{4,0}$ $Q_{3,0}$ $Q_{2,0}$ $Q_{1,0}$ $Q_{0,0}$
Thread P
W_P
$R_{4,P}$ $R_{3,P}$ $R_{2,P}$ $R_{1,P}$ $R_{0,P}$
$Q_{4,P}$ $Q_{3,P}$ $Q_{2,P}$ $Q_{1,P}$ $Q_{0,P}$

Condor also offers a set of operators to efficiently evaluate a synopsis stream. These operators follow the broadcast state strategy, where every synopsis is broadcasted to all parallel instances of an operator, so that the system can evaluate them against all the elements coming from the query stream. This way, the system can query the same synopsis simultaneously on every thread, even as each synopsis was constructed only once.

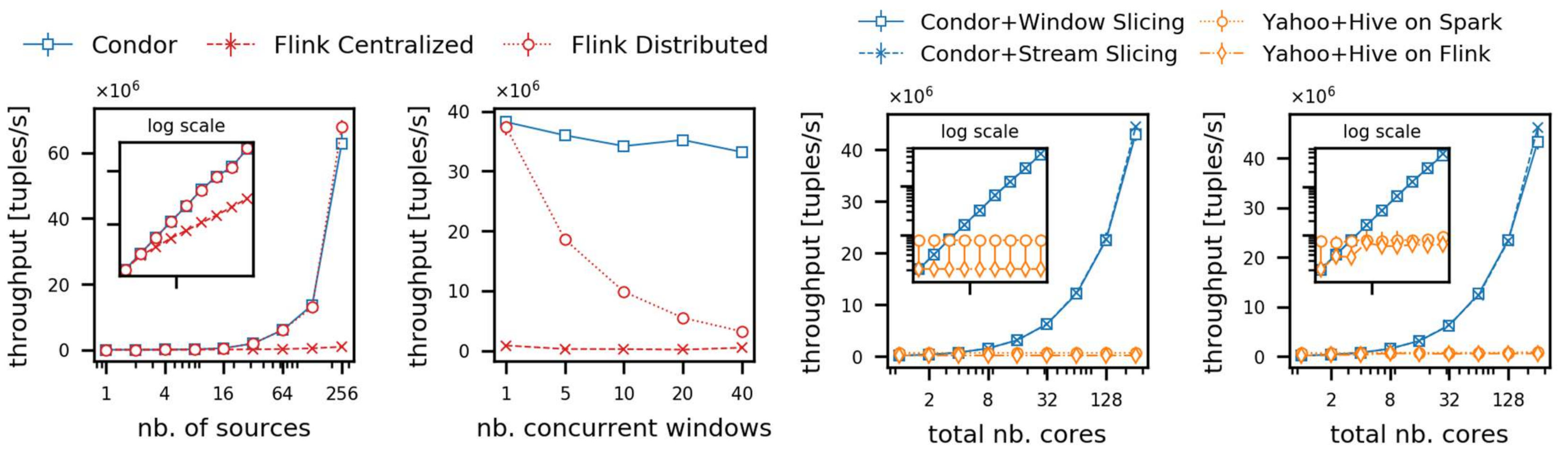
Synopses Computation



We model synopses as stateful window aggregate functions, doing so enables us to maintain any one-pass synopsis in a distributed setting as we can use the divide and conquer strategy. Condor splits this synopses computation into three phases and utilizes different processing strategies in each of them based on the user's configurations.

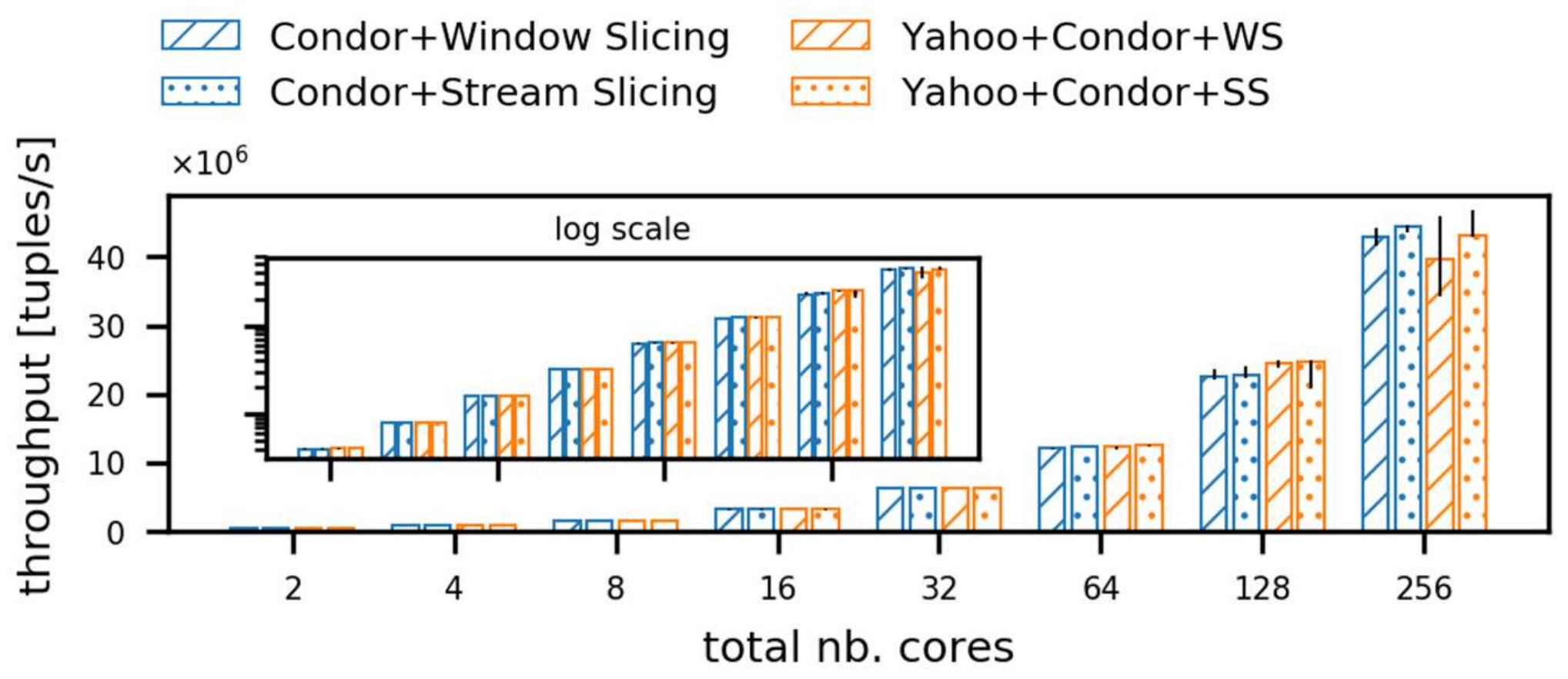
- **Divide Phase:** Condor distributes the input data stream among all nodes to fully exploit the available parallelism.
- **Compute Phase:** Condor maintains a synopsis for each partitioned window.
- **Merge Phase:** Condor merges the partial synopses without parallelism if both possible and required.

Evaluation



a) Serialization   b) Concurent Windows   c) Global Synopses   d) Stratified Synopses

Condor outperforms custom one-off implementations in Flink (Figures a) and b)), showing high performance even in scenarios with a high number of concurrent windows.
Condor also outperforms related works as Yahoo's DataSketches (Figures c) and d)), showing that our implementation is specifically designed for high parallelism applications.



e) Scalability Yahoo! DataSketches with Condor

An essential feature of Condor is that it allows users to implement their synopses via a simple API. This way, users can focus on the application logic instead of intricate internal details. We tested this feature by adapting Yahoo's HLL sketch implementation to our API, showing that Condor enables any mergeable synopsis to scale linearly with the parallelism.

Bundesministerium für Bildung und Forschung

software campus

holtzbrinck Publishing Group

Open Source Repository
TU-Berlin-DIMA/Condor