Adaptive Asynchronous Parallelization of Graph Algorithms

Wenfei Fan¹², Ping Lu², Xiaojian Luo³, Jingbo Xu²,³, Qiang Yin², Wenyuan Yu²,³, Ruiqi Xu³
¹University of Edinburgh ²BDBC, Beihang University ³7 Bridges Ltd.

INTRODUCTION
We propose an Adaptive Asynchronous Parallel (AAP) model for graph computations. As opposed to Bulk Synchronous Parallel (BSP) and Asynchronous Parallel (AP) models, AAP reduces both stragglers and stale computations by dynamically adjusting relative progress of workers.

- BSP, AP and Stale Synchronous Parallel model (SSP) are special cases of AAP.
- Better yet, AAP optimizes parallel processing by adaptively switching among these models at different stages.
- Employing the programming model of GRAPE, AAP parallelizes existing sequential algorithms based on fixpoint computation with partial and incremental evaluation.
- Under a monotone condition, AAP guarantees to converge at correct answers.
- It can optimally simulate MapReduce/PRAM/BSP/AP/SSP.

MOTIVATION

• BSP: stragglers keep other workers idle, thus wasting resources.
• AP: with redundant computation and communication.
• SSP: ad-hoc staleness, incremental improvement over BSP/AP.
• Is it possible to have a simple parallel model that inherits the benefits of BSP and AP, and reduces both stragglers and stale computations, without explicitly switching between the two?

PROGRAMMING MODEL
Data partitioned parallelism (shared-nothing architecture). Fragmented graph G = (G₁, ..., Gₙ), distributed to workers.

GRAPE API: three core functions for a graph query class Q
PEval: a (existing) sequential algorithm for Q, for partial evaluation;
IncEval: a (existing) sequential incremental algorithm for Q;
Assemble: a sequential algorithm (taking a union of partial results).

A fixpoint computation
Φ (R₁, ..., Rₙ)
Rᵢ = PEval(Q, G₁)
Rᵢ₊₁ = IncEval(Q, Rᵢ, Gᵢ, Mᵢ)

• Convergence guarantee:
  - T1: Update parameters take values from a finite domain
  - T2: IncEval is contracting (the same run)
  - T3: IncEval is monotonic (different runs)
T1 + T2: termination
T1 + T2 + T3: the Church-Rosser property (all asynchronous runs converge at the same correct result)

DYNAMIC ADJUSTMENT

\[ DS_i = \begin{cases} +\infty & S(r_i, r_{\min}, r_{\max}) \land (\eta_i = 0) \\ T_{idle}^i - T_{L_i}^i & 1 \leq \eta_i < L_i \\ 0 & \eta_i \geq L_i \end{cases} \]

\[ S(r_i, r_{\min}, r_{\max}) : \text{whether } P_i \text{ should be suspended immediately.} \]
\[ T_{idle}^i : \text{the idle time of worker } P_i \text{ after the last round.} \]

IMPLEMENTATION

PERFORMANCE

Algorithms and their applications:
1. SSSP (traffic analysis);
2. Connected Component (social analysis);
3. PageRank;

CONCLUSION
We have proposed AAP to remedy the limitations of BSP and AP by reducing both stragglers and redundant stale computations. More information about GRAPE: https://7bridges.io