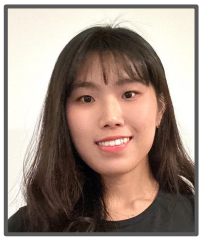
Grudon: A System for Deploying Graph Workloads on Disaggregated Architectures with Near-Data Processing


Vishal Rao


Nikhil Shashidhar


Suyeon Lee

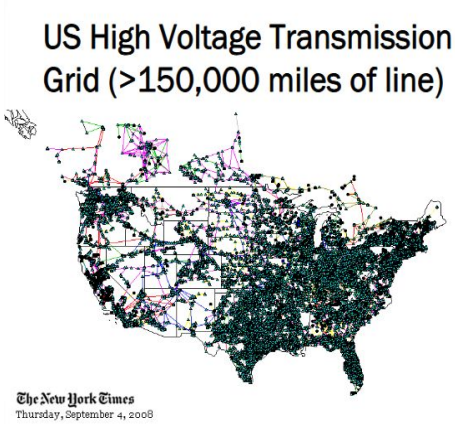

Ada Gavrilovska

Georgia Institute of Technology

Graph Analytics: What's New?

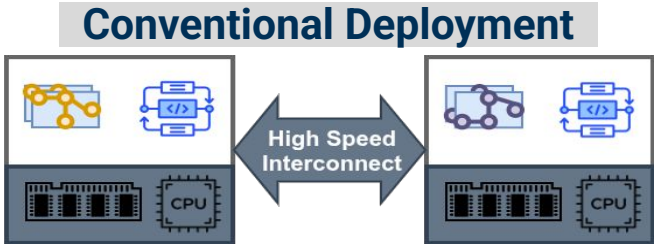
Explosion of interconnected data → Billion-scale graphs

Growing complexity of graphs → High cost/energy demands



Conventional Distributed Graph Runtimes

- Distributed graph and computations across all nodes
- Iterative execution: Traversal Phase + Update Phase



Limitations of Conventional Deployments

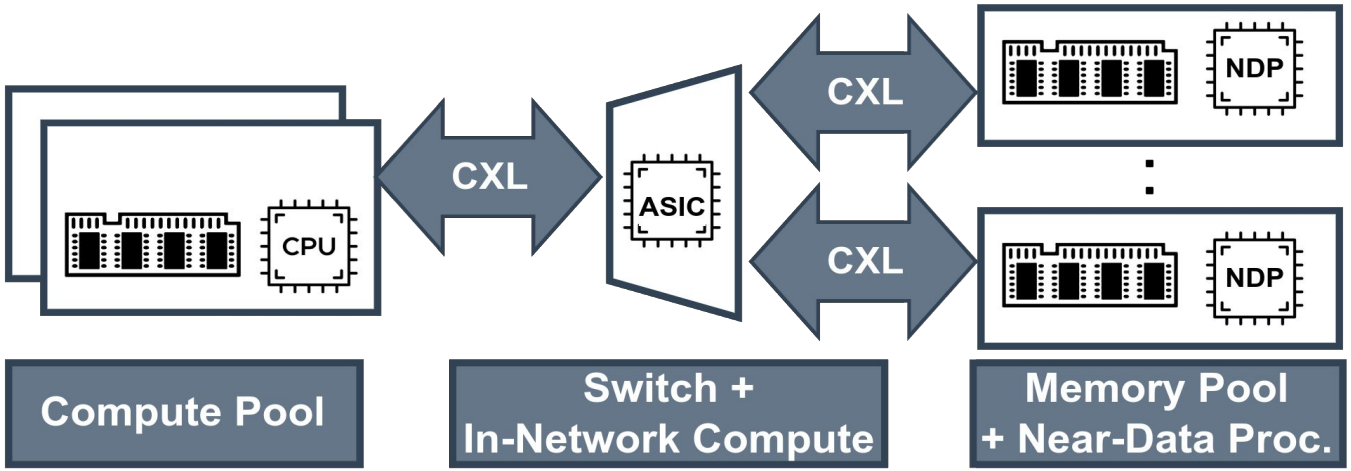
Drawback 1: Limited memory bandwidth

- Traversal and Update phases have different compute and bandwidth needs.
- Conventional servers cannot service the memory bandwidth needs of the traversal phase.

Drawback 2: Lack the ability to flexibly scale resources

- Compute and memory needs of graph workloads can significantly vary
- Conventional servers cannot flexibly scale compute and memory to service the varying needs.

Di-NDP: A New Platform for Graph Analytics?



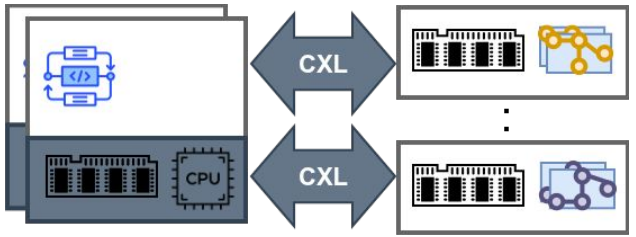
Can DiNDP tackle the limitations of conventional deployments by combining the benefits of memory disaggregation and near-data processing (NDP)?

Di-NDP: Memory Disaggregation for Graphs?

Key Idea: Disaggregation provides flexibility to service the varied compute/memory needs

- Distributed **graph on memory pool** and **computations on hosts**
- Achieves 30% lower energy consumption

Prior Disaggregated Deployments



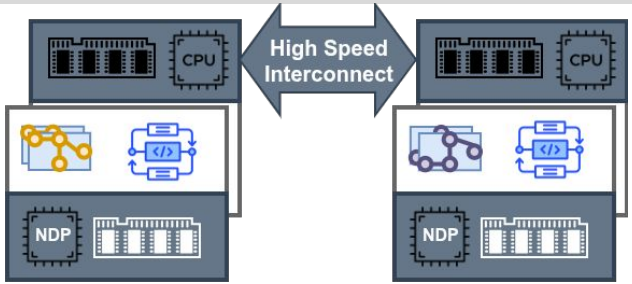
Prior disaggregated solutions provide flexibility BUT incur large data-movement overheads

Di-NDP: Near Data Processing for Graphs?

Key Idea: NDP provides high bandwidth access for bandwidth-intensive graph traversals

- Distributed graph and computations across all **NDP nodes**
- Achieves 3x runtime speedup!
- Consumes lower energy per memory access

Prior NDP-based Deployments



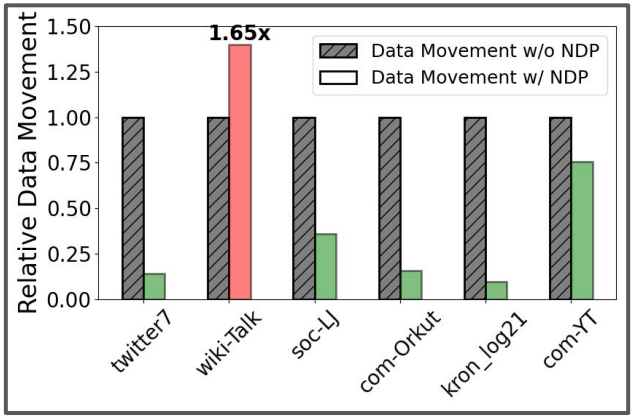
Prior NDP solutions greatly improve performance BUT lack resource flexibility

Design Goals

Trivial NDP-offload of memory-bound traversals does not work!

- Can incur high data-movement overheads

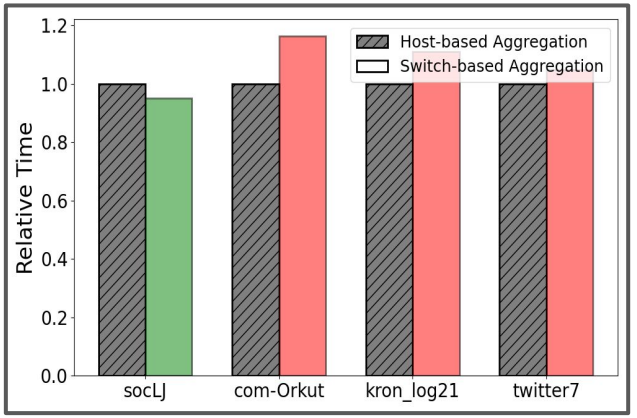
System has to identify scenarios that benefit from NDP offload



Trivially enabling in-network aggregation does not work!

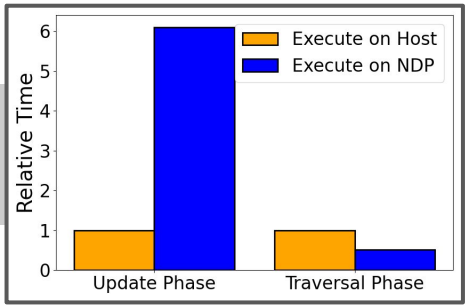
- Low computational power of switches adds runtime overheads

System has to identify scenarios where the computational penalty is outweighed by reduction in data-movement

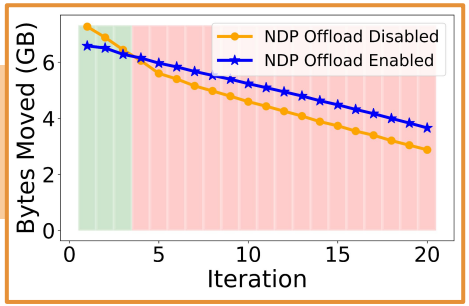


Challenges for Graphs on the DiNDP Platform

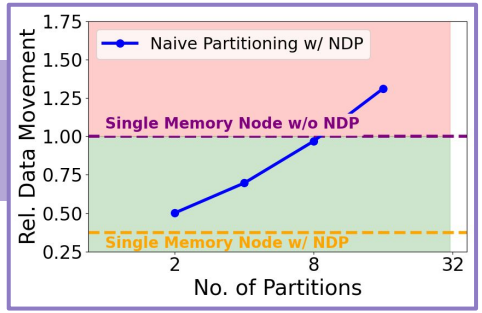
Workload phases have affinity to different components.
Need to separate the workload phases and offload accordingly.



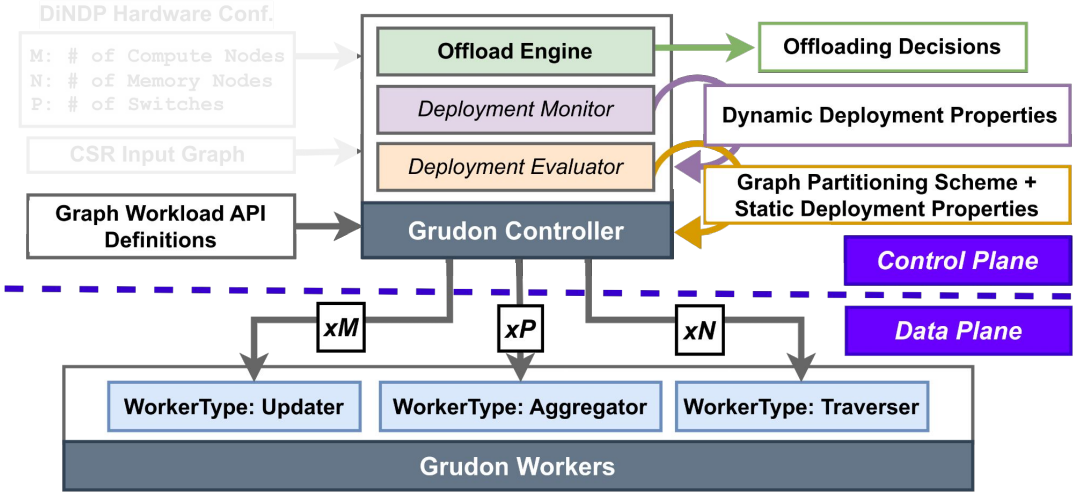
The benefits of NDP offload are dynamic.
Need to monitor the workload and adapt the offload strategy.



Graph distribution overheads can negate offload benefits.
Need to factor overheads of distribution into offload strategy.



Grudon: Deploying Graph Workloads on DiNDP

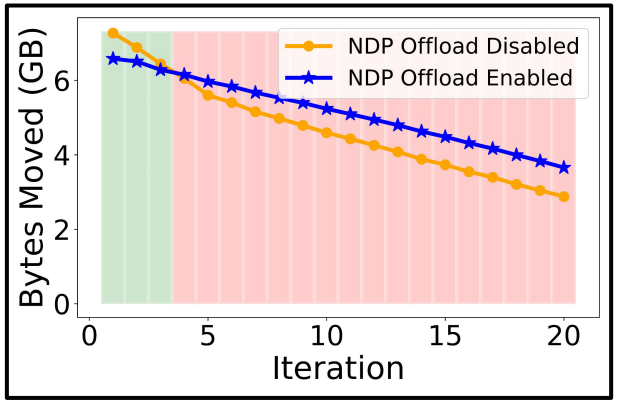


Key Components

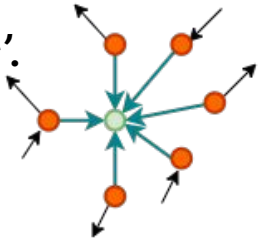
- Grudon API/Programming Model → Enables separation of phases for ease of placement.
- Grudon Deployment Monitor → Monitors workload to inform adaptive offload strategy.
- Grudon Deployment Evaluator → Feeds info about graph distribution to offload engine.

To Offload or Not to Offload?

When do we offload to NDP?



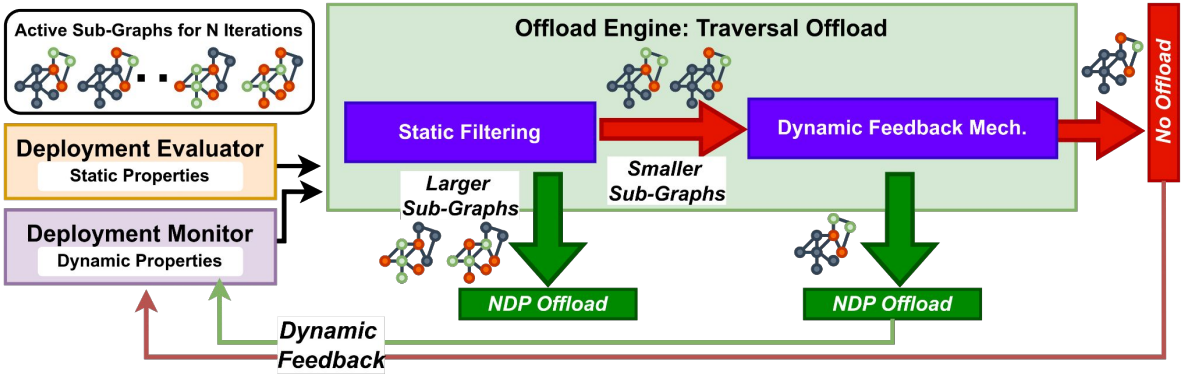
*NDP offload yields performance gains when operating on **Highly-connected 'Hubs'**.*



- Data reduction due to operations concentrating on the hubs.
- Traversing large number of edges benefits from high memory-bandwidth.

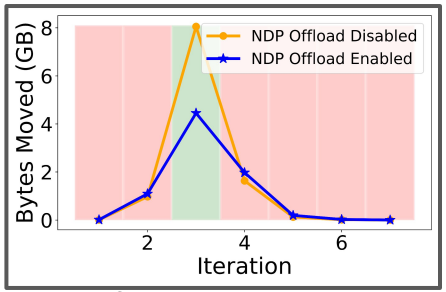
Irregular sub-graph activation and **sudden hub-density spikes** complicate the NDP offload decision**.**

Filtering and Feedback



Why Static Filtering?

Filter out **large sub-graphs** caused by hub-density spikes!



Config: Shortest Path, Twitter7

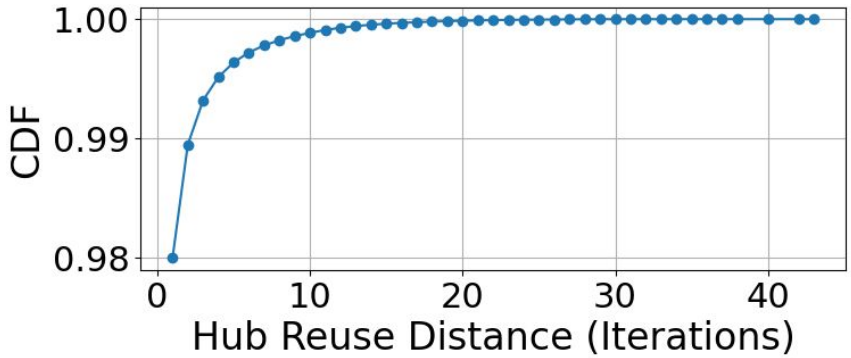
Why Dynamic Feedback?

Hubs exhibit **temporal locality,** can use prior iterations to inform upcoming decision.

How does Feedback work?

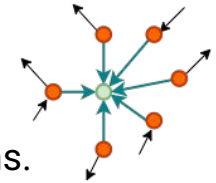
Feedback only activated when filtering is not sufficient.

- Fewer hubs, harder to assess the benefits of NDP offload.
- Smaller sub-graphs exhibit highly irregular activation patterns.



BUT, the hubs in smaller sub-graphs still show temporal locality!

- A hub activated in an iteration is likely to be activated in subsequent iterations.



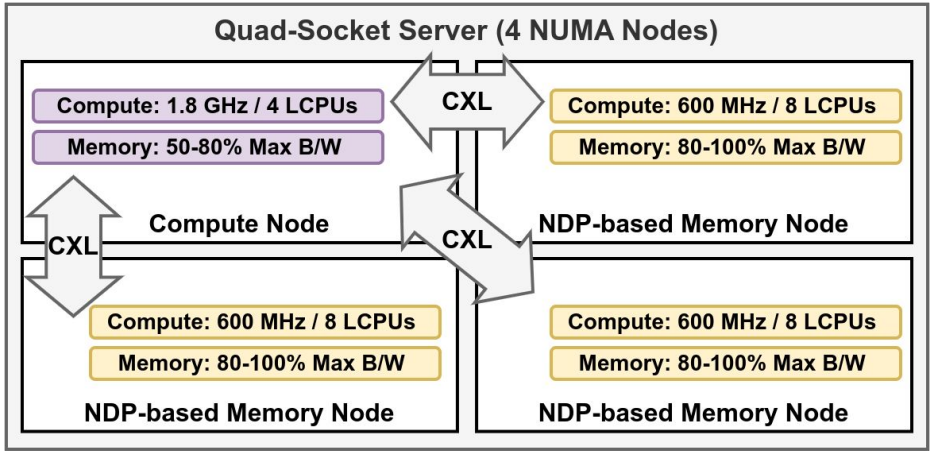
Emulating the DiNDP Platform

Compute Node Configuration –

- Powerful compute, limited parallelism
- Limited memory bandwidth

NDP-based Memory Node Configuration –

- Low-power compute, high parallelism
- High memory bandwidth



4 Socket Server, NUMA Node ≅ DiNDP Node

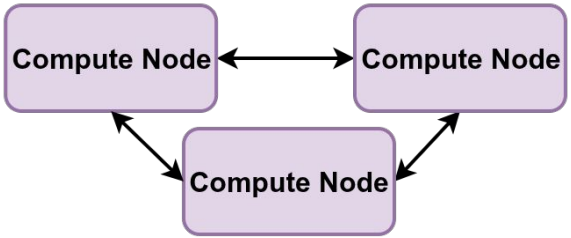
Cross-Socket Access ≅ CXL Access

Evaluation Setup

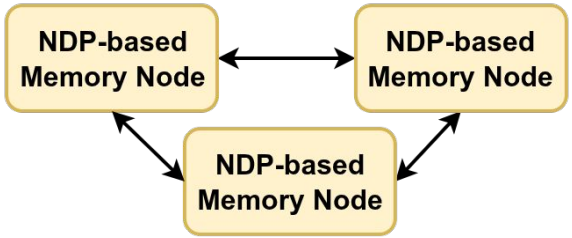
Datasets: 5 graphs — Web-Graphs (LJ, WT), Social-Networks (OR, TW) and Synthetic Kronecker Graph (KR)

Workloads: 3 algorithms — Connected-Components, Shortest Path, PageRank

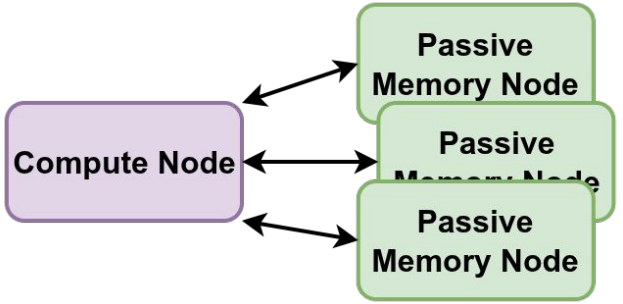
Baselines:



Gluon[1]: Vanilla distributed graph runtime.



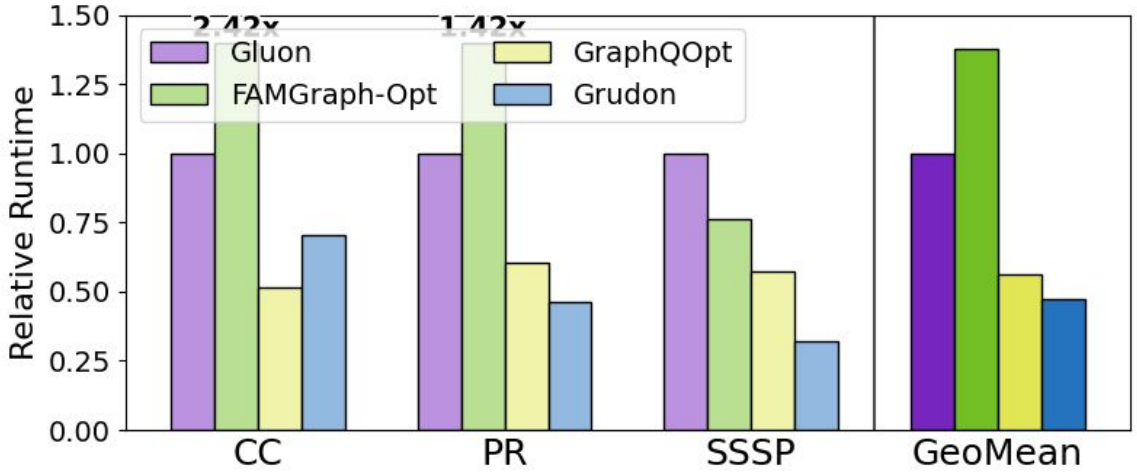
GraphQOpt: Distributed NDP-baseline (based on GraphQ)[2]



GraphQOpt: Disaggregated baseline (based on FAM-Graph)[3]

[1] Dathathri et al. "Gluon: A communication-optimizing substrate for distributed heterogeneous graph analytics." Proceedings of the 39th ACM PLDI. 2018.
[2] Y. Zhuo et al., "Graphq: Scalable pim-based graph processing," in Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture, 2019.
[3] D. Zahka et al., "Fam-graph: Graph analytics on disaggregated memory," in 2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2022.

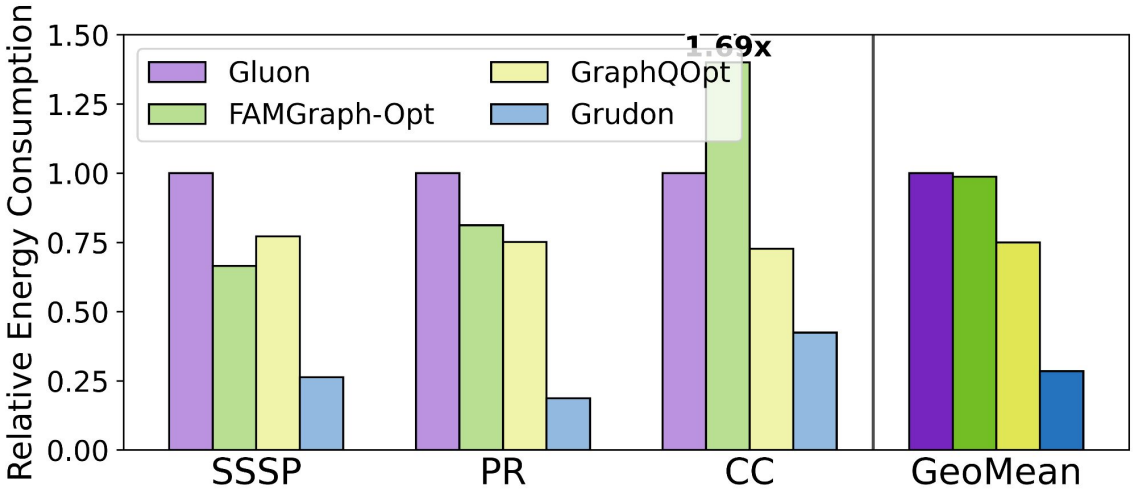
Does Grudon achieve a Runtime Speedup?



Key Takeaways

- Grudon is the best performing deployment overall.
- NDP-based deployment does better for CC because the workload is primarily memory-bound.
- Grudon effectively utilizes the near-data acceleration capabilities.

Does Grudon decrease Energy Demands?



Key Takeaways

Grudon combines the benefits of NDP and Disaggregation.

- NDP: Low-power memory accesses
- Disaggregation: Reduces energy demands by avoiding resource over-provisioning.

Qualitative Comparison of Deployments

Performance Metrics	Conventional Deployments	NDP-Based Deployments	Disaggregated Deployments	Grudon (DiNDP)
Flexible Resource Scaling				
Data Movement Overheads				
Energy Demands				

The DiNDP platform can be an effective alternative for Distributed Graph Analytics

Conclusion

Proposal

A new paradigm for deploying graph workloads that provides flexibility and high memory-bandwidth processing capabilities needed to effectively scale the workload.

Deployment Insight: Offload with Care, or Beware!

Grudon highlights the potential of the DiNDP platform as a cost-effective and energy-efficient alternative for distributed graph deployments.