Making Contention Scheduling Aware of Identical Data

Oliver Sinnen
With Croydon Dias

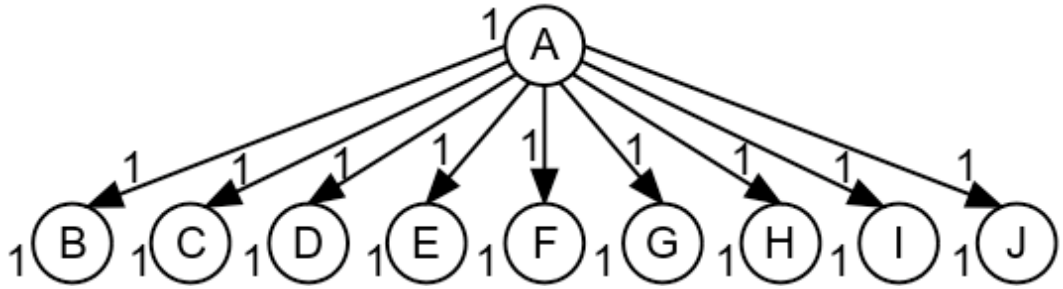
Department of Electrical and Computer Engineering
University of Auckland
New Zealand

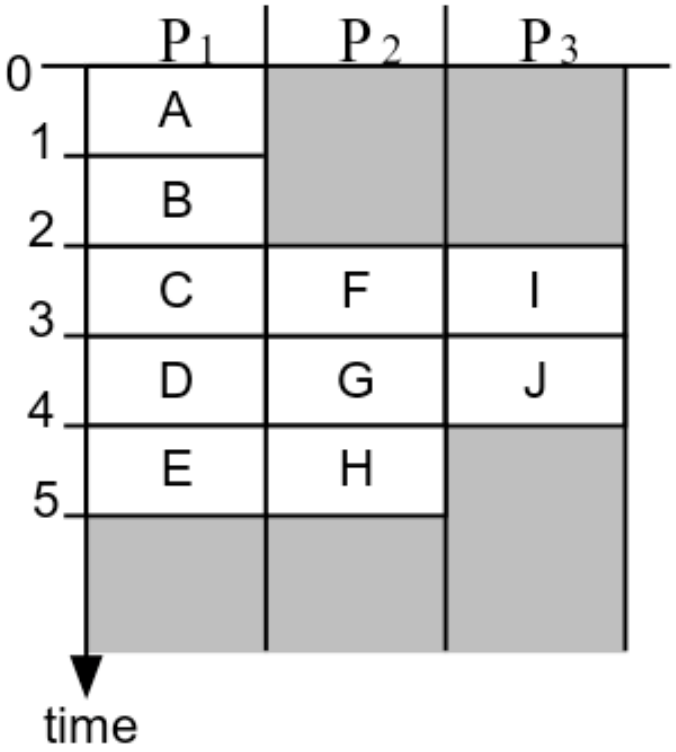
www.ece.auckland.ac.nz/~sinnen/

Oliver Sinnen – o.sinnen@auckland.ac.nz

How to schedule tasks with identical input data ?

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

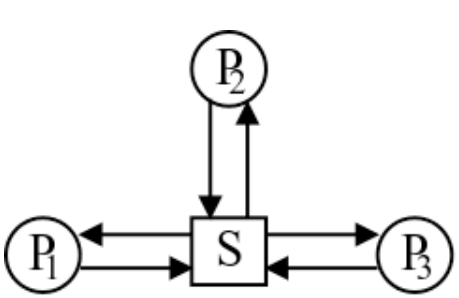


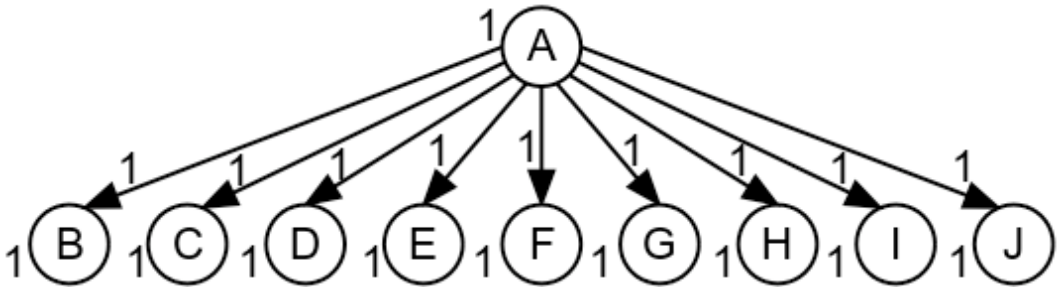


Oliver Sinnen – o.sinnen@auckland.ac.nz

How to schedule tasks with identical input data ?

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau





	L_{in}	P_1	L_{out}	L_{in}	P_2	L_{out}	L_{in}	P_3	L_{out}
0		A							
1		B	*A->G*	*A->G*					
2		C	*A->I*		G		*A->I*		
3		D	*A->H*	*A->H*				I	
4		E	*A->J*		H		*A->J*		
5		F						J	
time									

Oliver Sinnen – o.sinnen@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Outline

- Classic task scheduling
- Contention aware task scheduling
- Identical data
  - Problem for scheduling
  - In task-graph
  - Awareness in scheduling
- Algorithm (preliminary)
- Evaluation results (preliminary)
- Conclusions

Oliver Sinnen – o.sinnen@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Classic task scheduling

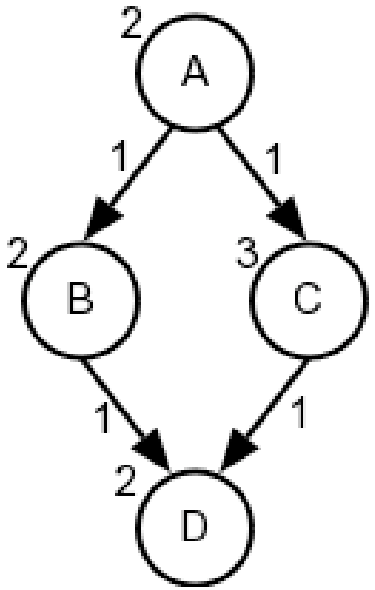
Oliver Sinnen – o.sinnen@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Classic task scheduling

Schedule definitions: DAG: *G(V,E)*, node *n*, edge *e*

- start time: $t_s(n)$; finish time: $t_f(n)$
- processor assignment: *proc(n)*



Constraints:

- Processor constraint:

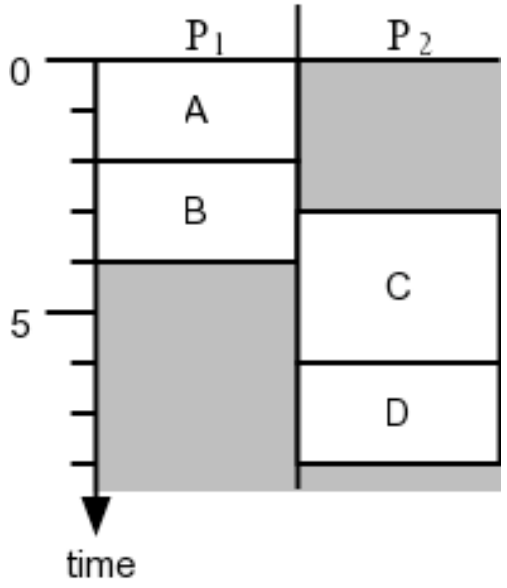
  $proc(n_i)=proc(n_j) \Rightarrow t_s(n_i) \geq t_f(n_j)$ or $t_s(n_j) \geq t_f(n_i)$

- Precedence constraint:

  for all edges e_{ji} from n_j to n_i

  $t_s(n_i) \geq t_f(n_j)$

  $+ c(e_{ji})$ if remote, i.e. $proc(n_i) \neq proc(n_j)$

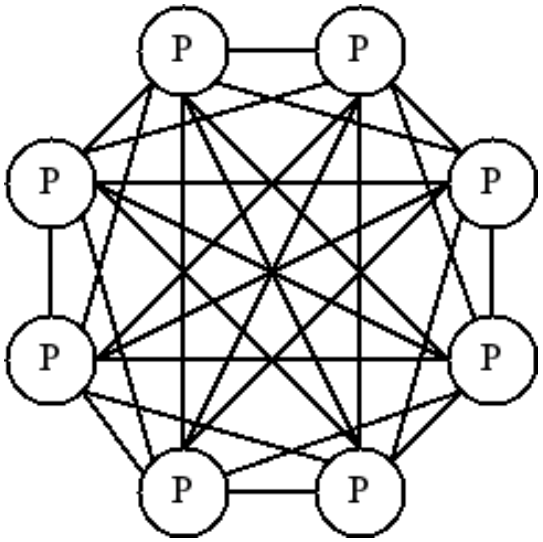


Oliver Sinnen – o.sinnen@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Classic system model

system model



e.g. 8 processors

Properties:

- Dedicated system
- Dedicated processors
- Zero-cost local communication
- Communication subsystem
- Concurrent communication
- Fully connected

Goal: find schedule with shortest schedule length (makespan)

=> NP-hard problem => many heuristics

Oliver Sinnen – o.sinnen@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

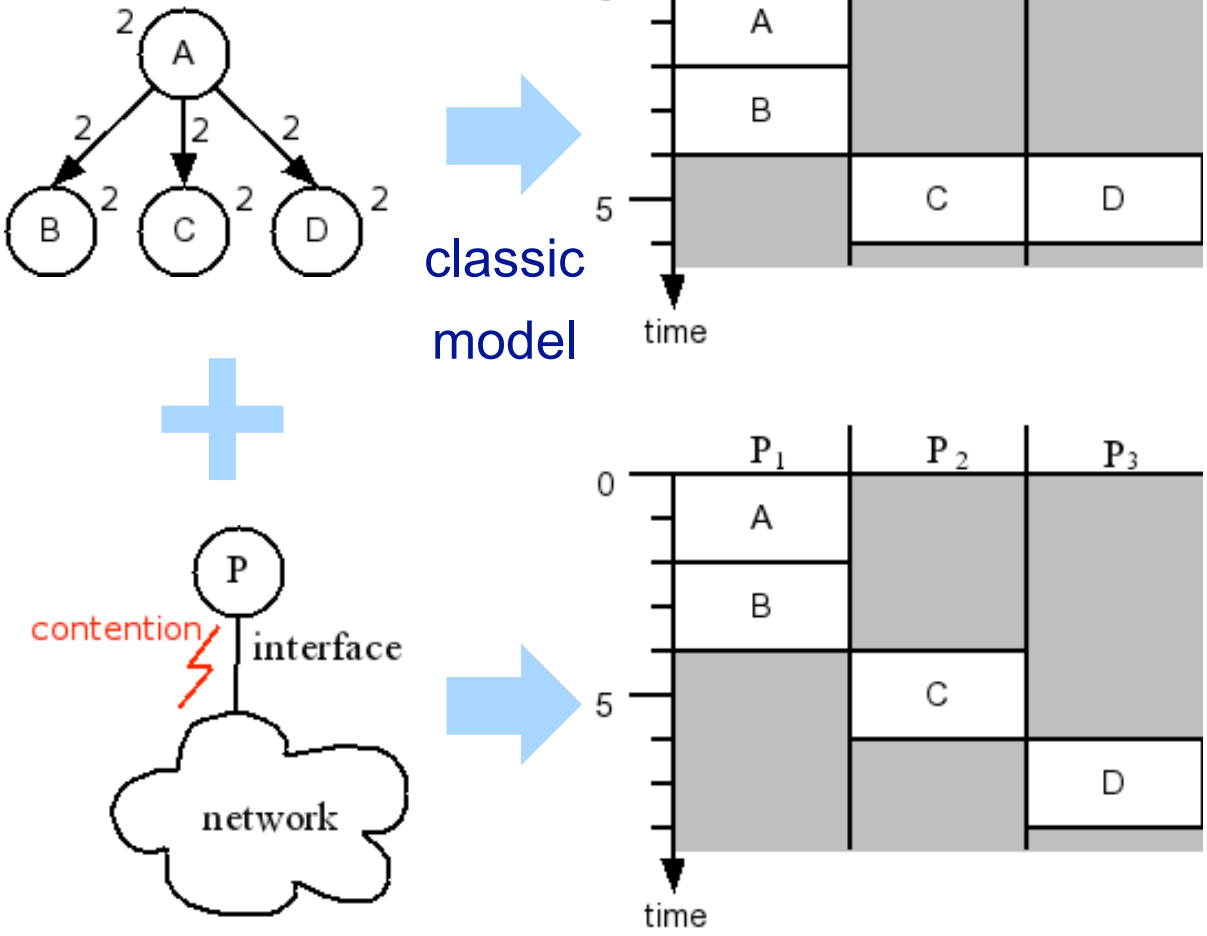
Contention aware task scheduling

Oliver Sinnen – o.sinnen@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Communication contention

contention example



- End-point contention
  - For Interface
- Most networks *not* fully connected
- Network contention
  - For network links

Oliver Sinnen – o.sinnen@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Network model

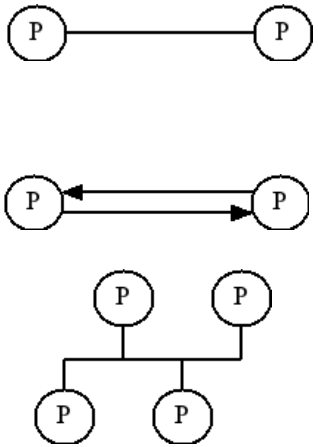
New network graph:

Vertices: processors (P) and switches (S)

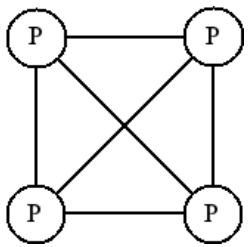
- Static and dynamic networks
- End-point and network contention

Edges: communication links (L)

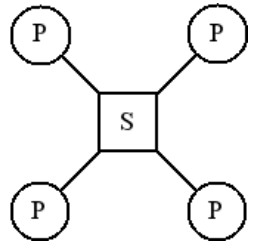
- Undirected edges
  - Half duplex
- Directed edges
  - Full duplex
- Hyperedges
  - Bus



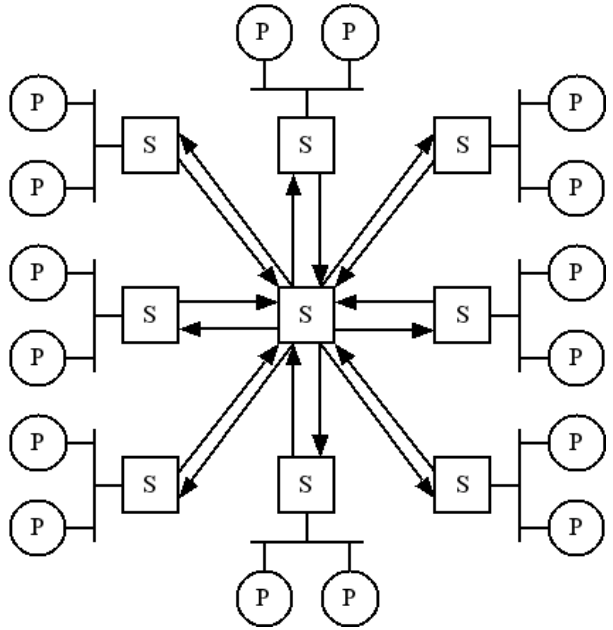
fully connected



switched LAN



example: 8 dual-processor cluster



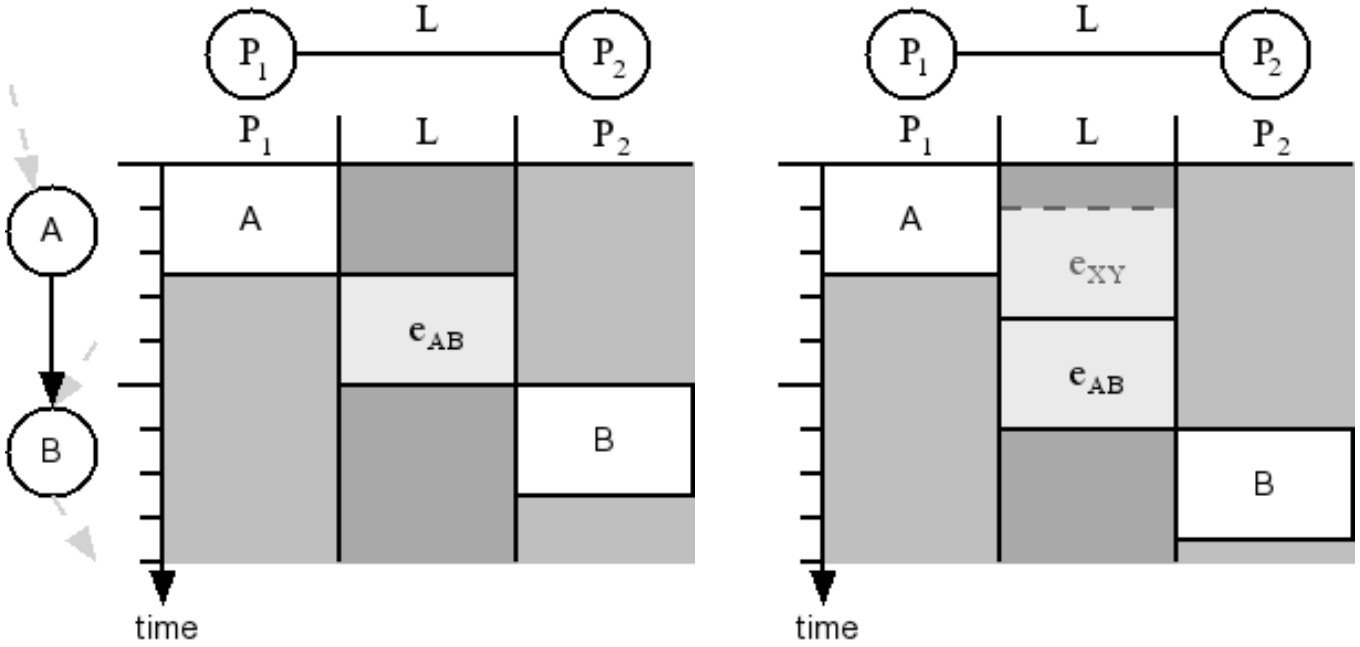
Oliver Sinnen – o.sinnen@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Contention aware scheduling

Contention awareness:

- Schedule **edges** on links
- Integration of edge scheduling into task scheduling
  - Only impact on start time of node:
  - $t_s(n_i) \geq t_f(e_{ji})$ (precedence constraint)



without contention

with contention

Oliver Sinnen – o.sinnen@auckland.ac.nz

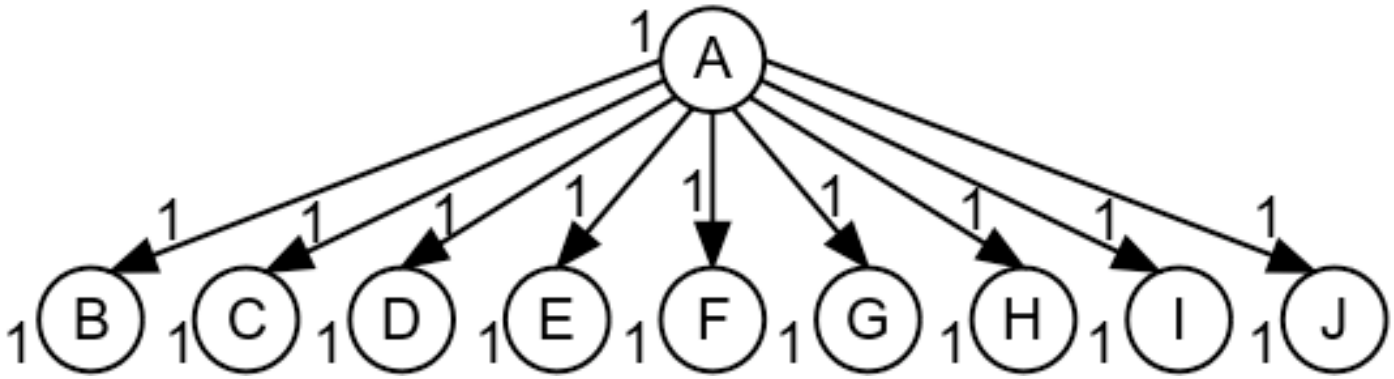
THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Identical data

Sending identical data

- Common situation: many tasks work on (partially) identical data
- Sent from one task
- Multicast – fork in task graph

Example, fork-graph with 10 tasks, *A* sends same data to child tasks

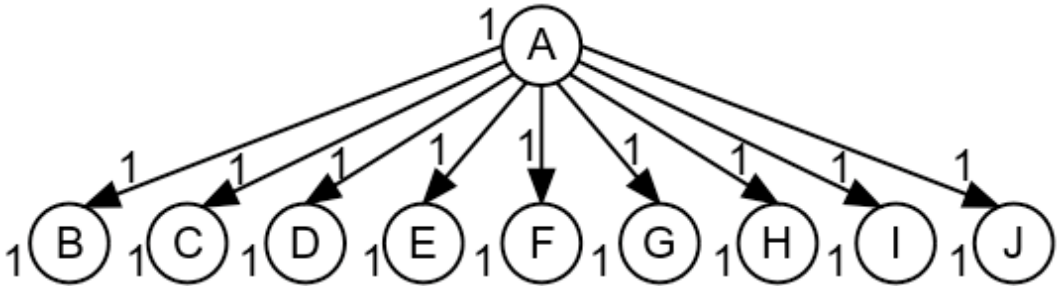


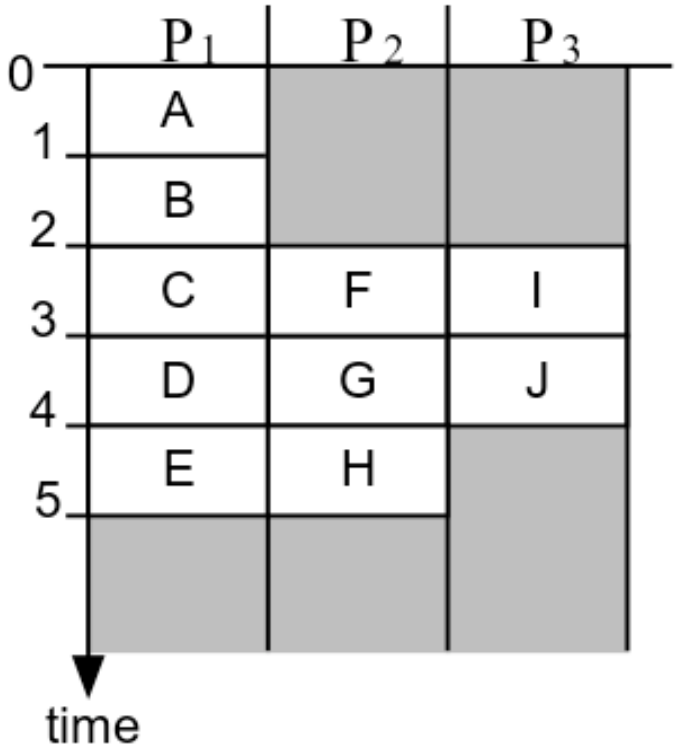
Oliver Sinnen – o.sinnen@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Identical data – classic model

Classic model: outgoing communication from A "happens" concurrently





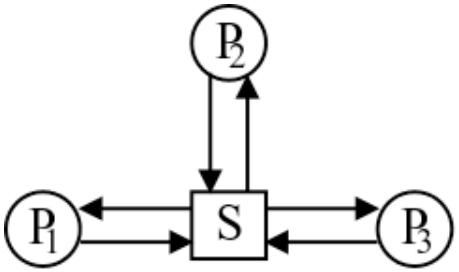
Identical data awareness would not change scheduling!

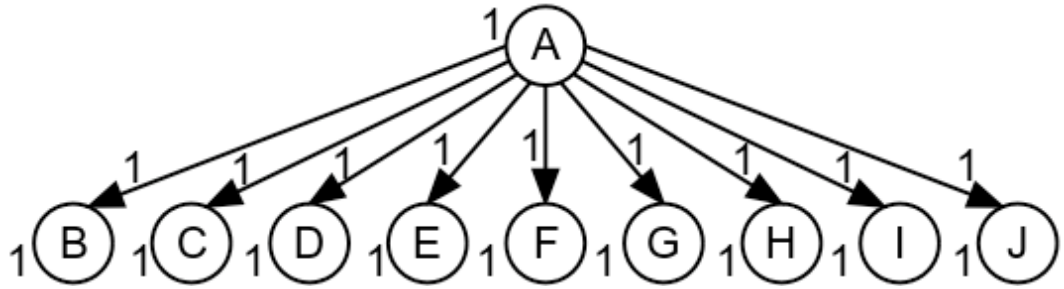
Oliver Sinnen – o.sinnen@auckland.ac.nz

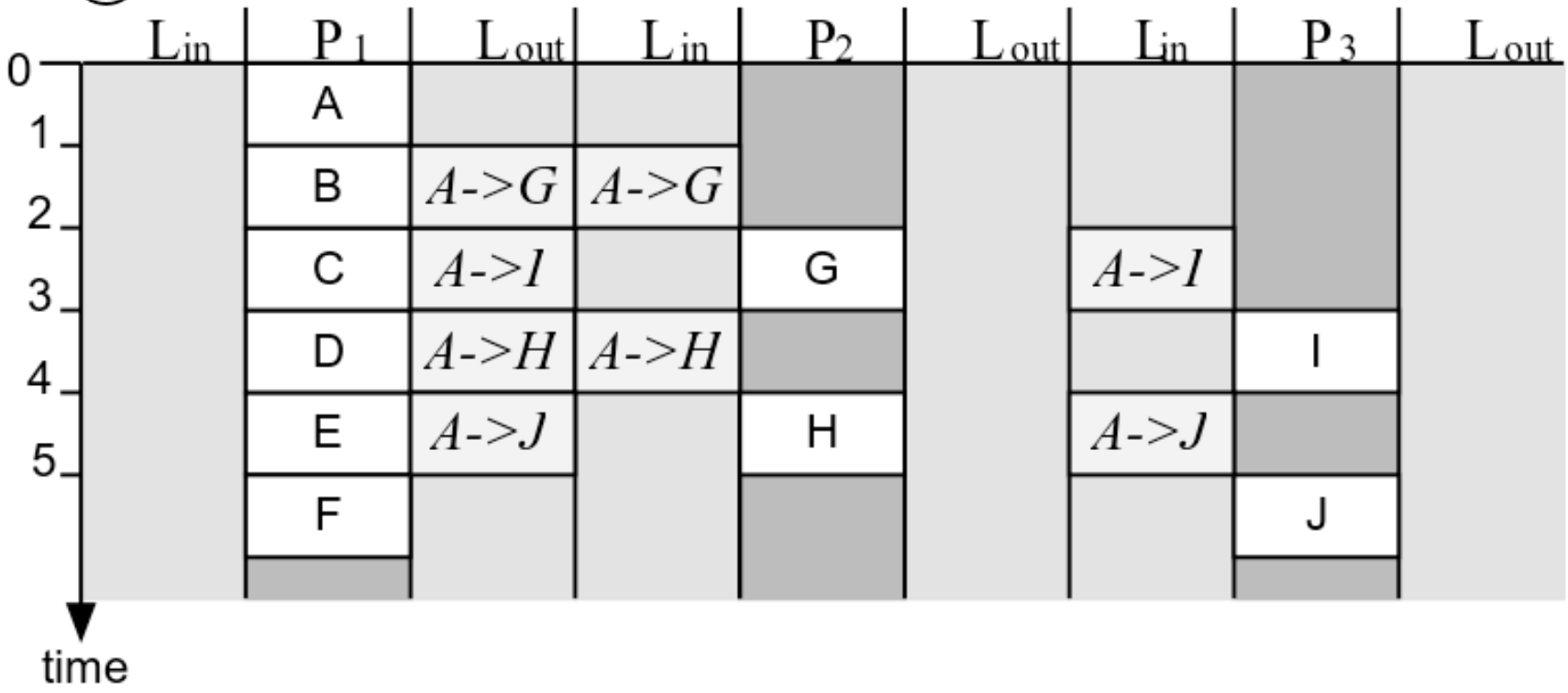
THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Identical data – contention model

Contention model: Same data is sent/scheduled many times!





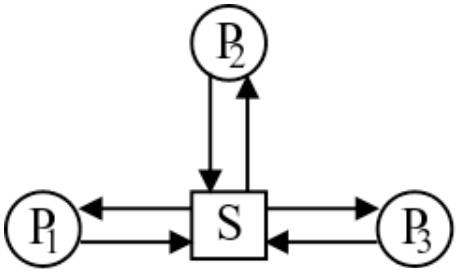


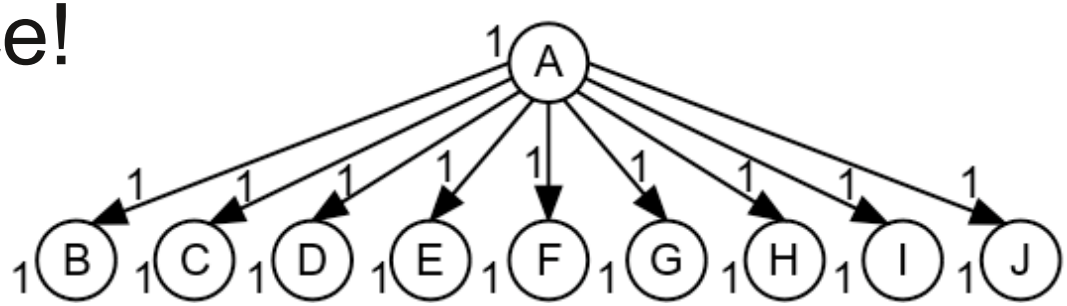
Oliver Sinnen – o.sinnen@auckland.ac.nz

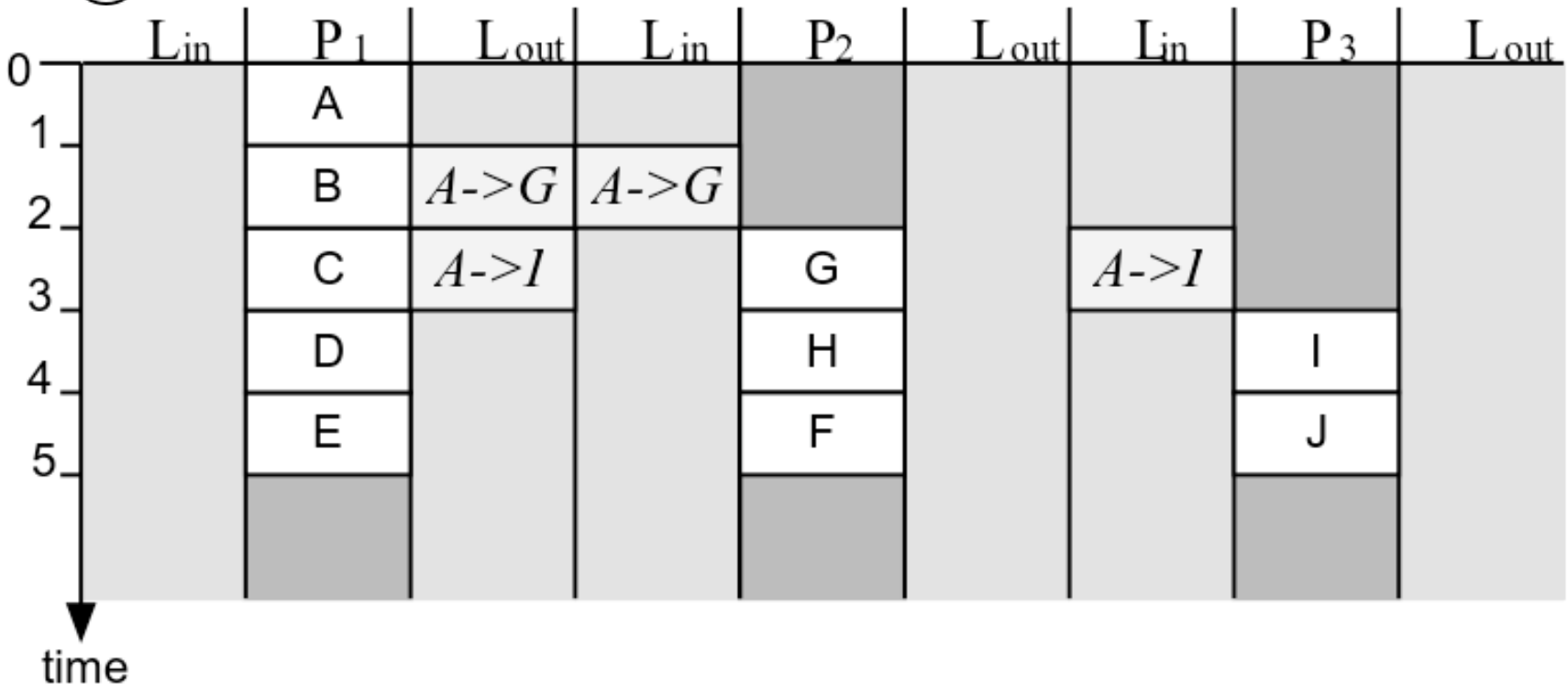
THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Identical data awareness

Identical data awareness: Same data is sent/scheduled only once!







Oliver Sinnen – o.sinnen@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Enhancing the task graph

Enhancing task graph model

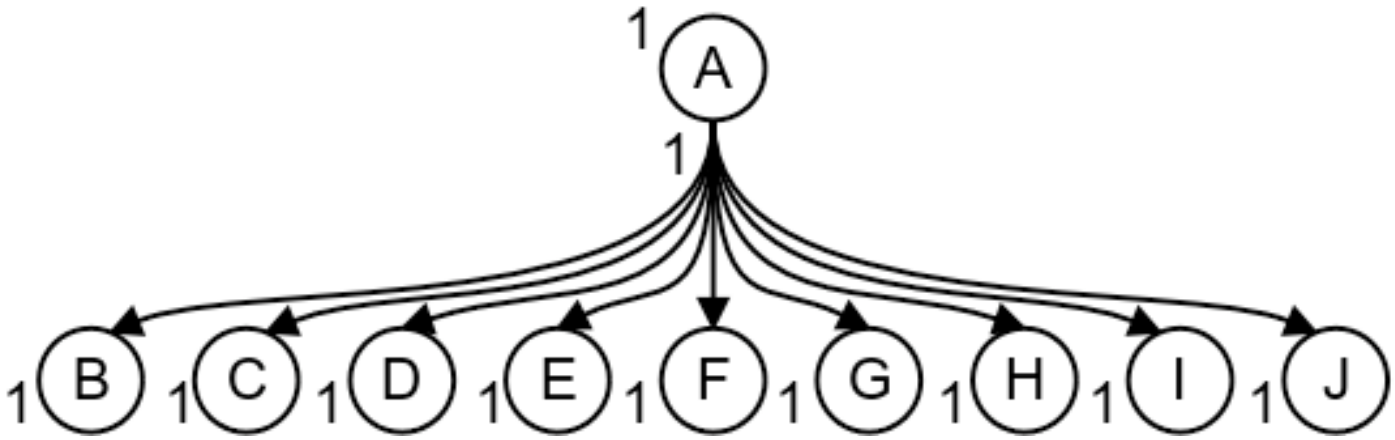
- Need to indicate when data is identical

=> **fork edges (*1*-to-*n* hyperedge)**

Enhancing task graph model

- Need to indicate when data is identical

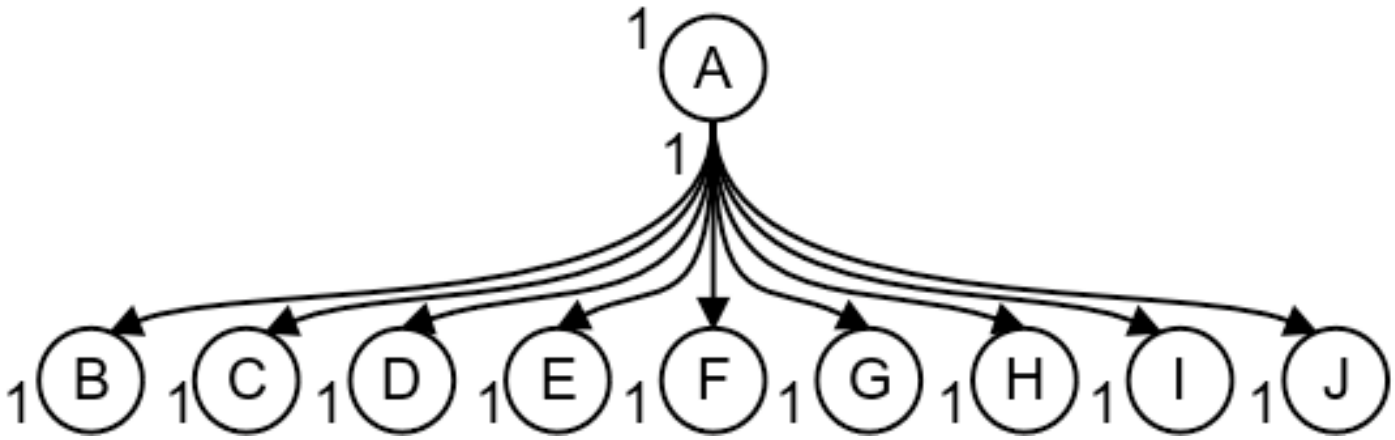
=> **fork edges (*1*-to-*n* hyperedge)**



Enhancing task graph model

- Need to indicate when data is identical

=> **fork edges (*1*-to-*n* hyperedge)**



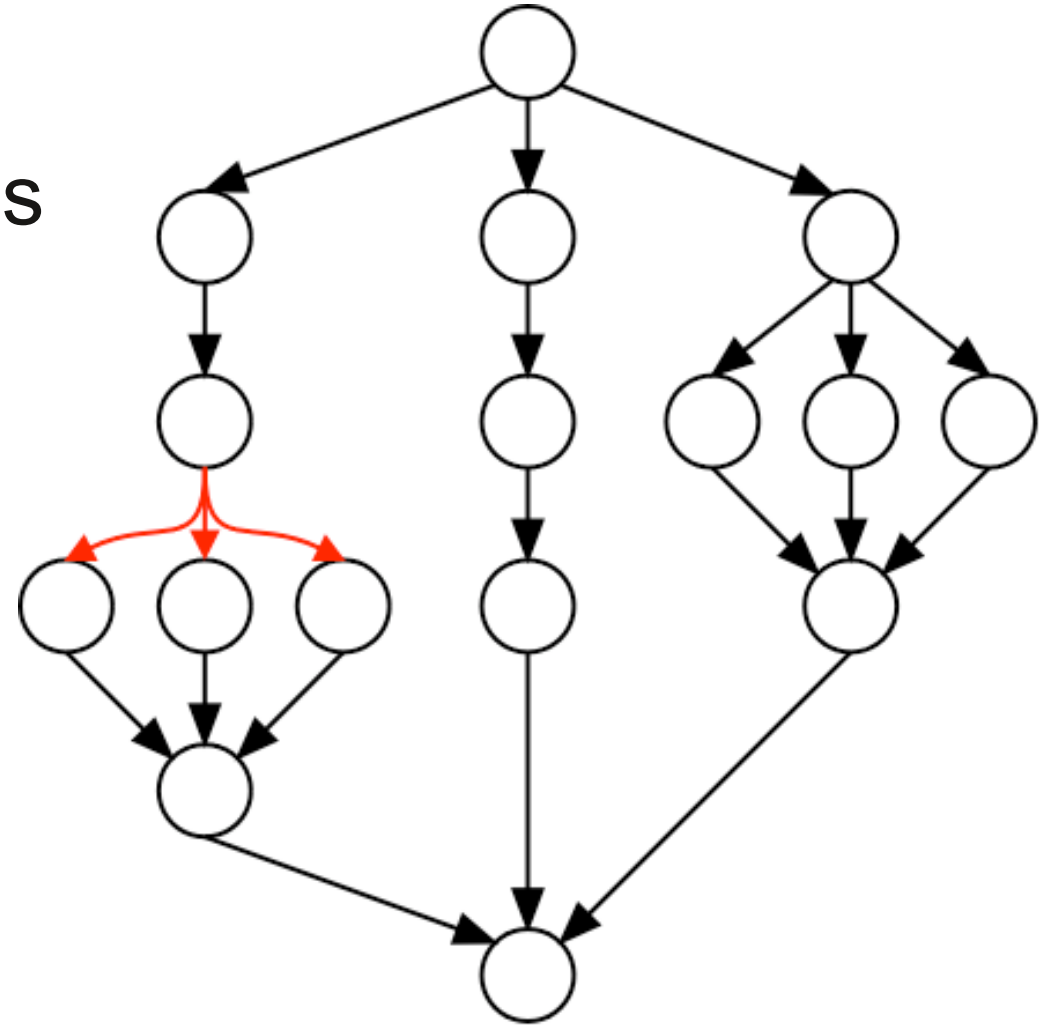
New task graph

- Still DAG
- Simple and hyperedges combined

Enhancing task graph model

New task graph

- Still DAG
- Simple and hyperedges combined



Oliver Sinnen – o.sinnen@auckland.ac.nz

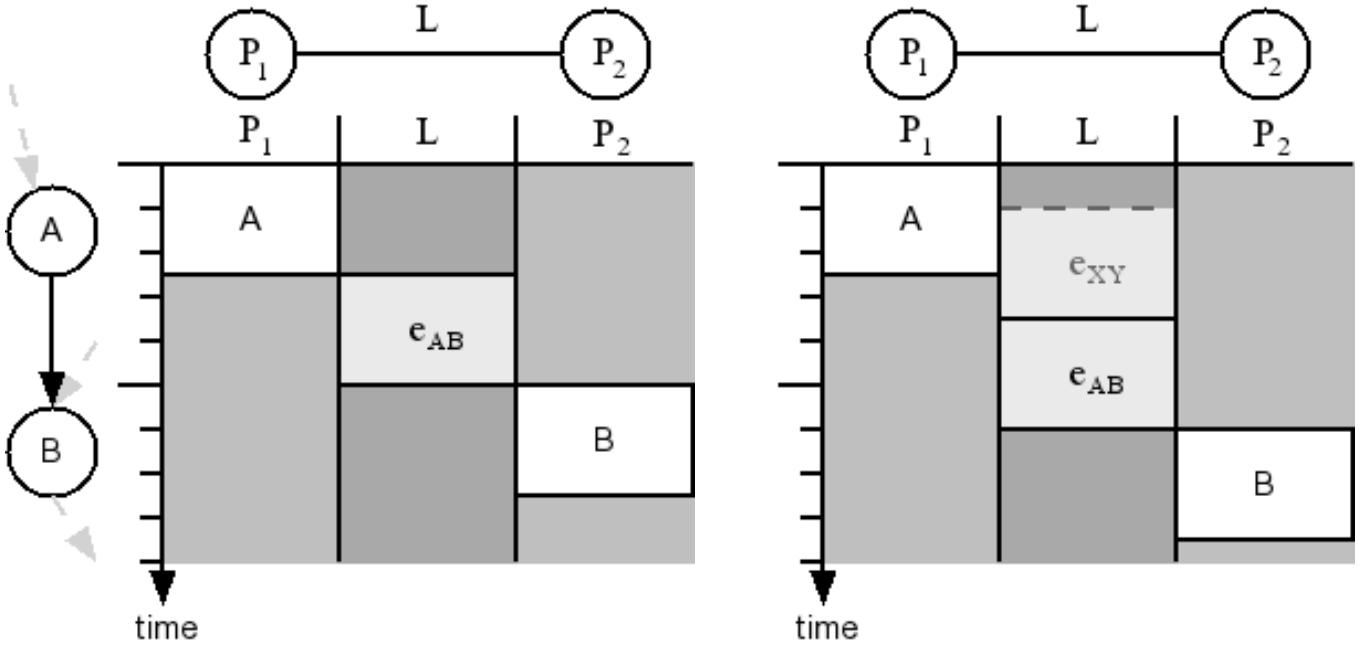
THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Identical data awareness

Remember: Contention aware scheduling

Contention awareness:

- Schedule **edges** on links
- Integration of edge scheduling into task scheduling
  - Only impact on start time of node:
  - $t_s(n_i) \geq t_f(e_{ji})$ (precedence constraint)



without contention

with contention

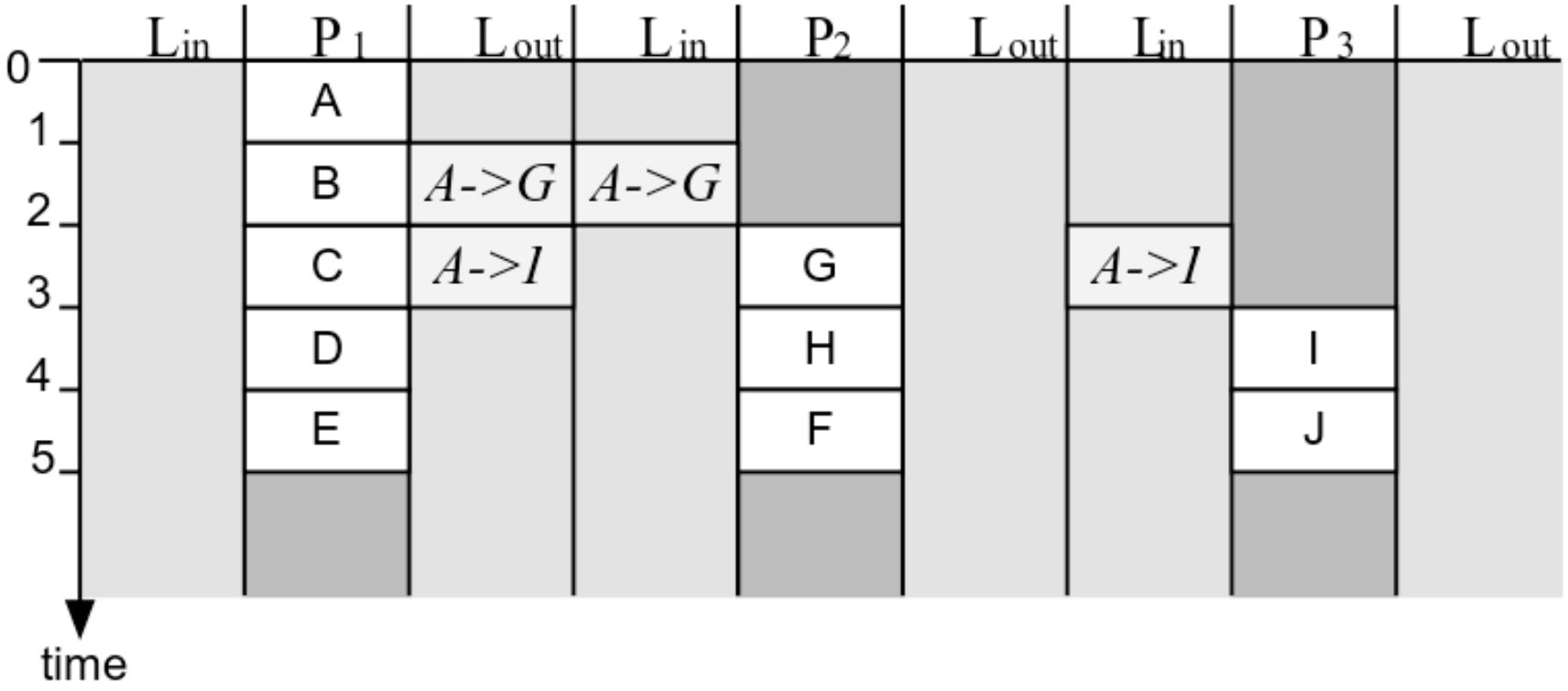
Oliver Sinnen – o.sinnen@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Scheduling procedure

Virtually no impact on contention scheduling procedures:

- If edge is already scheduled on incoming communication link: **Do not schedule again!**



Scheduling procedure

Virtually no impact on contention scheduling procedures:

- If edge is already scheduled on incoming communication link: **Do not schedule again!**

Alternative:

- Schedule only fraction, say 10%, if data is only *partially* identical

Oliver Sinnen – o.sinnen@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

(preliminary)

Algorithm

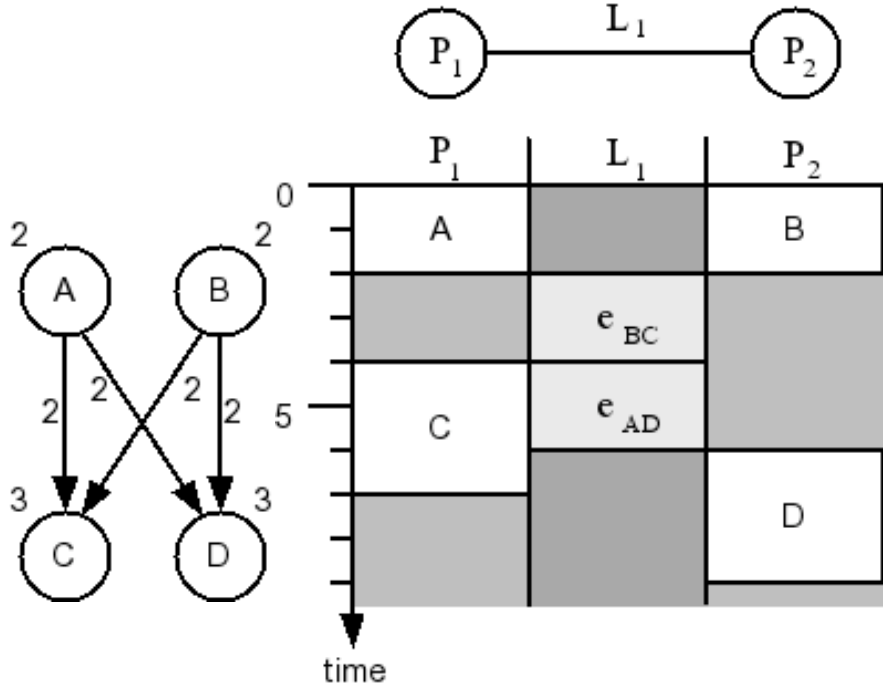
Oliver Sinnen – o.sinnen@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Contention aware list scheduling

List Scheduling

1. Make node list
   - according to priority, respecting precedence constraints
2. For each *n* in node list:
   a) Find *P* that allows earliest start time of current *n*, by tentatively scheduling all incoming edges on links
   b) Schedule *n* on chosen *P* and incoming edges on respective links



With identical data awareness

Proposal:

- Use contention aware list scheduling
- Input: adapted task graph model

Consequence:

- Each edge is only scheduled once per link/port

=> **automatic identical data awareness**

Oliver Sinnen – o.sinnen@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

(preliminary)

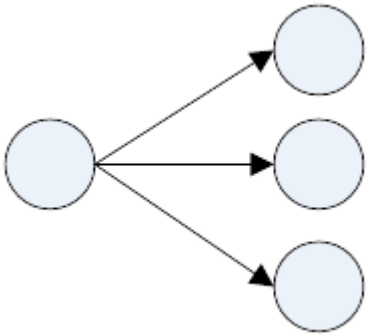
Evaluation

Simulation evaluation

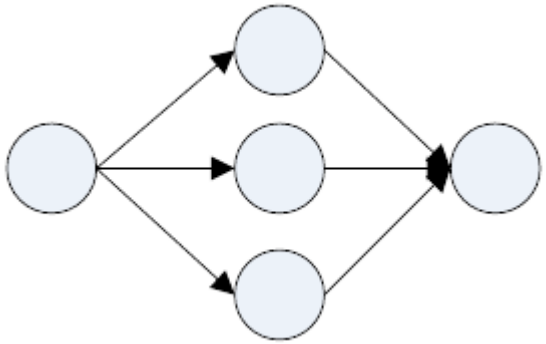
Workload:

- Scheduled large set of graphs: sizes (50-500 tasks), densities, CCR (0.1, 1, 10), graph structures (fork, fork-join, random, series-parallel, stencil)
- Certain percentage (10% or 50%) of forks are converted into hyperedges

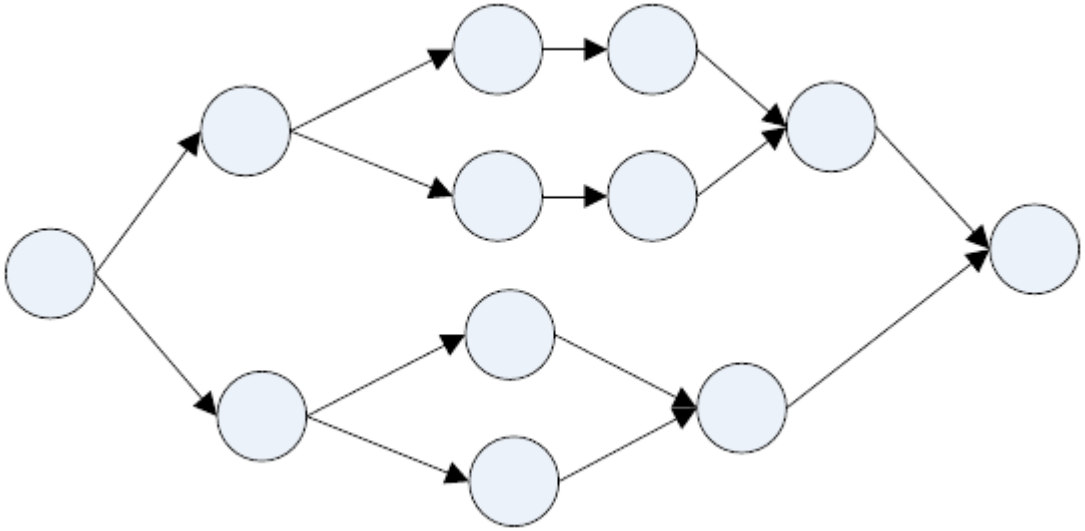
Task graph structures



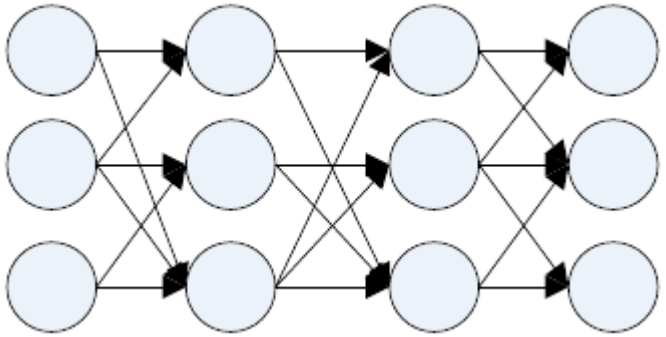
fork



fork-join



series-parallel



stencil

Oliver Sinnen – o.sinnen@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Simulation evaluation

Workload:

- Scheduled large set of graphs: sizes (50-500 tasks), densities, CCR (0.1, 1, 10), graph structures (fork, fork-join, random, series-parallel, stencil)
- Certain percentage (10% or 50%) of forks are converted into hyperedges

Two algorithms:

- Contention-aware list scheduling
  1) With identical data awareness (with IDA)
  2) Without identical data awareness (without IDA)
- ***To compare:*** *Take produced schedules from 2, remove unnecessary edges, then compact schedule (i.e. remove any gaps)*

Simulation evaluation

Workload:

- Scheduled large set of graphs: sizes (50-500 tasks), densities, CCR (0.1, 1, 10), graph structures (fork, fork-join, random, series-parallel, stencil)
- Certain percentage (10% or 50%) of forks are converted into hyperedges

Two algorithms:

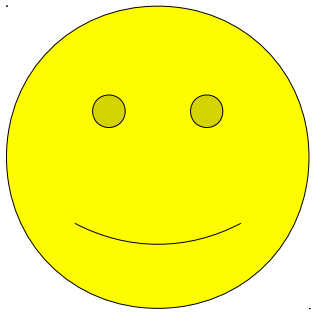
- Contention-aware list scheduling
  1) With identical data awareness (with IDA)
  2) Without identical data awareness (without IDA)
- ***To compare:*** *Take produced schedules from 2, remove unnecessary edges, then compact schedule (i.e. remove any gaps)*

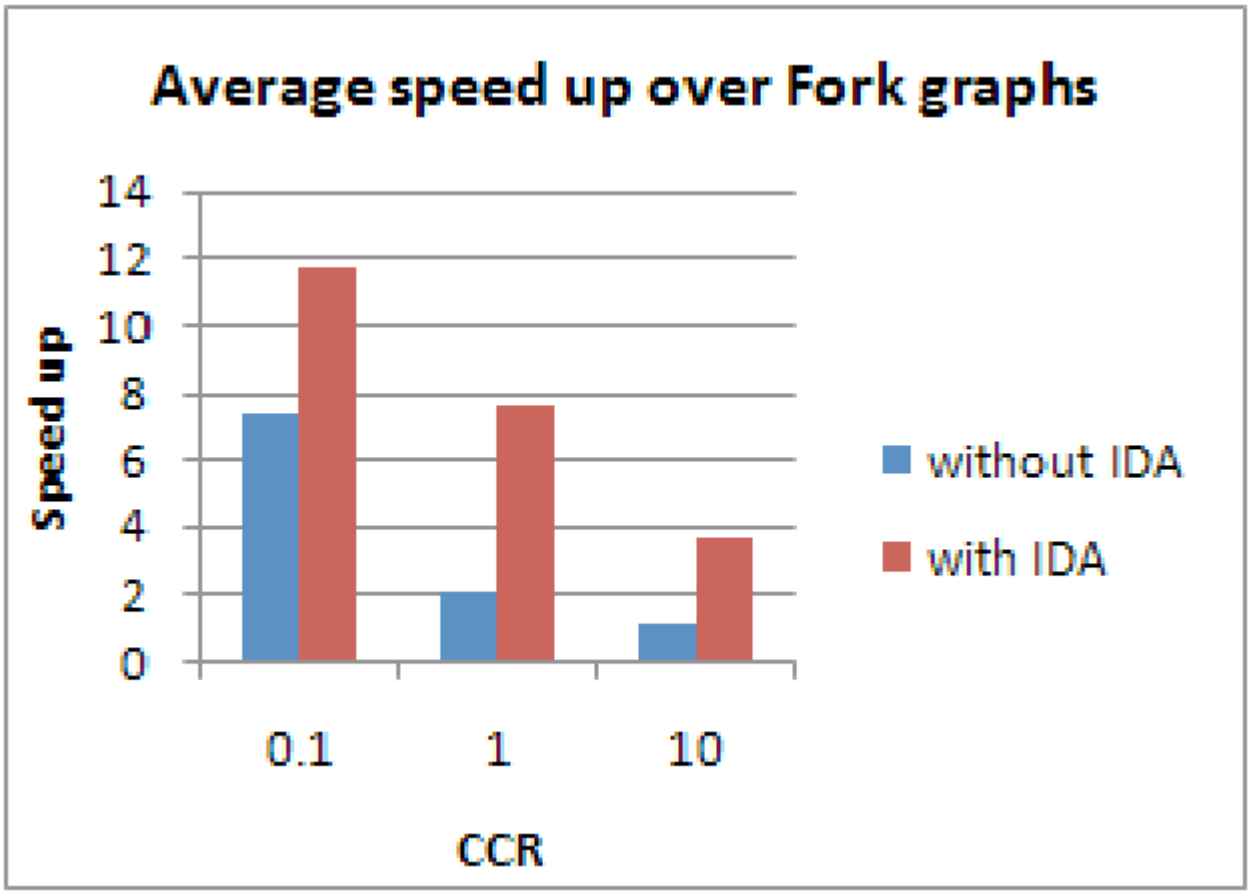
Target systems:

- Star-network with bi-directional links (one port model), various number of processors (4, 8, 32)

Results

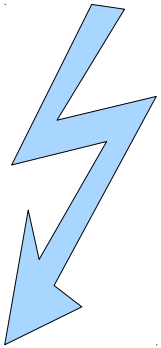
- Dramatic improvements for fork graphs

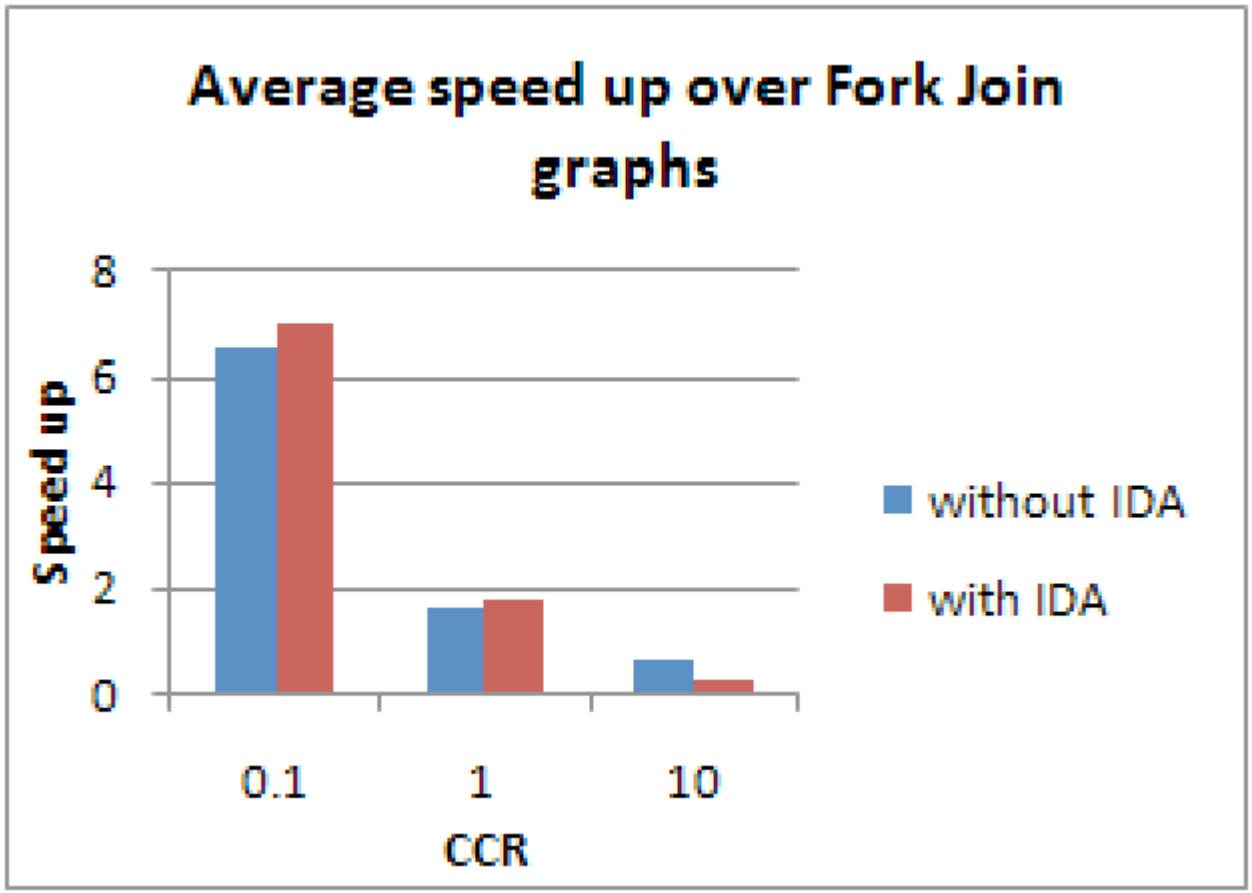




Results

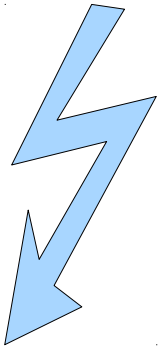
- But only small for fork-joins (CCR 0.1 and 1)

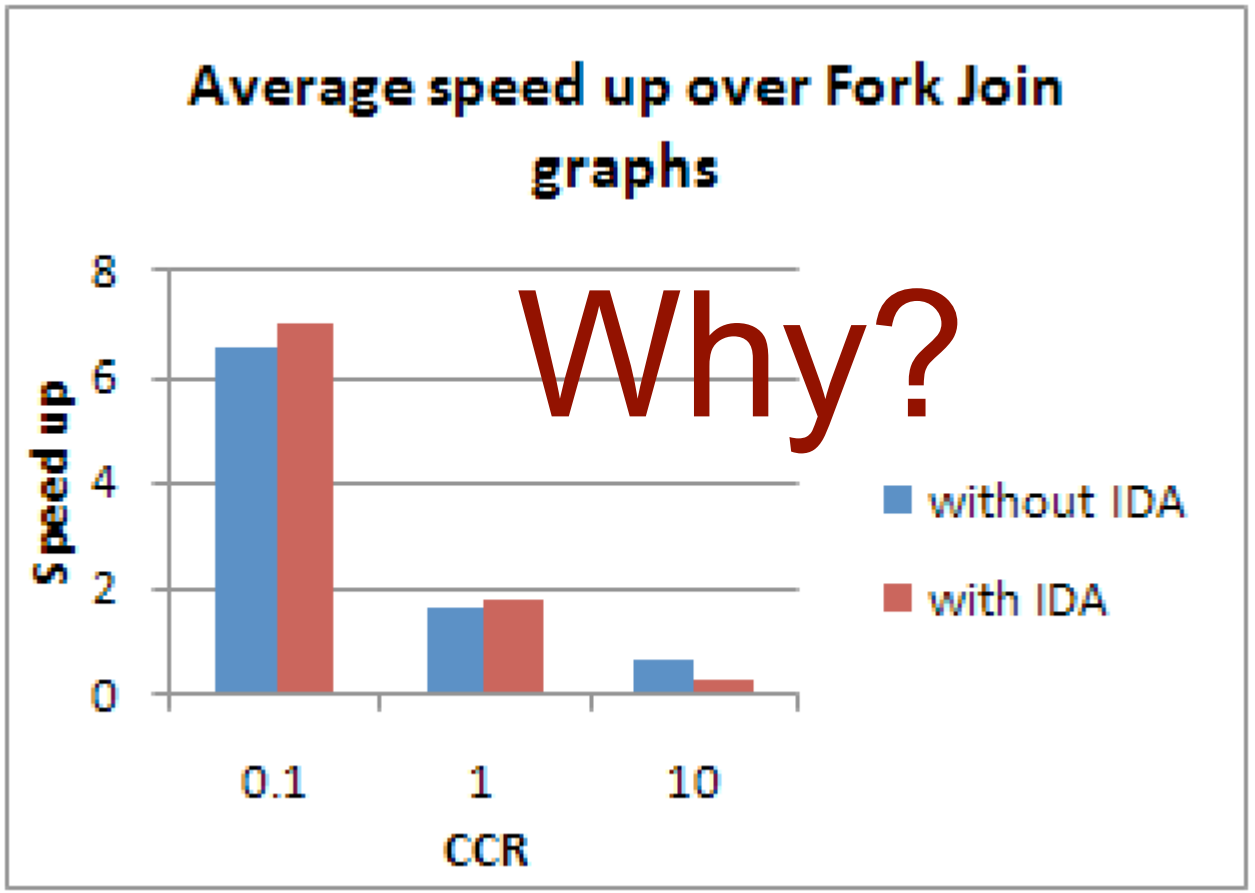




Results

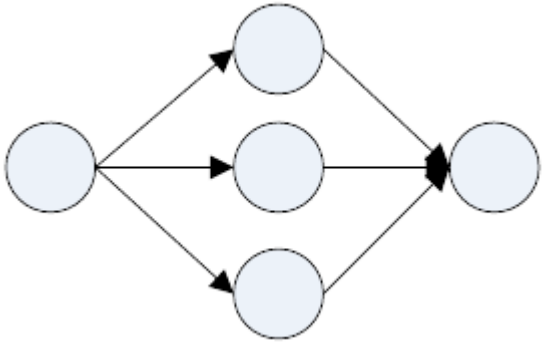
- But only small for fork-joins (CCR 0.1 and 1)





Results

Why are schedules for fork-join bad?

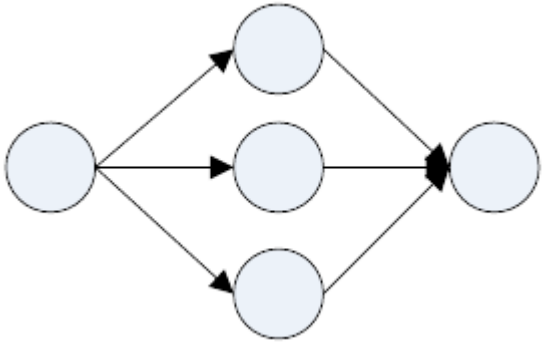


- Problem is scheduling heuristic
- List scheduling is not appropriate for contention scheduling, especially with IDA

=> Does not look ahead

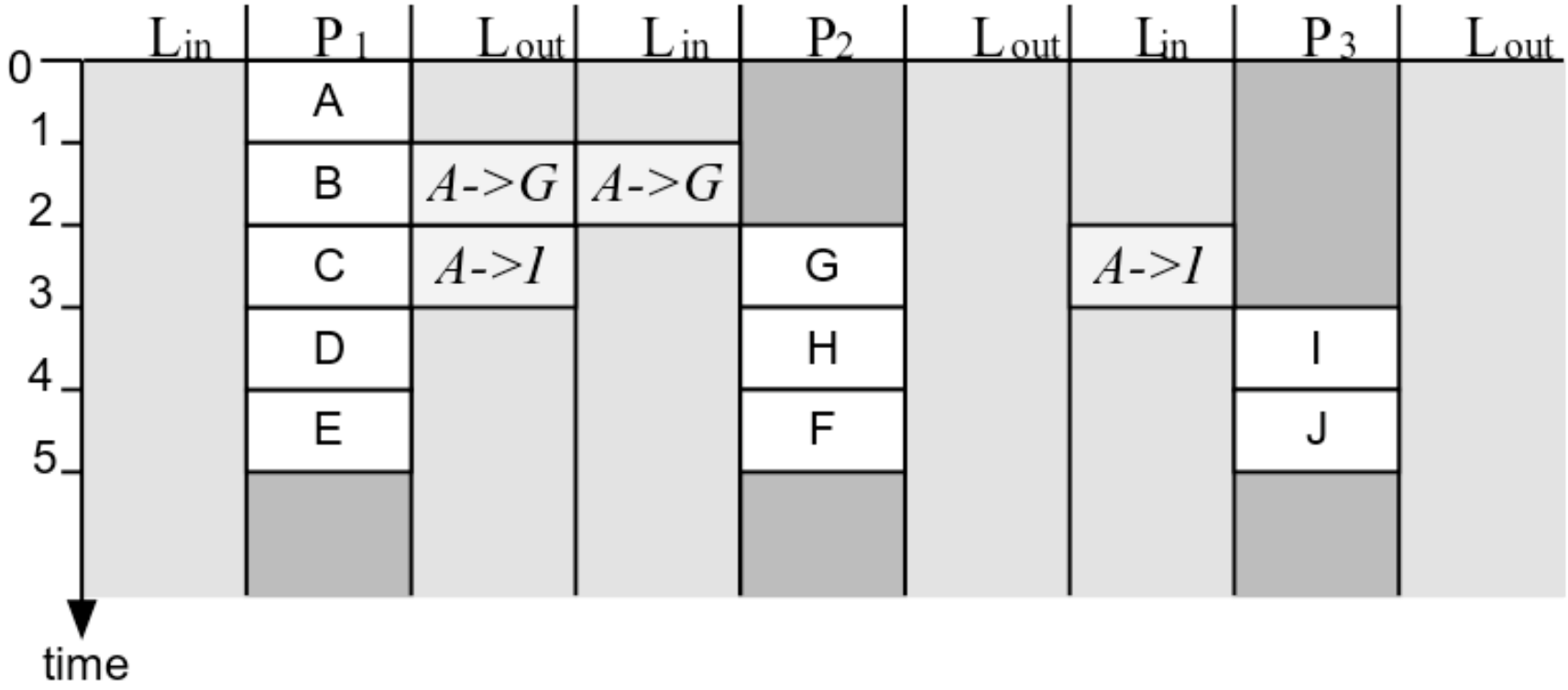
Results

Why are schedules for fork-join bad?



- Problem is scheduling heuristic
- List scheduling is not appropriate for contention scheduling, especially with IDA

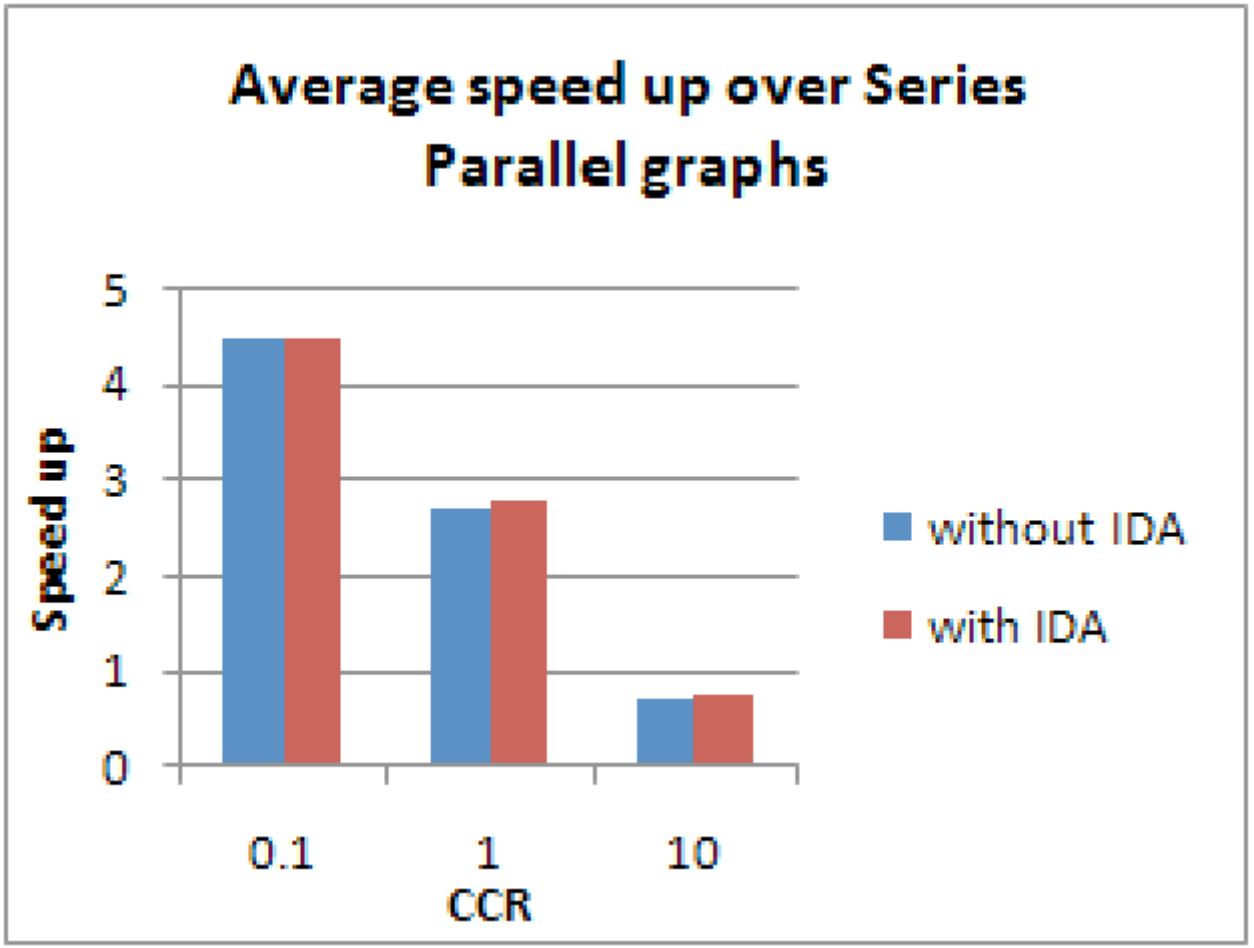
=> Does not look ahead



Oliver Sinnen – o.sinnen@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

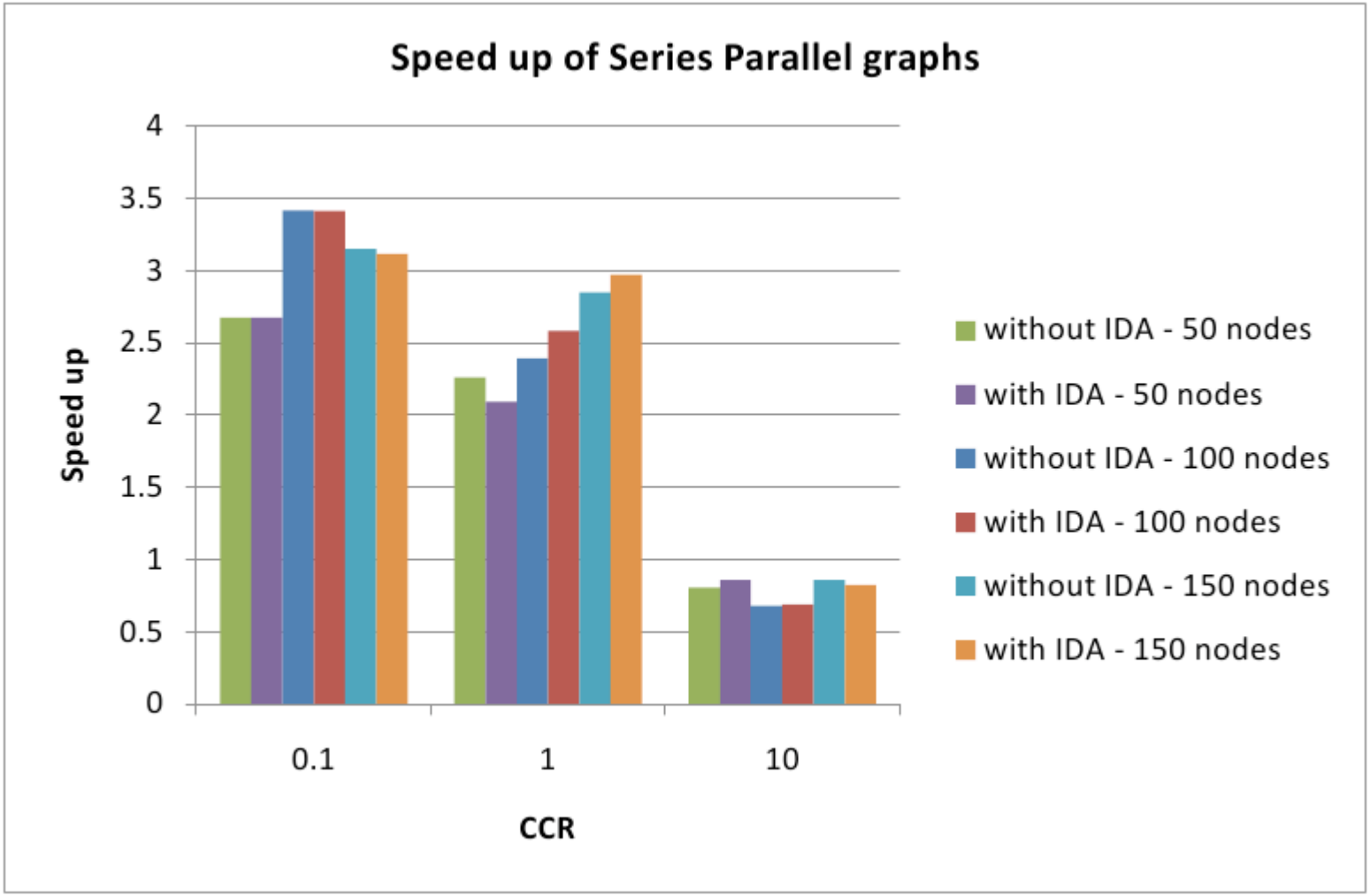
Results



Oliver Sinnen – o.sinnen@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Results



Conclusions

- Identifying identical data
  - *unnecessary* under classic model
  - *crucial* under contention model
- Identical data awareness is easy to integrate with enhanced task graph model
- Preliminary results show *great promise*, but also that better scheduling algorithm is necessary

- We are currently working on that!
- Also important to create realistic graphs

www.ece.auckland.ac.nz/~sinnen/