Integrated Approach of Task Scheduling and Domain Decomposition in Particle Transport Computation

Ping (Ann) An
Dr. Nancy Amato

Parasol
Smarter computing.
Texas A&M University

Outline

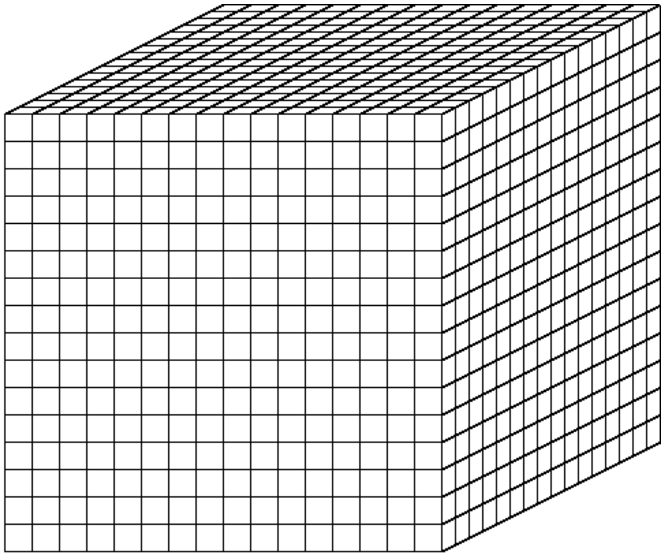
- Parallelizing Transport Sweeps
- Overview of the structure of TAXI_Scheduler
- Using Performance Modeling to Guide Optimizations for Partitioning and Ordering
- Conclusions and Future work

Transport Sweeps

Parasol

- To find the position, velocity and energy of particles traveling in a physical domain

- Discrete ordinates particle transport algorithm
  - Discretise the physical domain (grids), the angle domain (angle sets), and energy domain (energy groups).



Challenges:

- Parallelization of multiple independent sweeps through the spatial grid
- The data is too large to be replicated

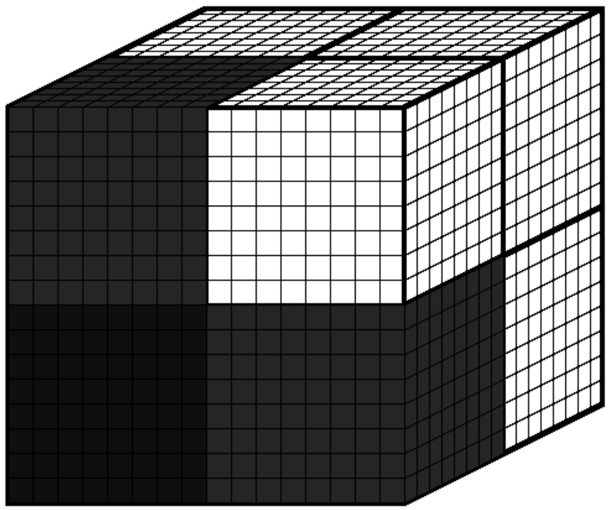
Traditional Approach

Parasol

Traditional approach: Domain Decomposition (DD)

• Physical domain is partitioned into sub domains for parallel processing

• Spatially close data have stronger dependencies on each other than on data farther away



Problems:

Standard DD techniques are not sufficient

• Considers only data partitioning

• The partitioning maybe good for sweep through a certain direction, but won't be good for all the sweeps.

Task Scheduling Approach

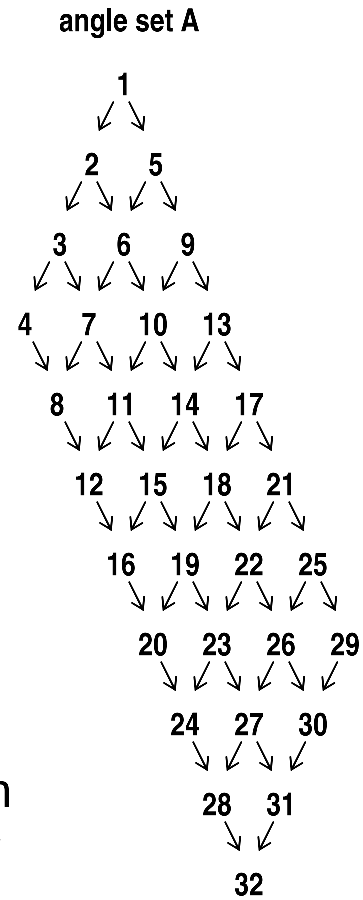
An alternative approach is Task Scheduling (TS)

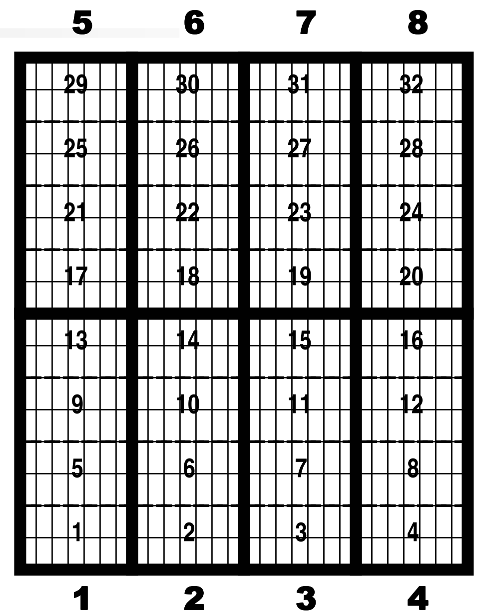
- Computations are partitioned into tasks

- Sweep dependences among tasks creates Task Dependency Graph (TDG)

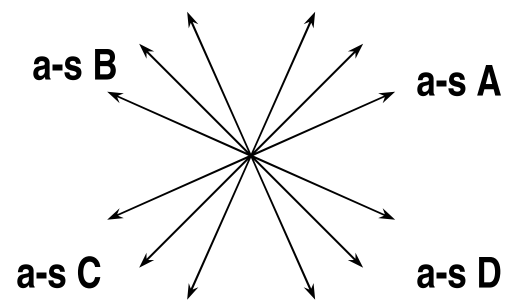
Problems:

- Only considers computation partitioning
- A general Task scheduling algorithm cannot handle multiple TDGs working on the same set of data concurrently

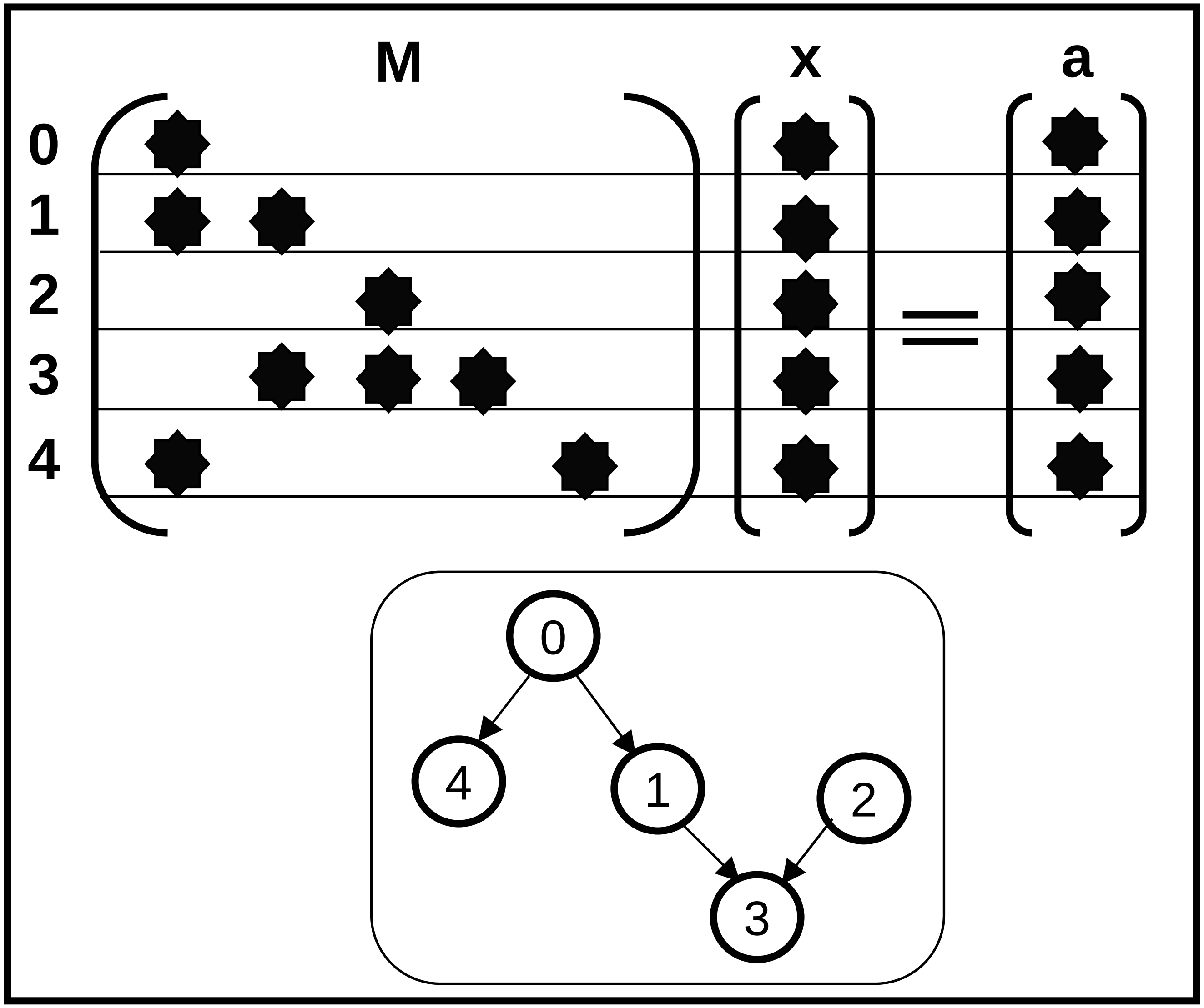




direction
(or “angle”)

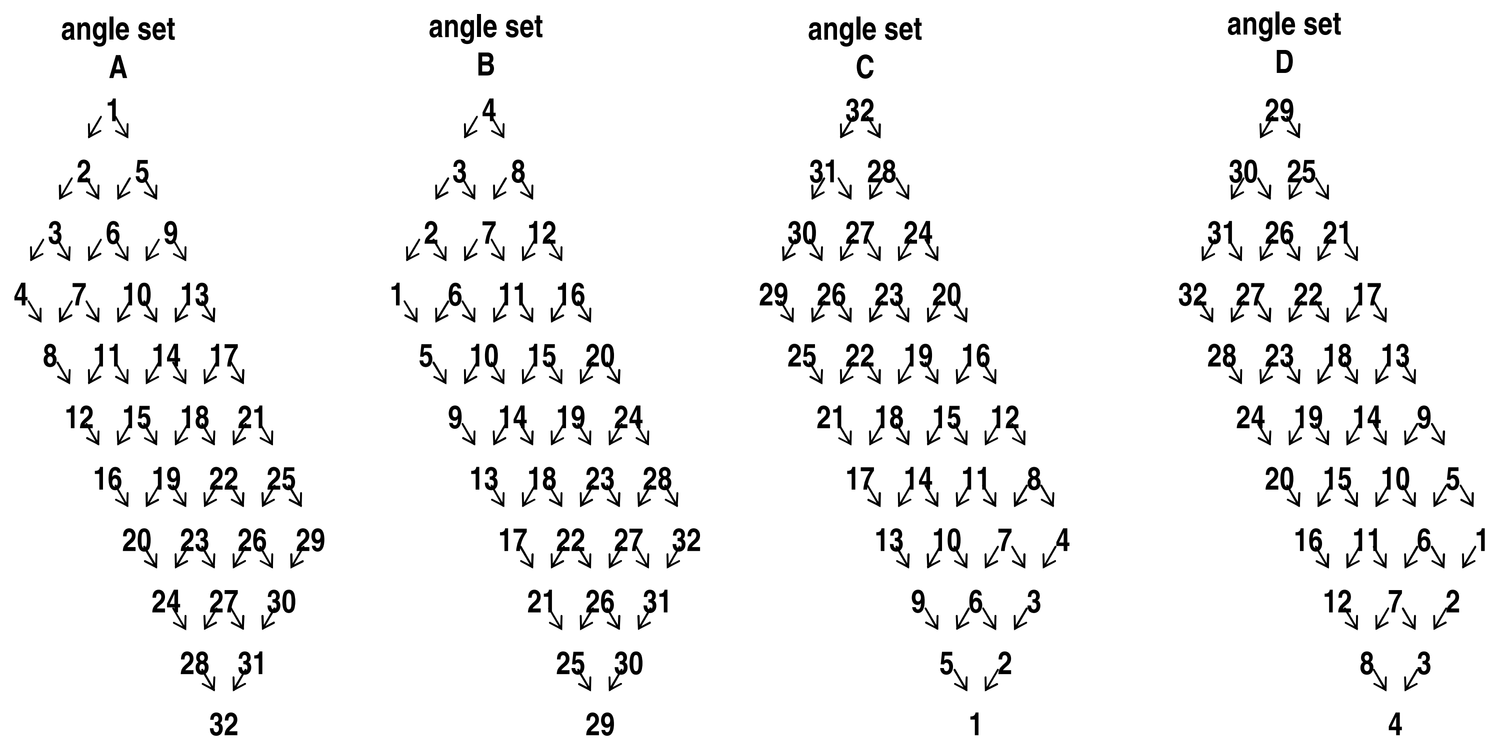


Example



Integrated approach of Task Scheduling and Domain Decomposition

- Considers both computation partitioning and data partitioning
- Partitioning of multiple TDGs and data simultaneously
  - Partitioning of the TDGs is critical to the parallelization
  - Ordering of multiple ready tasks in the same processor is also important



Partitioning and Scheduling Optimizations

High-Level Optimizations
(*More or less machine independent*)

– Task Scheduling to partition the computation

– Domain Decomposition to partition the data

– Match computation and data partition

– Task ordering

Outline

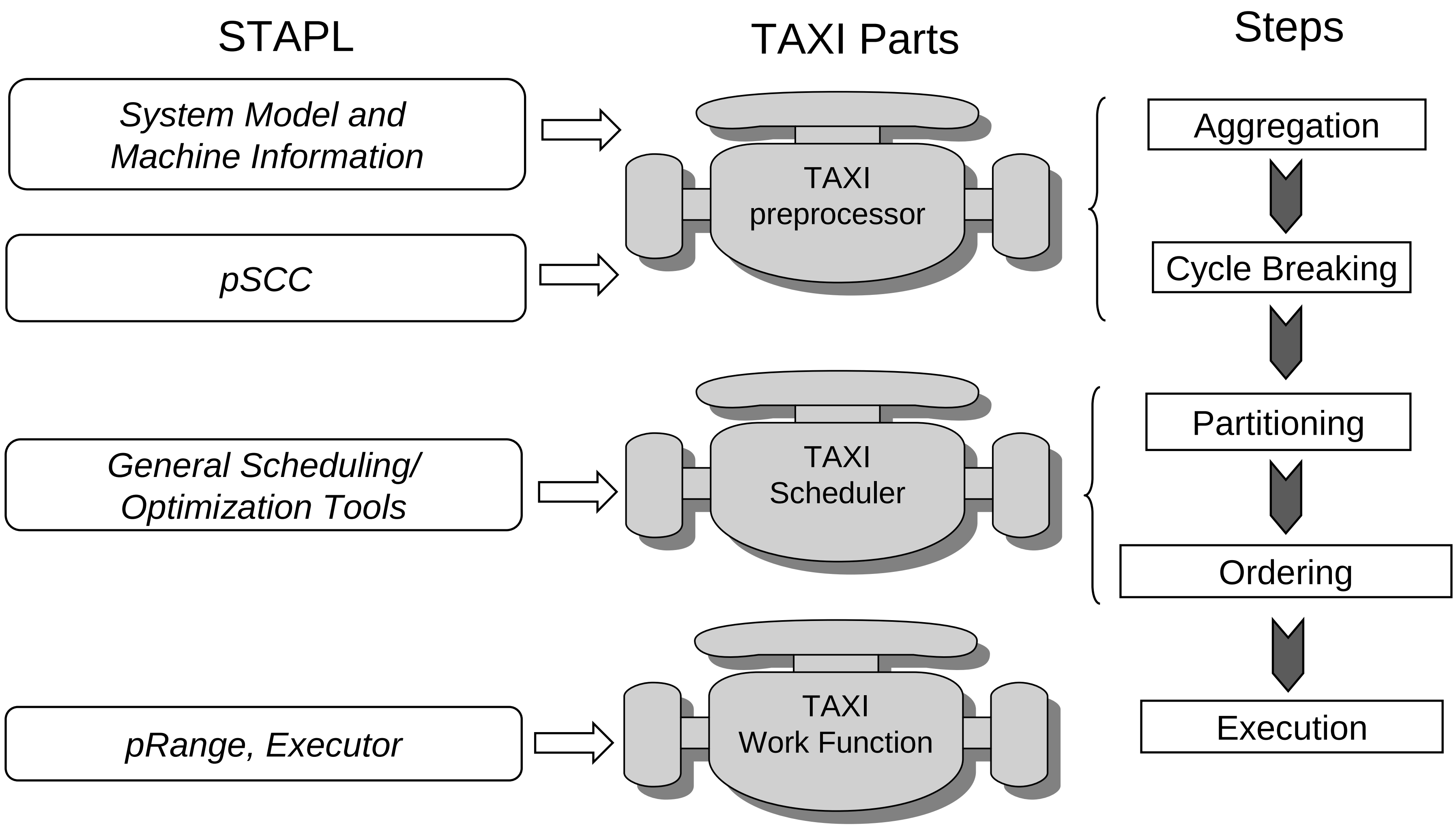
- Parallelizing Transport Sweeps
- Overview of the structure of TAXI_Scheduler
- Using Performance Modeling to Guide Optimizations for Partitioning and Ordering
- Conclusions and Future work

Structural Overview

- TAXI Scheduler
  - specialized partitioning algorithms
  - general scheduling algorithms provided by STAPL
- Object Oriented design
  - partitioning, scheduling/ordering, aggregation,
  - Easy to plug in external modules
- Sweep Simulator
  - Simulates both computation and communication
  - Evaluate different partitioning/ordering schemes

Structural Overview

Parasol

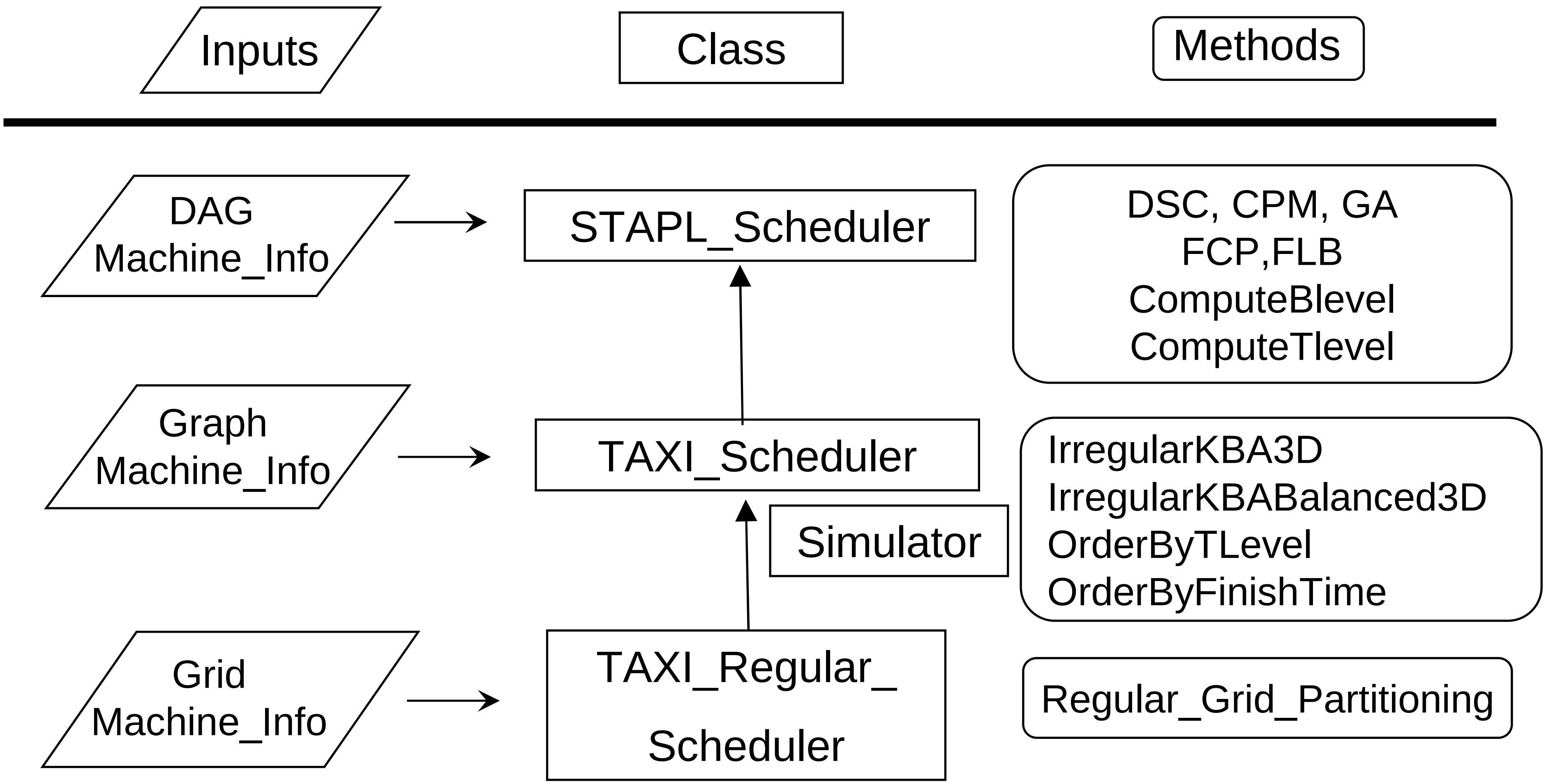


TAXI_Scheduler

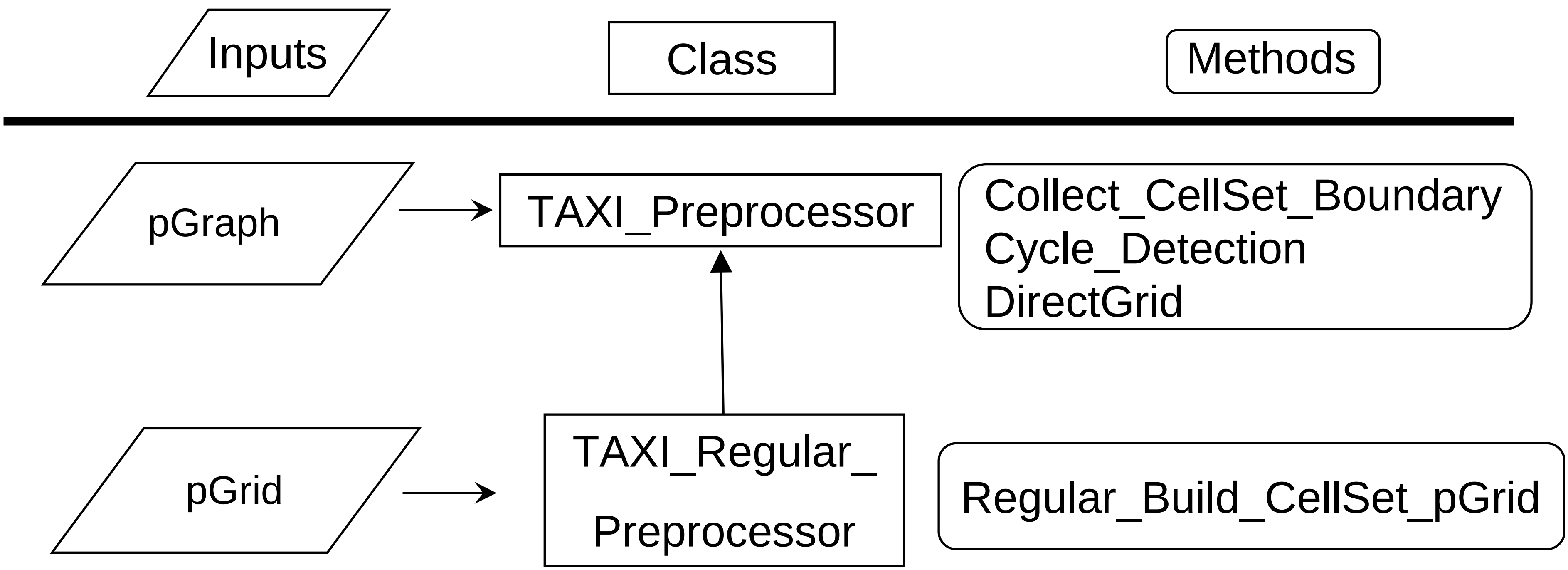
- Specialized partitioning algorithms designed for discrete ordinates particle transport problem
  – KBA, Hybrid, Volumetric etc. for regular grids
  – KBA projection, metis for irregular grids

- STAPL optimization tools can be applied in the partitioning processor
  – High level machine independent optimization
    Task scheduling algorithms for partitioning and ordering
    - Dominant Sequence Clustering
    - Critical Path Merge
    - Scheduling using Genetic Algorithm
  – Low level machine dependent optimization
    - To optimize resource contention
    - To optimize computation and communication granularity

OO Design of TAXI_Scheduler



OO Design of TAXI_Preprocessor



```
template<class TASKWEIGTH, class COMMWEIGHT>
class GeneralScheduler {
  //Partitioning algorithms
     //UNC algorithms
            Schedule DominantSequenceClustering();
            Schedule CriticalPathMerge();
            Schedule HeavyEdgeMerge();
     //BNP algorithm
            Schedule FastCriticalPath();
  //Mapping algorithms
            Schedule GA_Mapping();
            Schedule MapbyLLB();
  //Scheduling algorithm
            Schedule ScheduleByTLevel();
            Schedule ScheduleByBLevel();
```

Outline

- Parallelizing Transport Sweeps
- Overview of the structure of TAXI_Partitioner/Scheduler
- Using Performance Modeling to Guide Optimizations for Partitioning and Scheduling/Ordering
- Conclusions and Future work

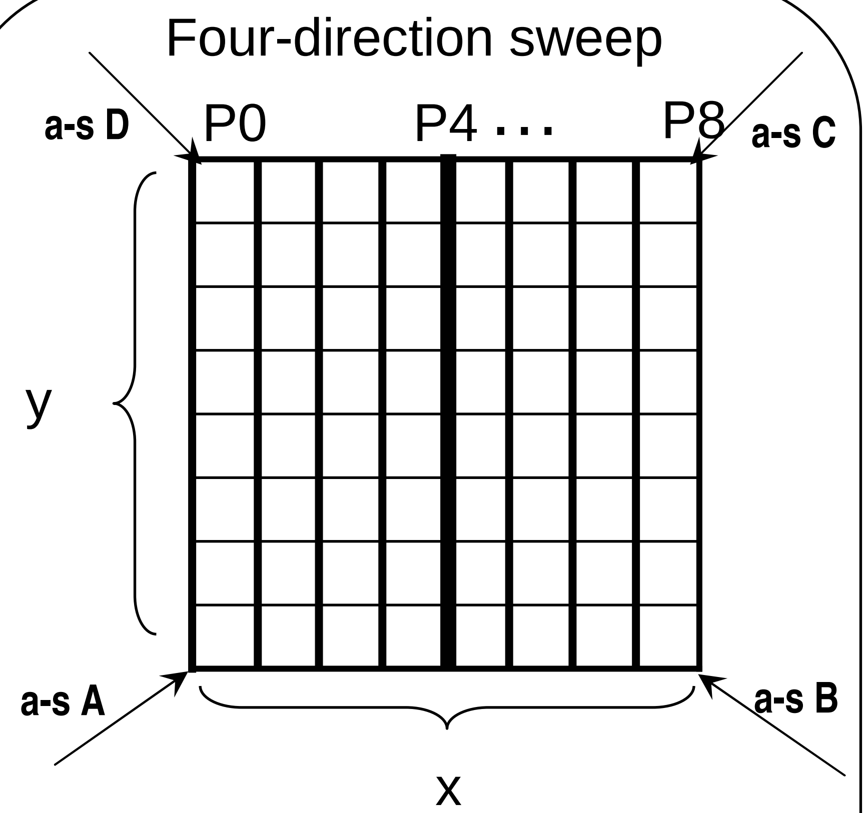
Performance Modelling

Parasol

- Predict the running time
- The model can be used to evaluate and compare different partitioning and ordering approaches
  - To identify the critical performance issues that need to be optimized
- Our objective is to model the general problem
  - We start from analytical performance model of regular grids
    - Processor and network contention

Analytical Study of Regular Grids Processor Contention

- Example of an analytical model to help us to detect the problem
- KBA partitioning scheme
- Contention on processor P4 added extra terms into T_total, the parallel running time

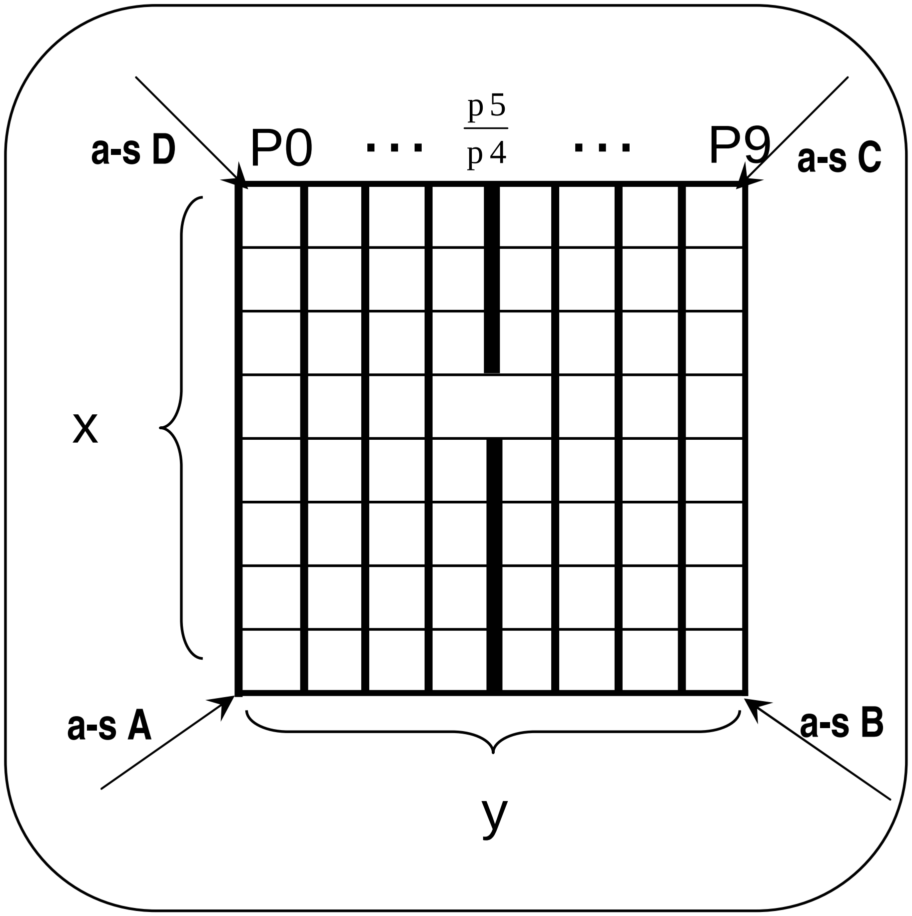


T_total = (y + x - 1)*Cp + (x-1)*Cm + y*Cp + y*Cp + y*Cp

Cp: computation cost

Cm: communication cost

Partitioning

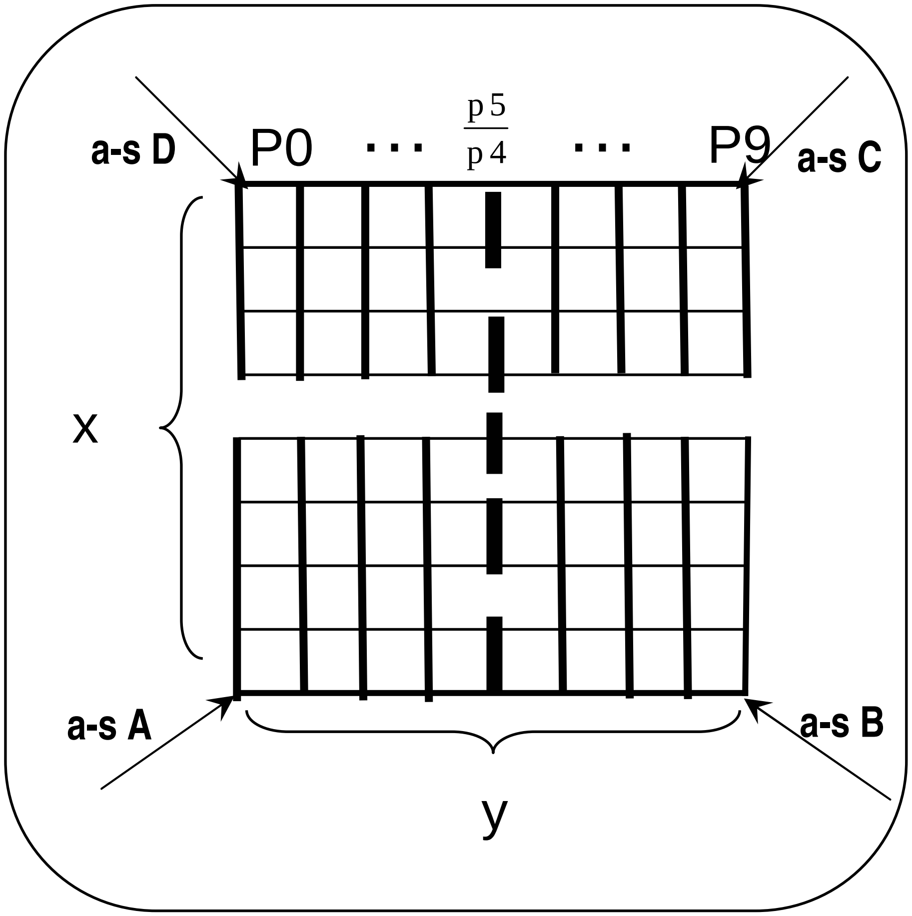


Add more processors to the middle column

Cp = Cm = 1; x = y = 9; x_L = 5;

T_total	KBA (compute & simulate)	One more processor (simulate)
Direction A	25	26
Directions A-B	34	34
Directions A-D	34	34
Directions A-B-C-D	52	48

Partitioning

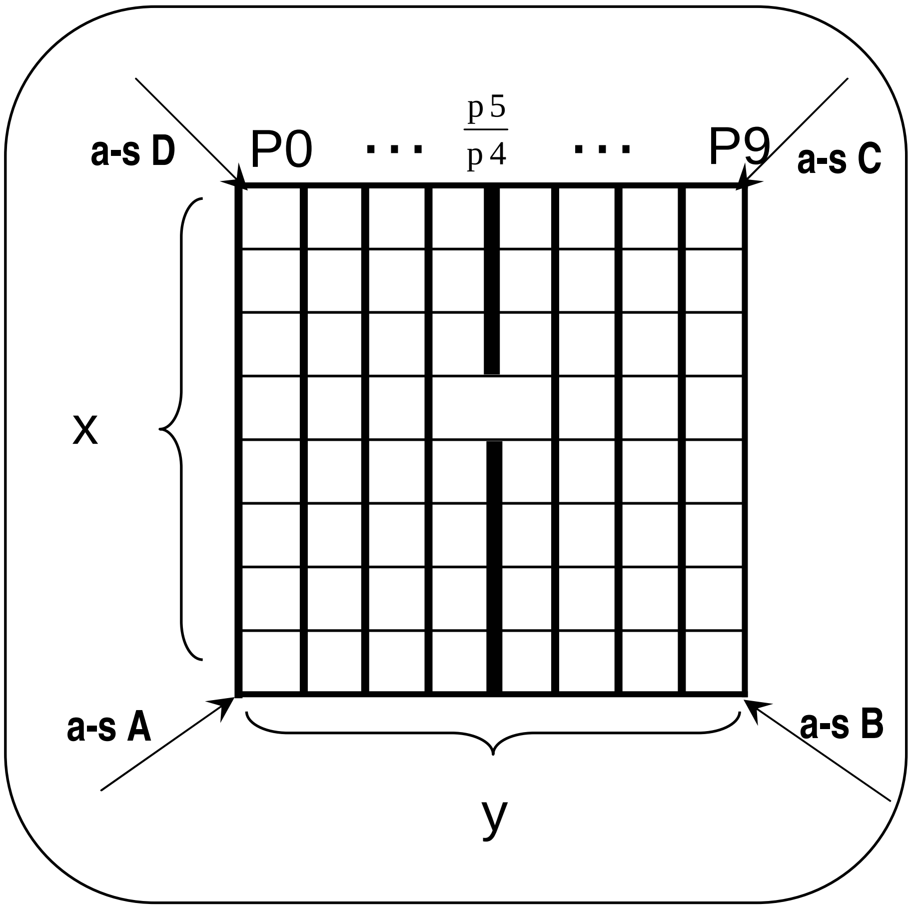


Hybrid method w/ more processor in the middle column

Cp = Cm = 1; x = y = 9; x_L = 5;

T_total	One more processor (simulate)	Modified Hybrid (simulate)
Direction A	26	29
Directions A-B	34	34
Directions A-D	34	34
Directions A-B-C-D	48	36

Ordering Schemes



Different ordering scheme affects running the time too.

Cp = Cm = 1; x = y = 9; x_L = 5;

T_total	Fixed Consistent Ordering	Ordering by Level (simulate)
Direction A	26	26
Directions A-B	31	34
Directions A-D	34	34
Directions A-B-C-D	45(?)	48

Conclusions

Parasol

- Integrated task scheduling and domain decomposition approach is the appropriate approach for particle transport computation
- It is found that the contention problem is the critical part in the sweep
- The computation can be optimized by solving the contention problem properly
- The ordering strategies also affects the performance

Future Work

Parasol

Extend the work to arbitrary grids

- Partitioning optimization
  - We are working on different partitioning schemes
  - Also genetic scheduling algorithm
- Ordering optimization
  - By finish time in a single direction sweep
  - Incorporate other researcher’s heuristics