Scaling up Correlation Clustering through Parallelism and Concurrency Control

Xinghao Pan

Dimitris Papailiopoulos

Benjaminn Recht

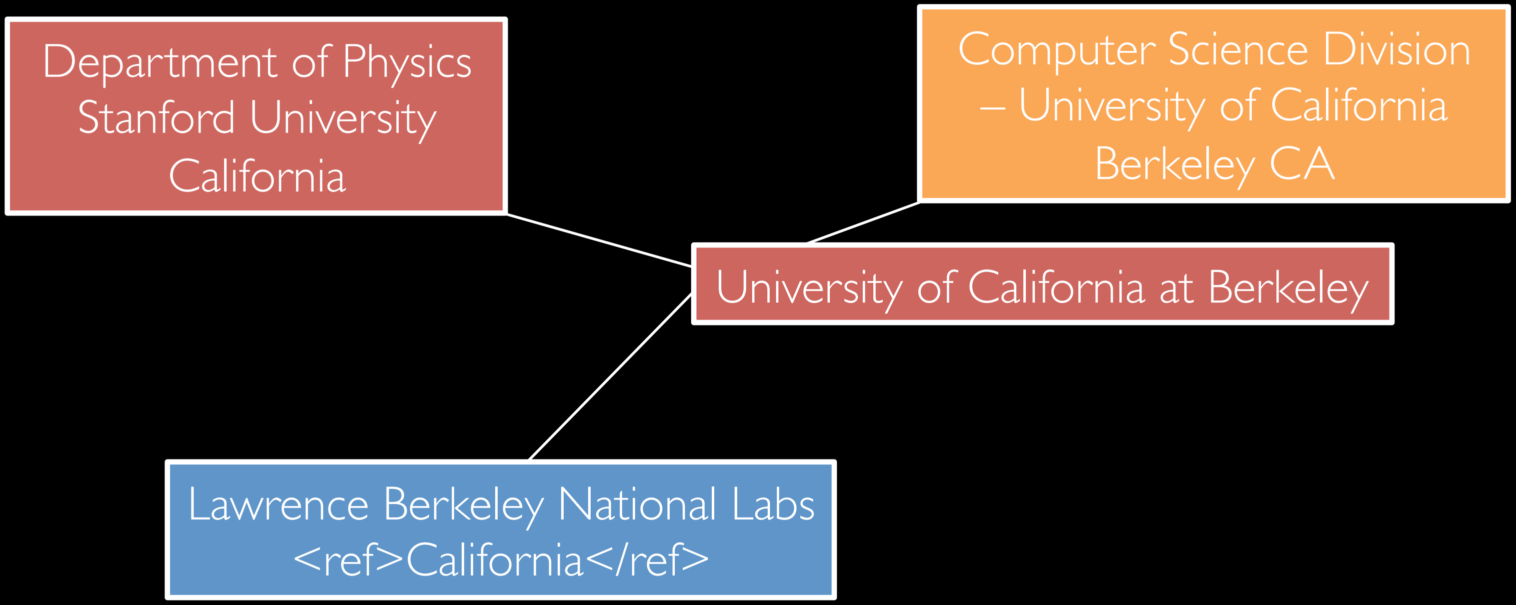
Kannan Ramchandran

Michael I. Jordan

amplab

THE UNIVERSITY OF CALIFORNIA BERKELEY
LET THERE BE LIGHT
1868

Correlation Clustering for Deduplication



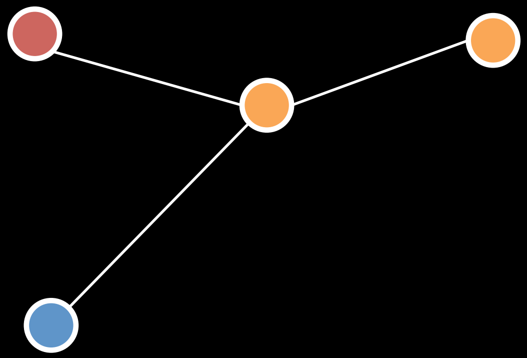
Department of Physics Stanford University California

Computer Science Division – University of California Berkeley CA

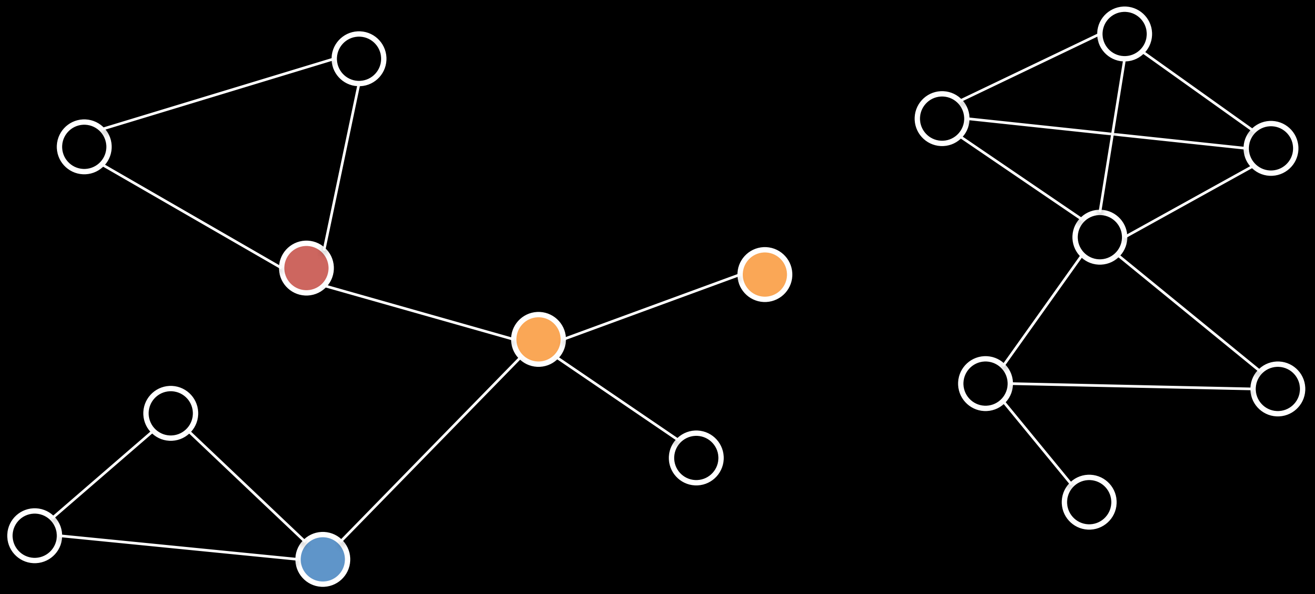
University of California at Berkeley

Lawrence Berkeley National Labs <ref>California</ref>

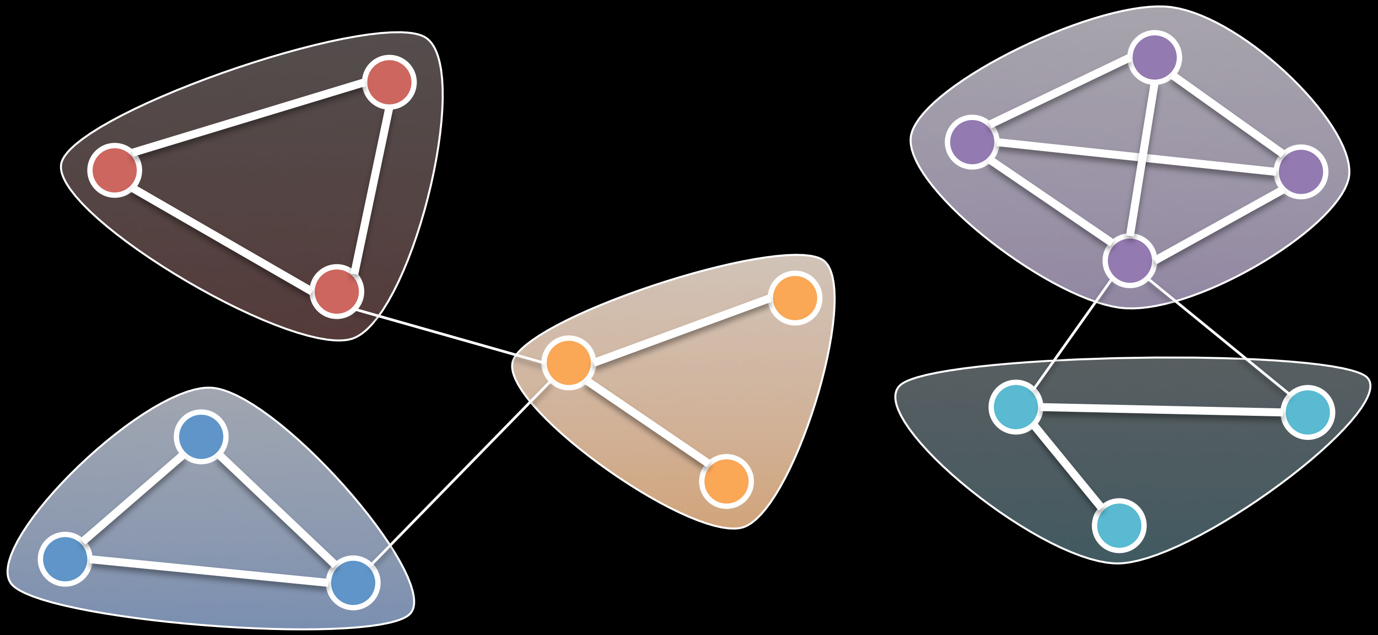
Correlation Clustering for Deduplication



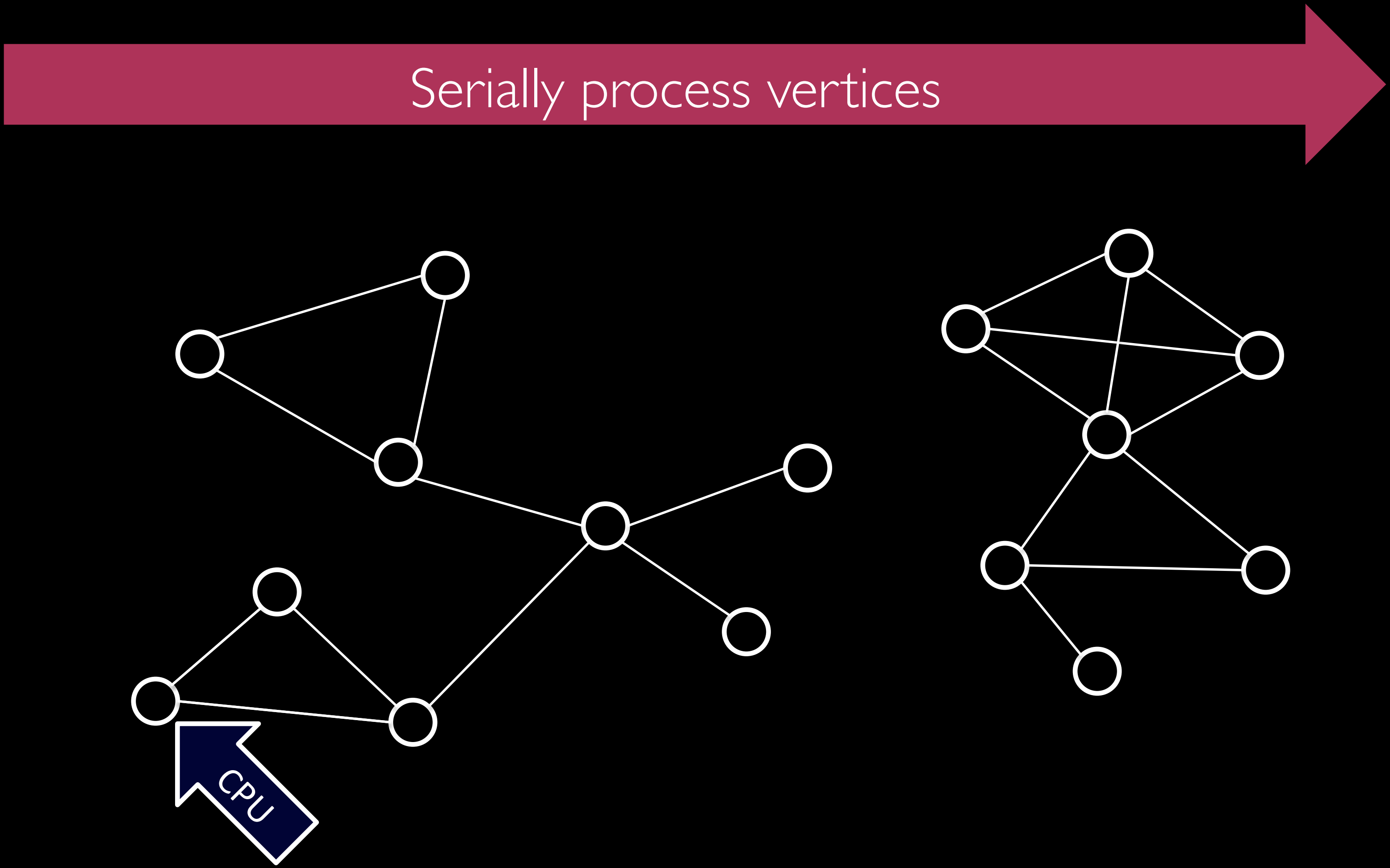
Correlation Clustering for Deduplication



Correlation Clustering for Deduplication



Serial Correlation Clustering

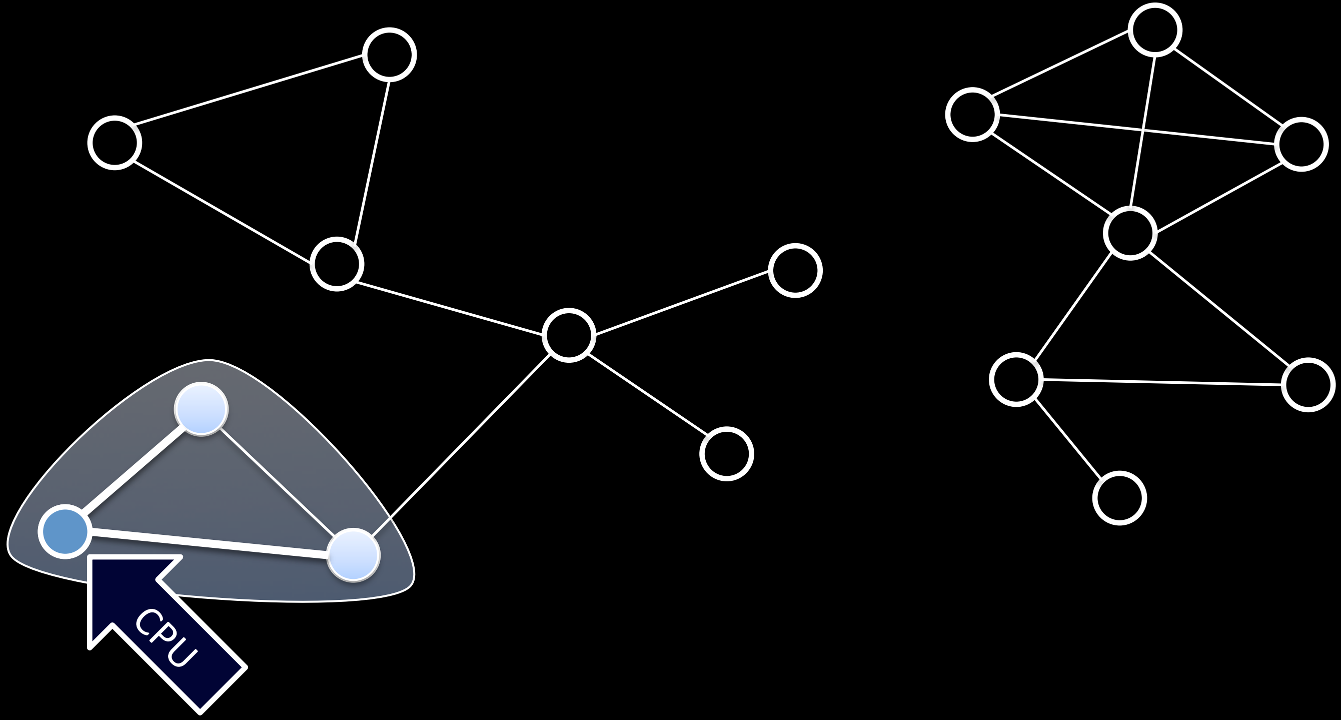


Serially process vertices

Nir Ailon, Moses Charikar, and Alantha Newman.
Aggregating inconsistent information: ranking and clustering.
Journal of the ACM (JACM), 55(5):23, 2008.

Serial Correlation Clustering

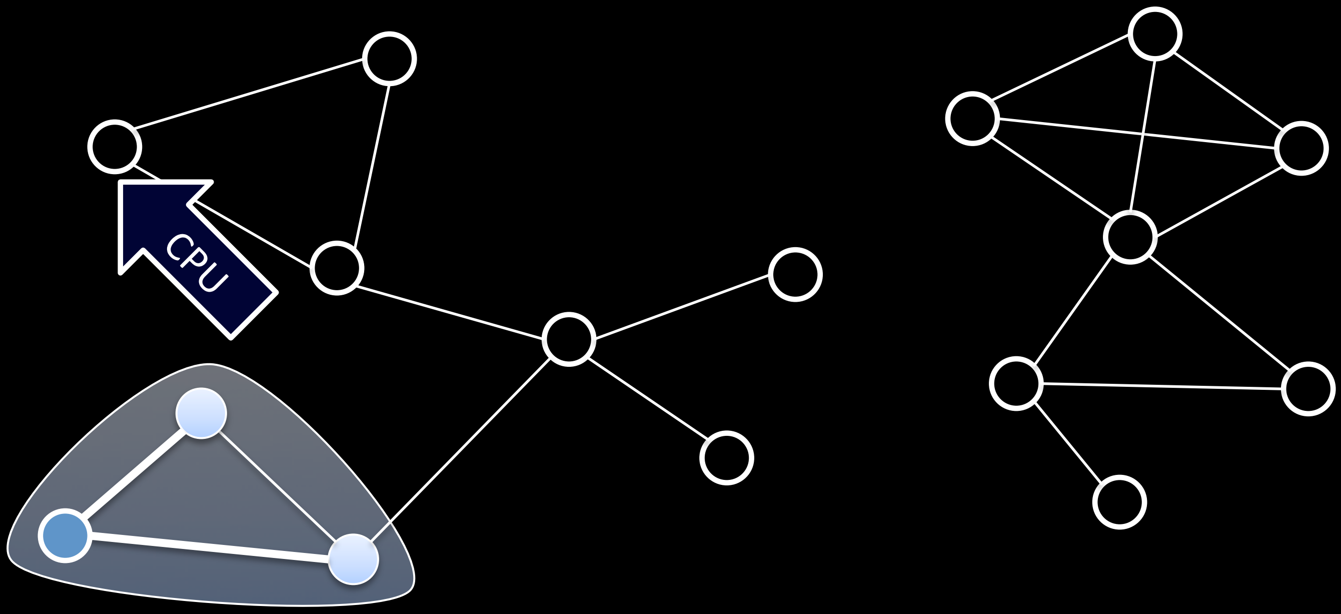
Serially process vertices



Nir Ailon, Moses Charikar, and Alantha Newman.
Aggregating inconsistent information: ranking and clustering.
Journal of the ACM (JACM), 55(5):23, 2008.

Serial Correlation Clustering

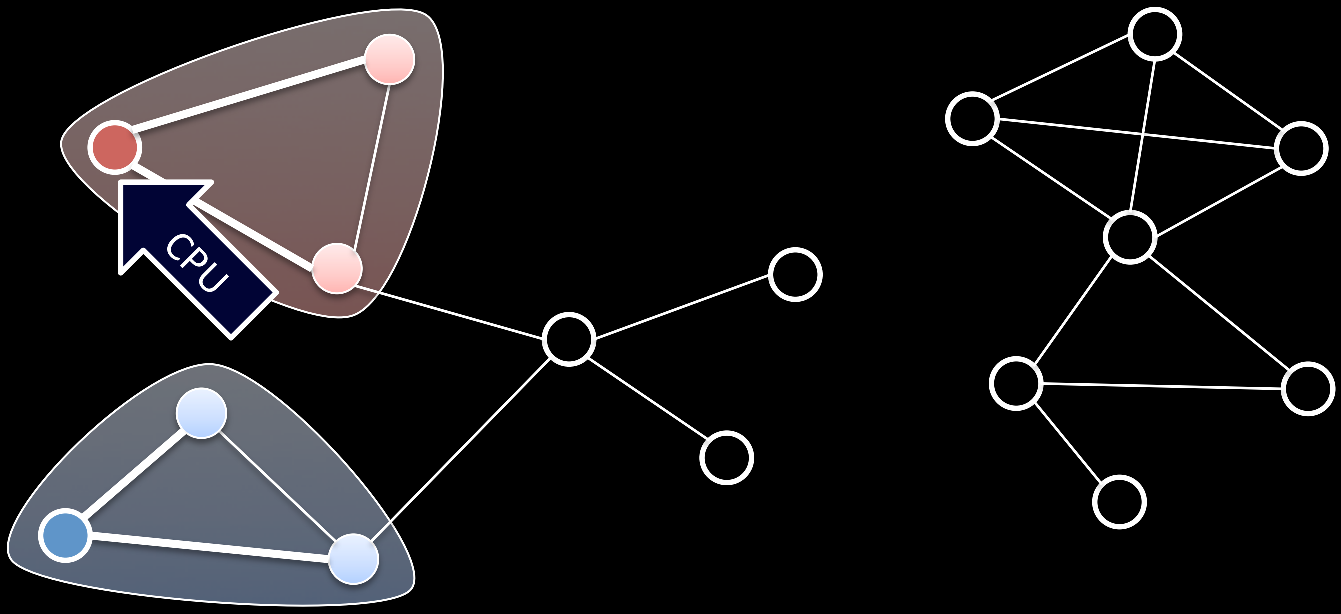
Serially process vertices



Nir Ailon, Moses Charikar, and Alantha Newman.
Aggregating inconsistent information: ranking and clustering.
Journal of the ACM (JACM), 55(5):23, 2008.

Serial Correlation Clustering

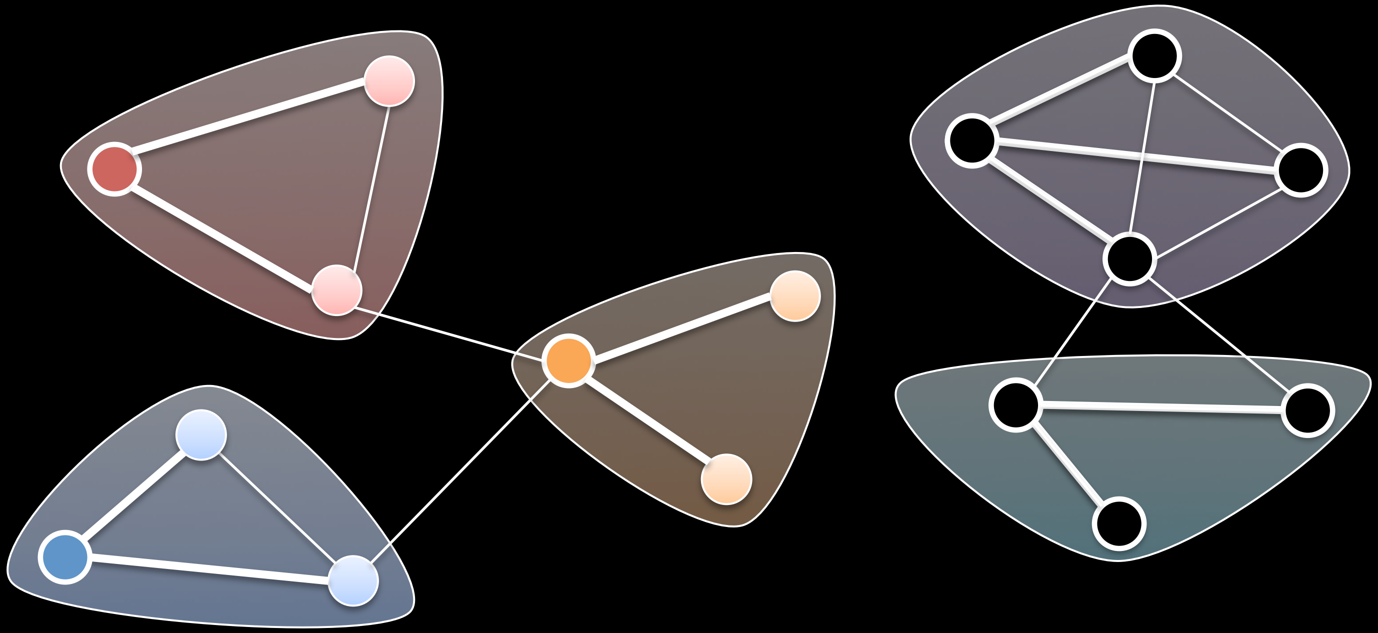
Serially process vertices



Nir Ailon, Moses Charikar, and Alantha Newman.
Aggregating inconsistent information: ranking and clustering.
Journal of the ACM (JACM), 55(5):23, 2008.

Serial Correlation Clustering

Serially process vertices



Approximation 3 OPT (in expectation)

Nir Ailon, Moses Charikar, and Alantha Newman.
Aggregating inconsistent information: ranking and clustering.
Journal of the ACM (JACM), 55(5):23, 2008.

Serial Correlation Clustering

Serially process vertices

Objective: Parallelize **Correlation Clustering**
with strong guarantees and high concurrency.

Idea: Operations on vertices ⇔ Database transaction.
Apply concepts of **concurrency control** from databases.

Approximation 3 OPT (in expectation)

Nir Ailon, Moses Charikar, and Alantha Newman.
Aggregating inconsistent information: ranking and clustering.
Journal of the ACM (JACM), 55(5):23, 2008.

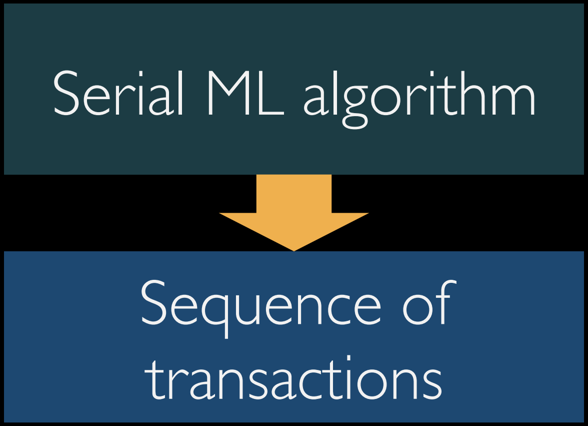
Machine Learning + Concurrency Control

(Xinghao Pan et al.)

Serial ML algorithm

Machine Learning + Concurrency Control

(Xinghao Pan et al.)

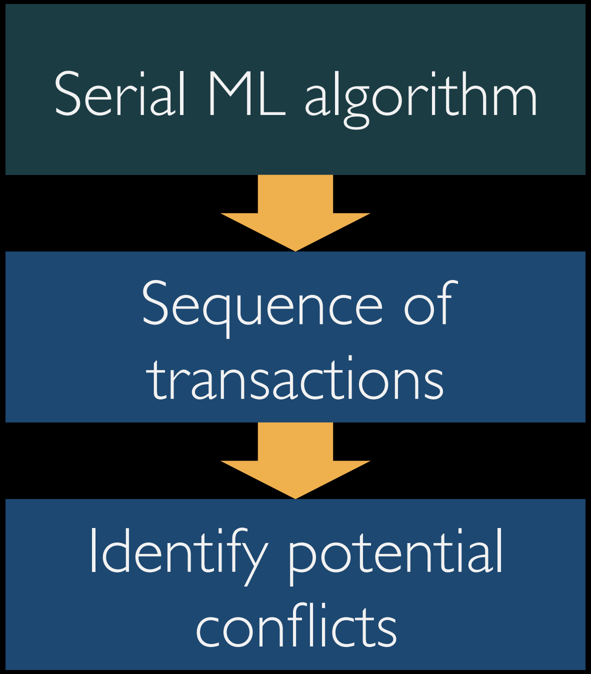


Serial ML algorithm

Sequence of transactions

Machine Learning + Concurrency Control

(Xinghao Pan et al.)



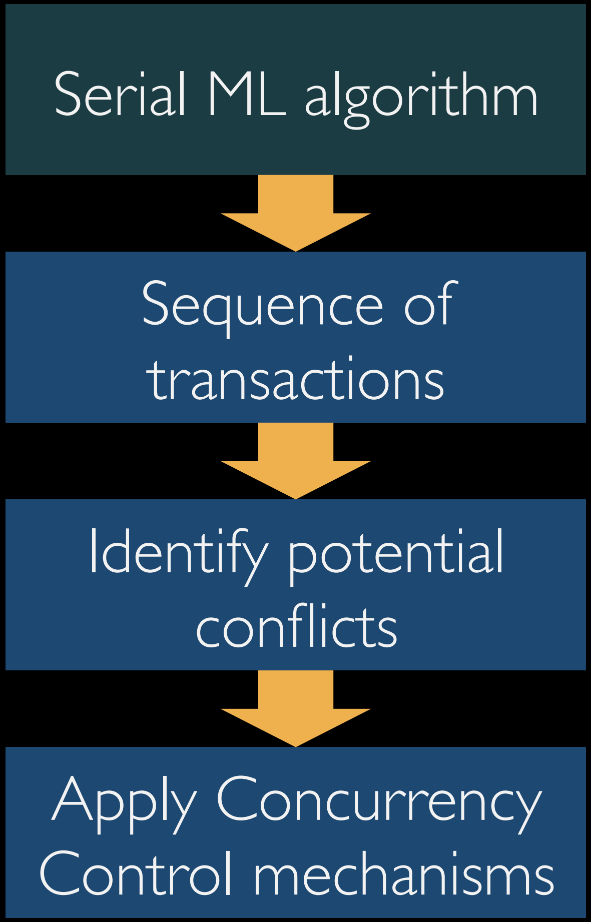
Serial ML algorithm

Sequence of transactions

Identify potential conflicts

Machine Learning + Concurrency Control

(Xinghao Pan et al.)



Serial ML algorithm

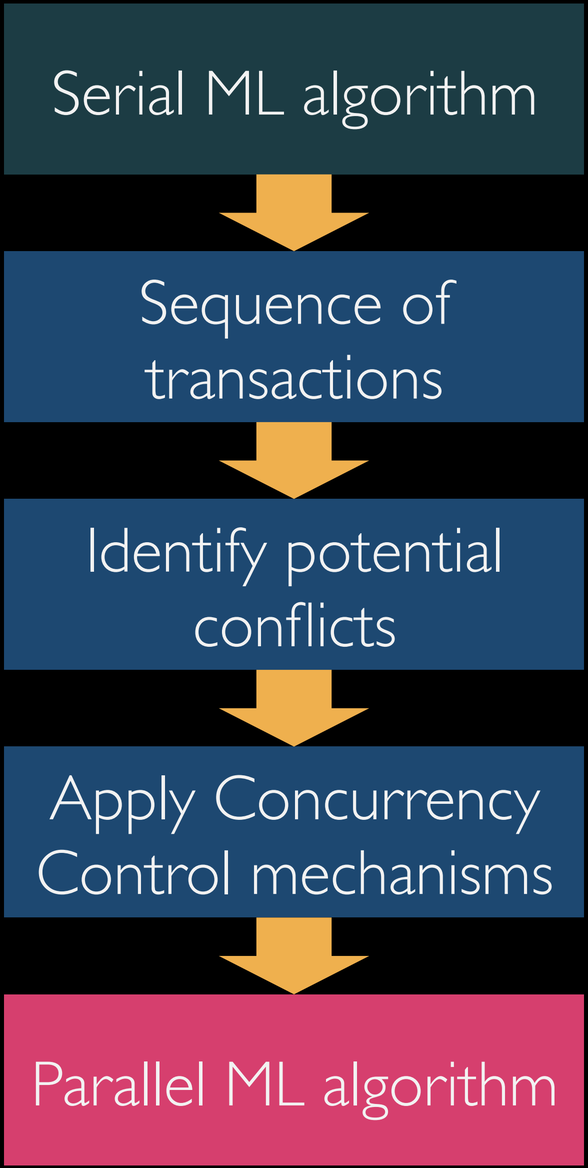
Sequence of transactions

Identify potential conflicts

Apply Concurrency Control mechanisms

Machine Learning + Concurrency Control

(Xinghao Pan et al.)



Serial ML algorithm

Sequence of transactions

Identify potential conflicts

Apply Concurrency Control mechanisms

Parallel ML algorithm

Properties of C4
(Concurrency Control Correlation Clustering)

Correctness

Theorem: C4 is **correct.**

C4 preserves same guarantees as serial algorithm (3 OPT).

Properties of C4
(Concurrency Control Correlation Clustering)

Correctness

Theorem: C4 is **correct.**

C4 preserves same guarantees as serial algorithm (3 OPT).

Concurrency

Theorem: C4 has **small overheads.**

= almost **linear speedup**

Expected #blocked transactions < $2\tau\ |E| / |V|$.

$\tau \equiv$ diff in parallel cpu's progress

Properties of C4
(Concurrency Control Correlation Clustering)

Correctness

Theorem: C4 is **correct.**

C4 preserves same guarantees as serial algorithm (3 OPT).

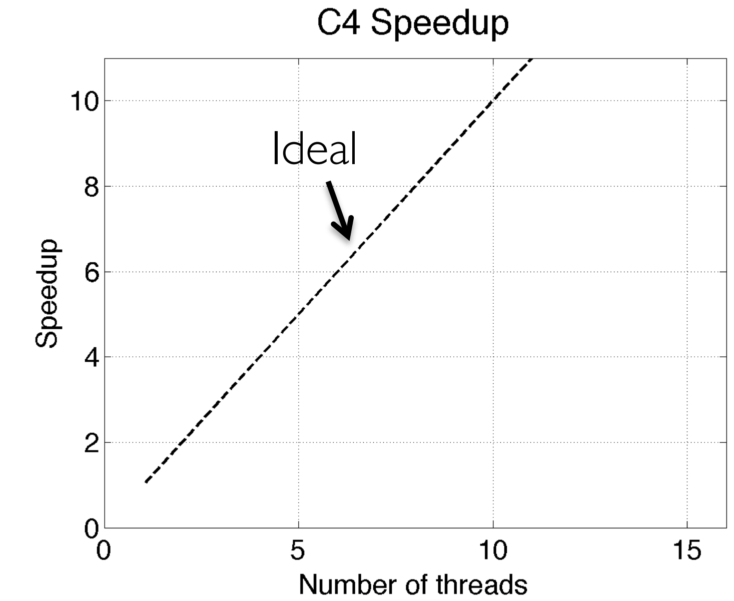
Concurrency

Theorem: C4 has **small overheads.**

= almost **linear speedup**

Expected #blocked transactions < 2τ |*E*| / |*V*|.

τ ≡ diff in parallel cpu's progress



Properties of C4
(Concurrency Control Correlation Clustering)

Correctness

Theorem: C4 is **correct.**

C4 preserves same guarantees as serial algorithm (3 OPT).

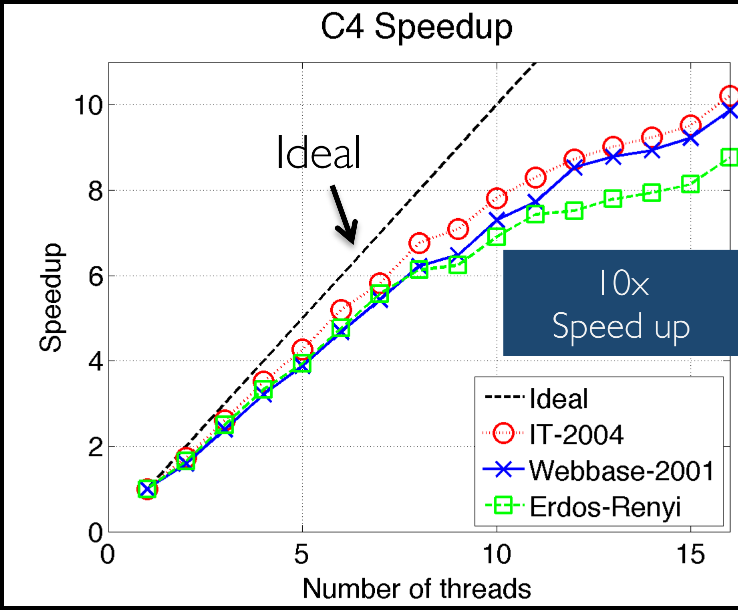
Concurrency

Theorem: C4 has **small overheads.**

= almost **linear speedup**

Expected #blocked transactions < $2\tau\ |E|\ /\ |V|$.

$\tau \equiv$ diff in parallel cpu's progress



10x
Speed up