Automating Data-Throttling Analysis for Data-Intensive Workflow

Ricardo J. Rodríguez, Rafael Tolosana-Calasanz, Omer F. Rana
{rjrodriguez, rafaelt}@unizar.es, o.f.rana@cs.cardiff.ac.uk

Universidad de Zaragoza
Zaragoza, Spain

Cardiff University
Cardiff, United Kingdom

May 15^{th}, 2012

CCGrid'12: 12^{th} IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing
Ottawa, Canada

Outline

Universidad Zaragoza

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Motivation (I): data movement issue

Scientific Workflows

- Control/data flow graph
  - Set of tasks
  - Dependencies

Universidad Zaragoza

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Motivation (I): data movement issue

Scientific Workflows

- Control/data flow graph
  - Set of tasks
  - Dependencies

Data-Intensive Workflows

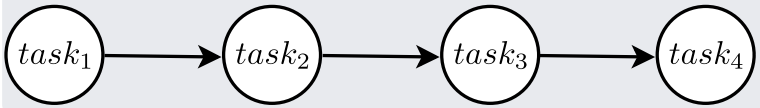
- Data workload $\geq$ computational workload
- How to deal with this?
  - Minimise data movement
  - Move data via higher capacity links as fast as possible

Universidad Zaragoza

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Motivation (II): data-movement policies

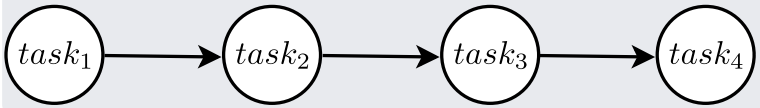
Pipeline workflow



- Transmit as fast as possible → OK!

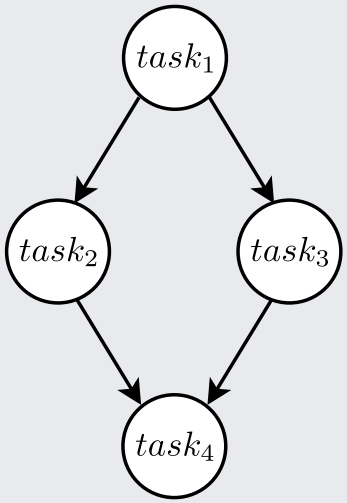
Motivation (II): data-movement policies

Pipeline workflow



- Transmit as fast as possible → OK!

Directed Acyclic Graph workflow



- Transmit as fast as possible → WRONG!
- Cluster tasks → minimise data movement
- Problems may arise
  - Isolation view: networking + buffering (merge tasks)
  - In-the-large: harmful (finite buffer capacity → cannot be reused)

Motivation (III): deriving data-throttling

The goal

- Balance workflow paths...
- ...eliminating unnecessary bandwidth usage

Motivation (III): deriving data-throttling

The goal

- Balance workflow paths...
- ...eliminating unnecessary bandwidth usage
  - Efficient buffer usage
  - Efficient network bandwidth usage

Universidad Zaragoza

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Motivation (III): deriving data-throttling

The goal

- Balance workflow paths...
- ...eliminating unnecessary bandwidth usage
  - Efficient buffer usage
  - Efficient network bandwidth usage

Open question: **WHAT strategy should I use?**

Universidad Zaragoza

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Motivation (III): deriving data-throttling

The goal

- Balance workflow paths...
- ...eliminating unnecessary bandwidth usage
  - Efficient buffer usage
  - Efficient network bandwidth usage

Open question: **WHAT strategy should I use?**

Our approach

- Automatically derive values for data-throttling

Motivation (III): deriving data-throttling

The goal

- Balance workflow paths...
- ...eliminating unnecessary bandwidth usage
  - Efficient buffer usage
  - Efficient network bandwidth usage

Open question: **WHAT strategy should I use?**

Our approach

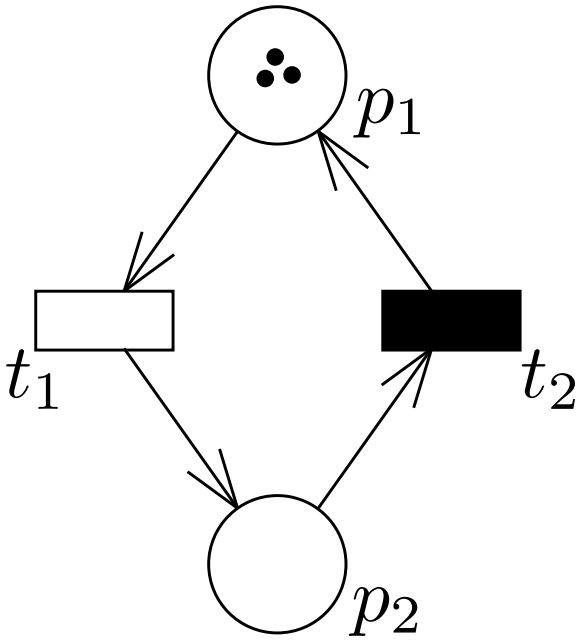
- Automatically derive values for data-throttling
- Directed Acyclic Graphs (DAGs) $\rightarrow$ Petri nets (performance model)
- Analysis on Petri net model (explained later :))

Outline

Universidad Zaragoza

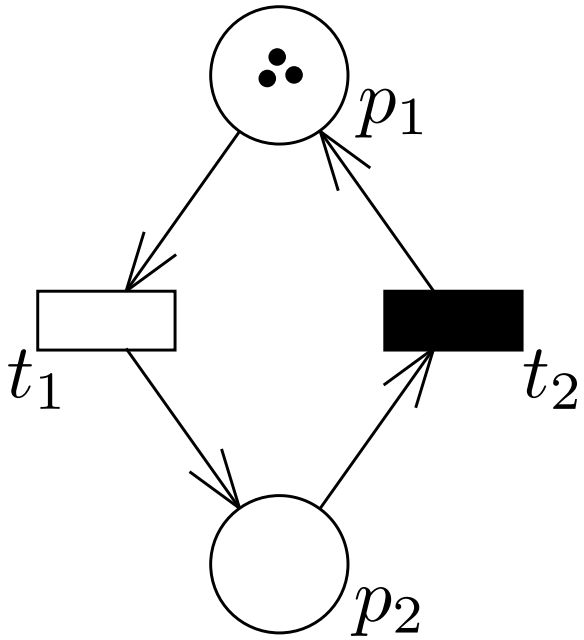
CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Background (I): Petri nets (PNs)



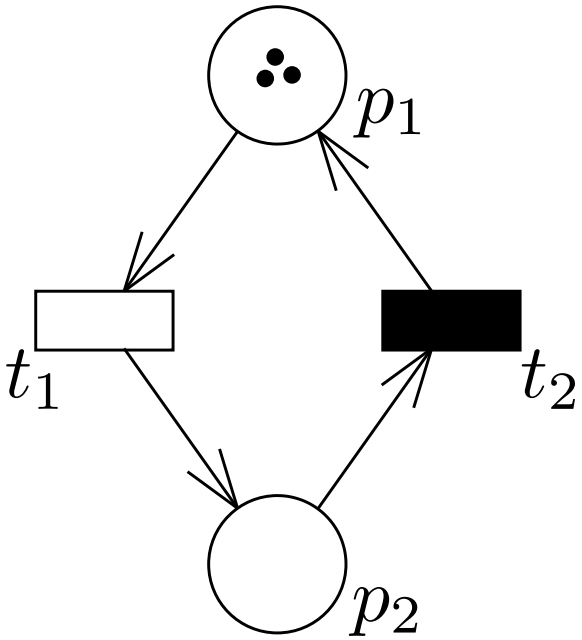
- Mathematical formalism

Background (I): Petri nets (PNs)



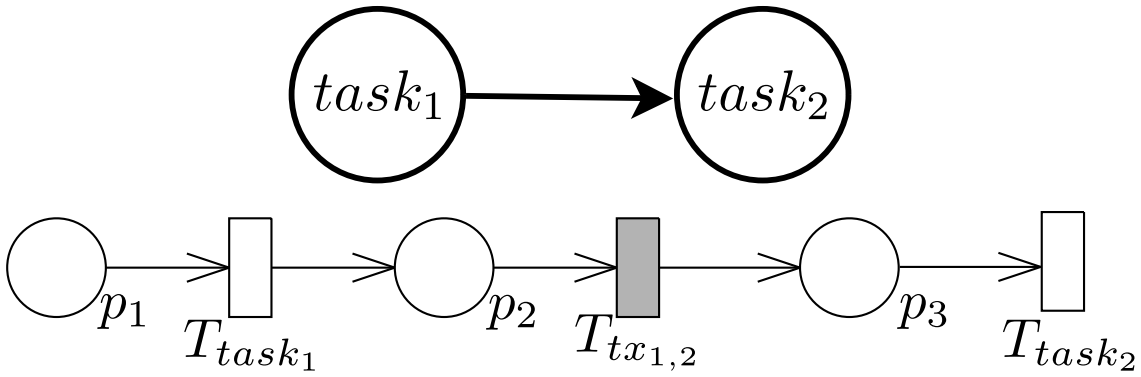
- Mathematical formalism
- Places (circles, p_X)
- Transitions (bars, t_X). Associated delay
  - Immediate ($\delta_{t_X} = 0$)
  - Timed (δ_{t_X} exponential distribution $\rightarrow$ Stochastic Petri Nets)
- Tokens (black dots)
- Directed arcs
  - Place $\rightarrow$ Transition
  - Transition $\rightarrow$ Place
- Initial marking: no. tokens on places

Background (I): Petri nets (PNs)



- Mathematical formalism
- Places (circles, p_X)
- Transitions (bars, t_X). Associated delay
  - Immediate ($\delta_{t_X} = 0$)
  - Timed (δ_{t_X} exponential distribution $\rightarrow$ Stochastic Petri Nets)
- Tokens (black dots)
- Directed arcs
  - Place $\rightarrow$ Transition
  - Transition $\rightarrow$ Place
- Initial marking: no. tokens on places
- Enables modelling
  - Concurrency
  - Synchronisation

Background (II): From DAG to PN

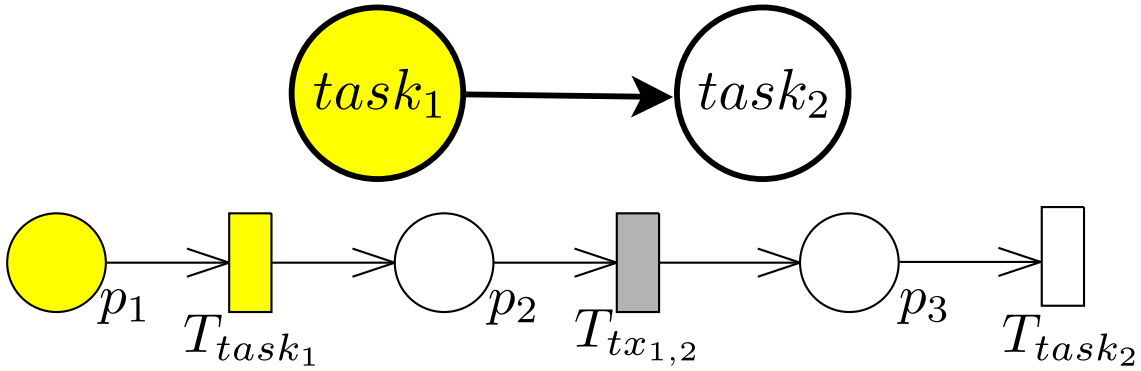


Transforming from a DAG to a PN

- Computation task $\rightarrow$ place + timed transition
  - Delay equal to computation task time

Zaragoza

Background (II): From DAG to PN

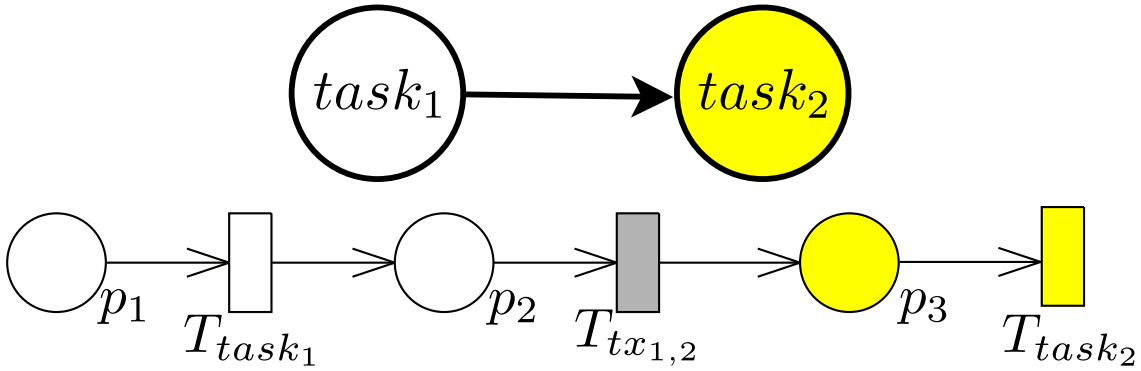


Transforming from a DAG to a PN

- Computation task $\rightarrow$ place + timed transition
  - Delay equal to computation task time

Zaragoza

Background (II): From DAG to PN

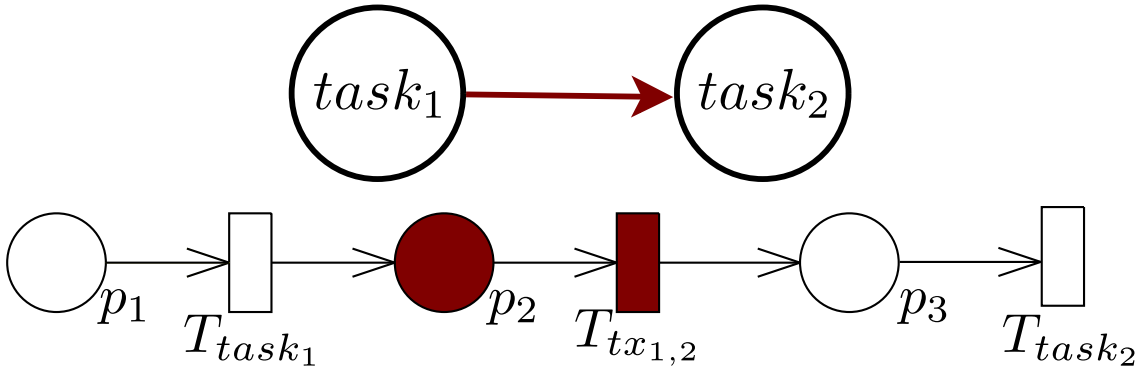


Transforming from a DAG to a PN

- Computation task → place + timed transition
  - Delay equal to computation task time

Zaragoza

Background (II): From DAG to PN



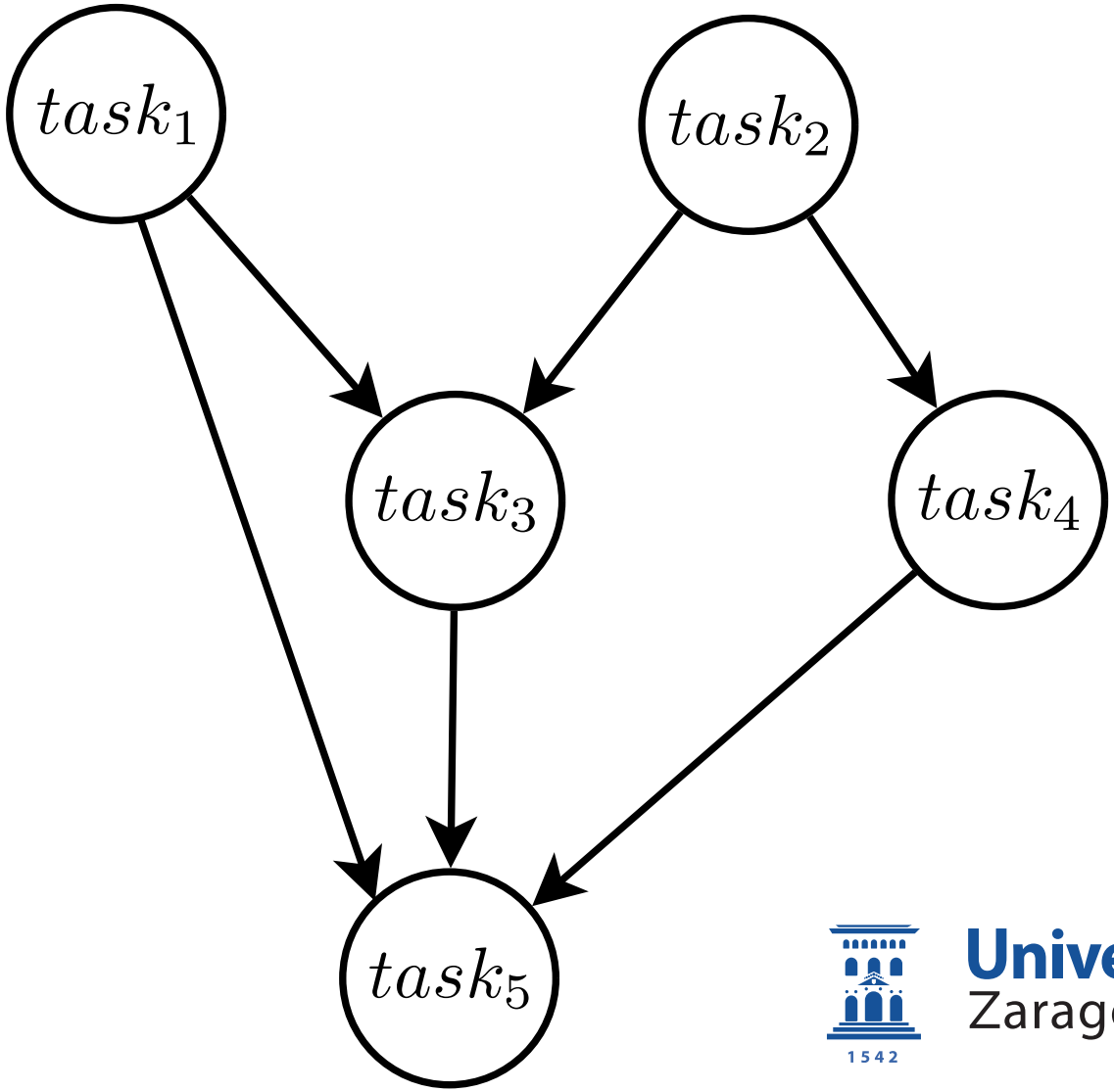
Transforming from a DAG to a PN

- Computation task $\rightarrow$ place + timed transition
  - Delay equal to computation task time
- Transmission link $\rightarrow$ place + timed transition
  - Delay equal to transmission time: $\delta_t = \frac{\textit{data size transmitted}}{\textit{link bandwidth}}$

Zaragoza

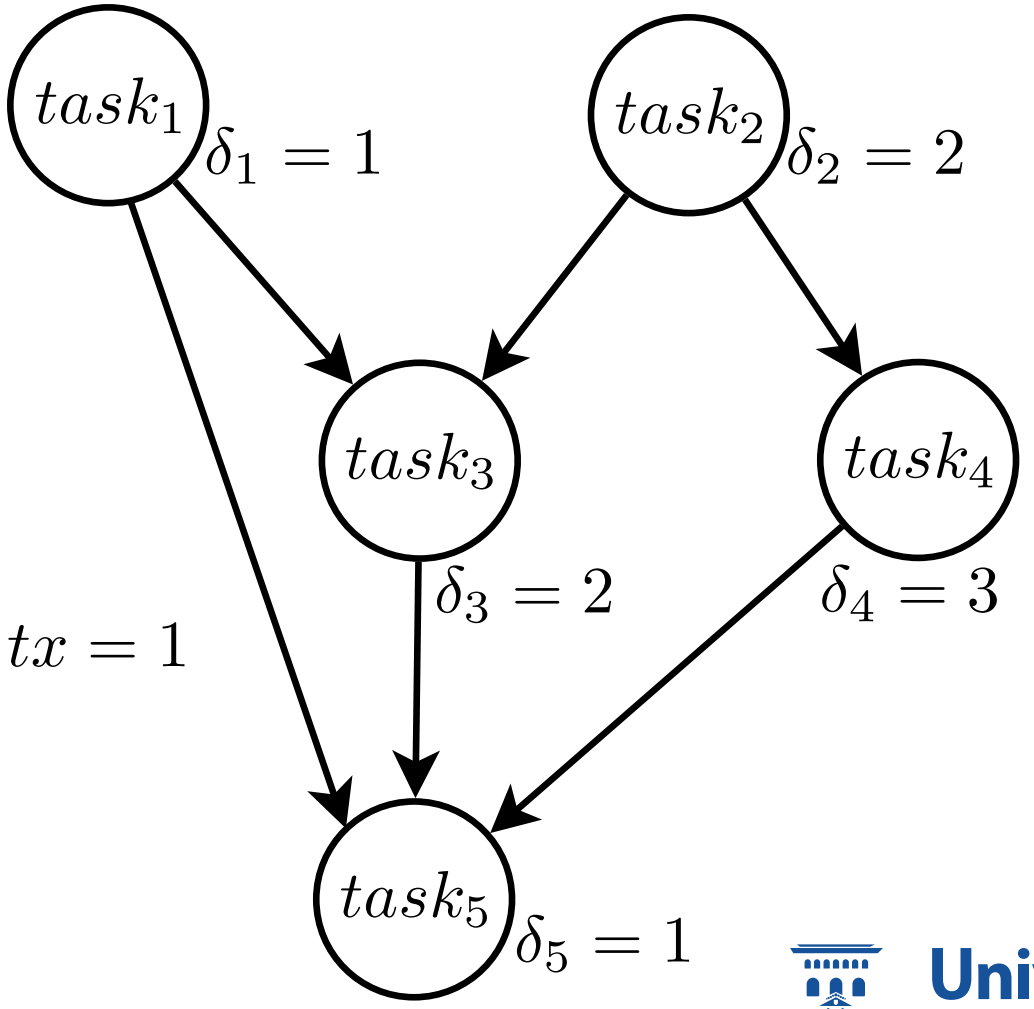
Background (III): Slack concept (1)

For human beings: an example



Background (III): Slack concept (1)

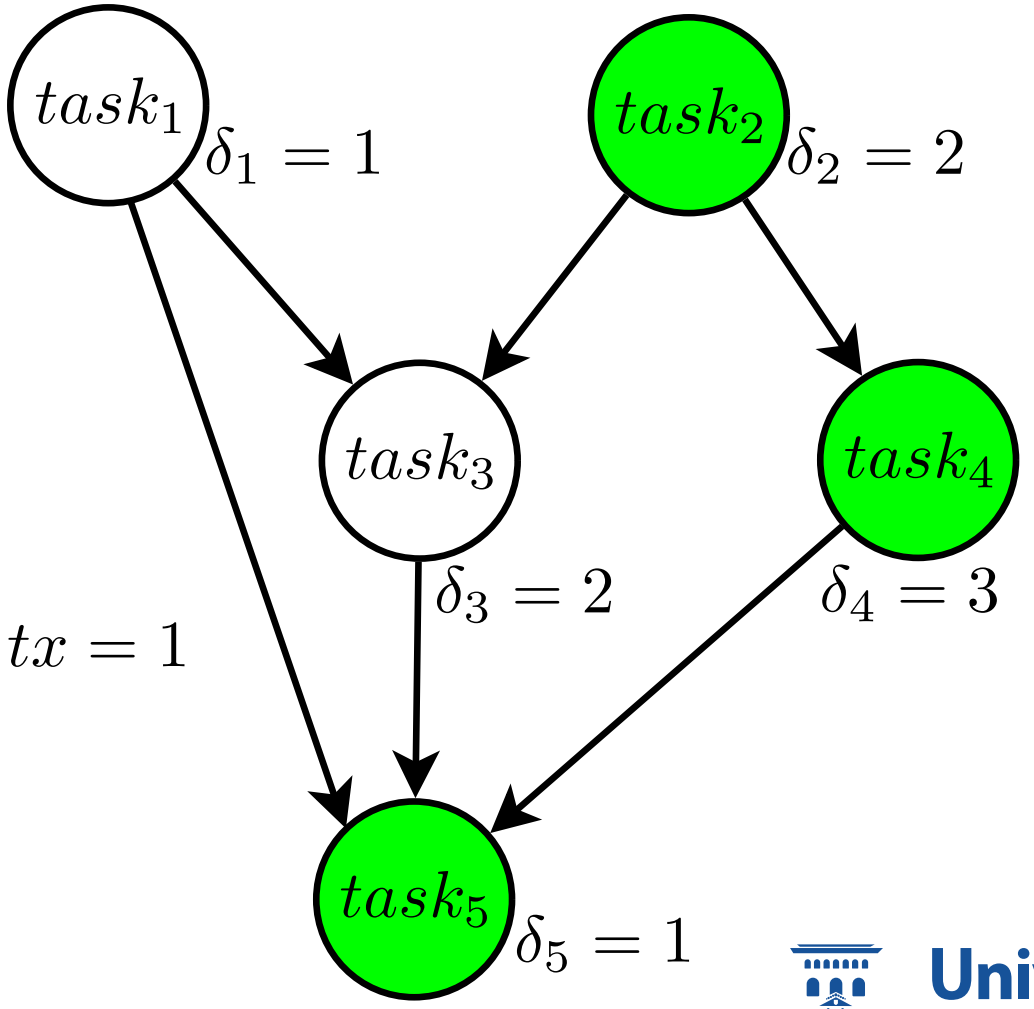
For human beings: an example



δ execution time; tx transmission time

Background (III): Slack concept (1)

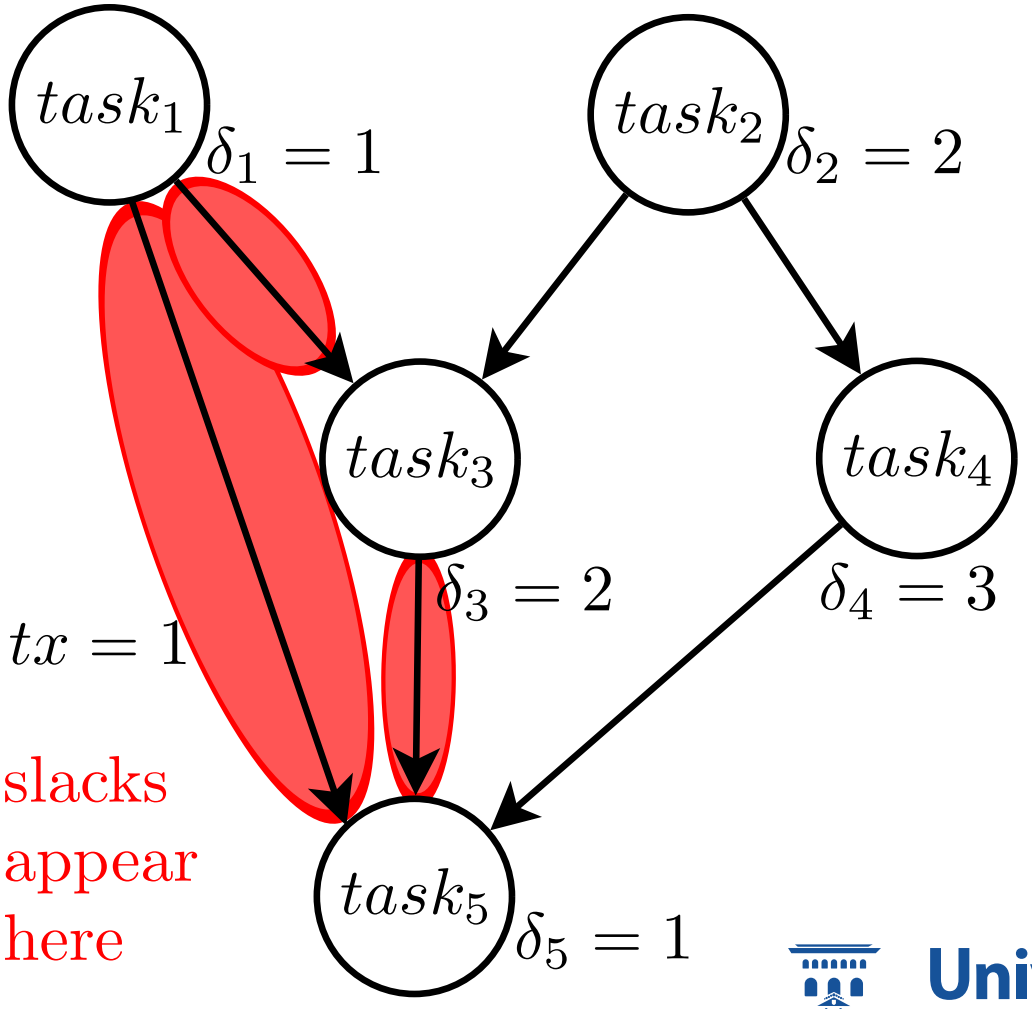
For human beings: an example



δ execution time; tx transmission time

Background (III): Slack concept (1)

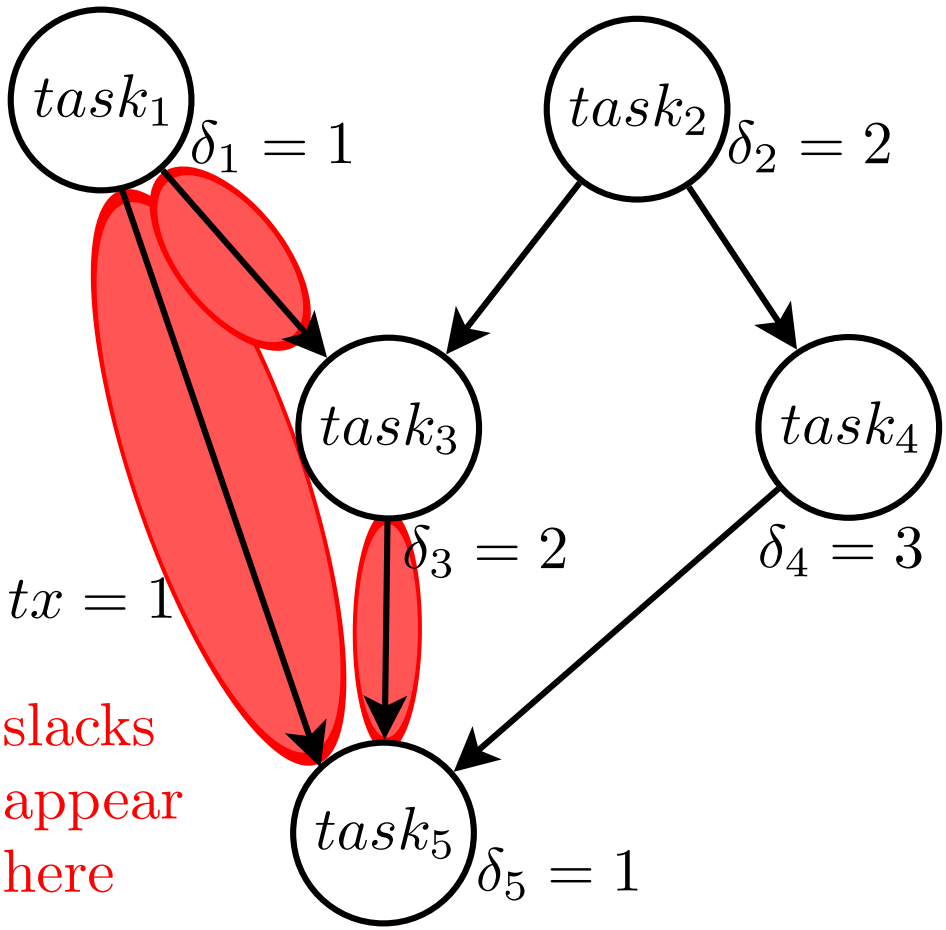
For human beings: an example



δ execution time; tx transmission time

Background (III): Slack concept (1)

For human beings: an example



- Slowest path:
  $\Theta = \frac{1}{\sum(\delta + tx)} = \frac{1}{8}$
  (Θ: inverse of execution time of slowest path)
- Slack: how much faster is a path w.r.t. another until a synchronisation point
  - $\mu_{1,3} = \frac{3-2}{8} = \frac{1}{8}$
  - $\mu_{1,5} = \frac{7-2}{8} = \frac{5}{8}$
  - $\mu_{3,5} = \frac{7-6}{8} = \frac{1}{8}$

δ execution time; tx transmission time

1542

Background (III): Slack concept (2)

For tough guys/gals – Maths fans

- Upper performance bound vs. exact analysis
  - Exact analysis needs reachability graph $\rightarrow$ NP-hard problem

Universidad Zaragoza

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Background (III): Slack concept (2)

For tough guys/gals – Maths fans

- Upper performance bound vs. exact analysis
  - Exact analysis needs reachability graph → NP-hard problem
- Use of Linear Programming (LP) techniques → polynomial complexity
- Based on Petri net theory: *tight marking* ($\tilde{\mathbf{m}}$) [p. 3 on the paper]
  - Further reading: google "RJ-EPEW-10"

$$\tilde{\mathbf{m}}(p) \geq \delta(p^\bullet) \cdot \Theta \rightarrow \mathbf{m}(p) = \delta(p^\bullet) \cdot \Theta + \mu(p) \quad (1)$$

$\delta(p^\bullet)$ delay of transition after place p;
Θ inverse of execution time of slowest path;
$\mu(p)$ slack of place p

Universidad Zaragoza

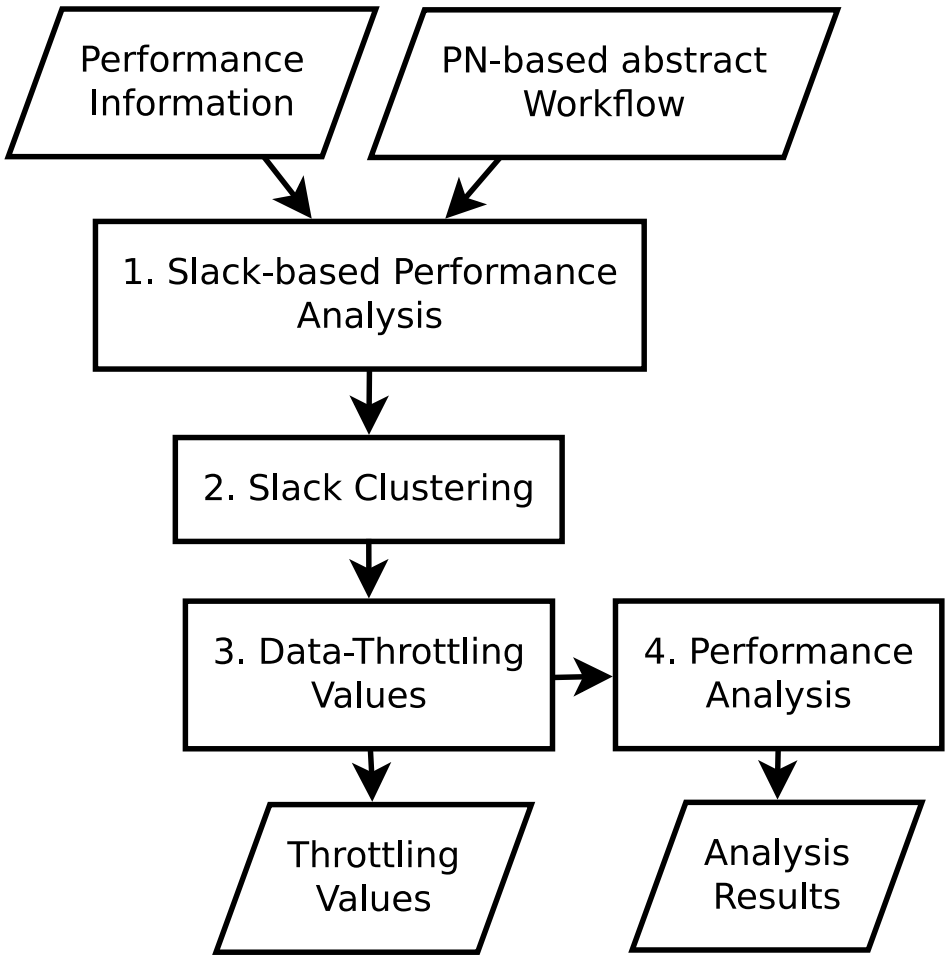
CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Outline

Universidad Zaragoza

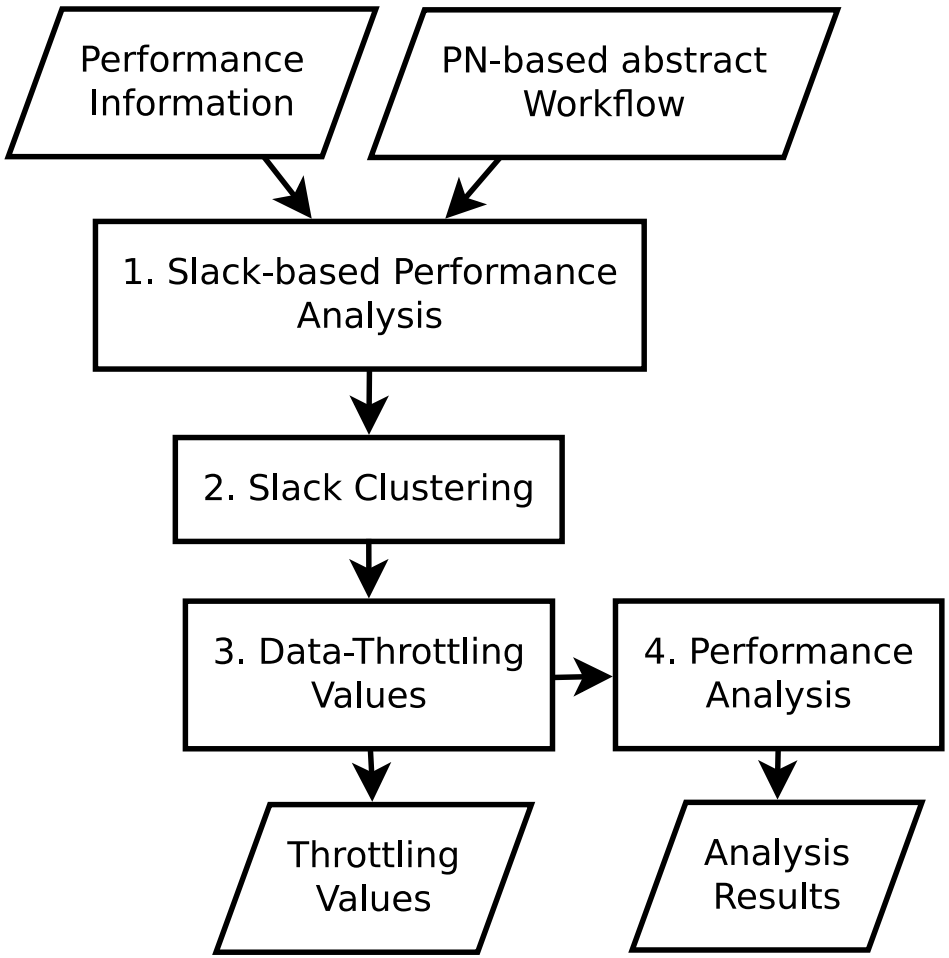
CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Automating Data-Throttling Analysis (I)



- Inputs: Performance estimation (i.e., DAX annotations) + PN-based model
- Outputs: Data-throttling values + Analysis results

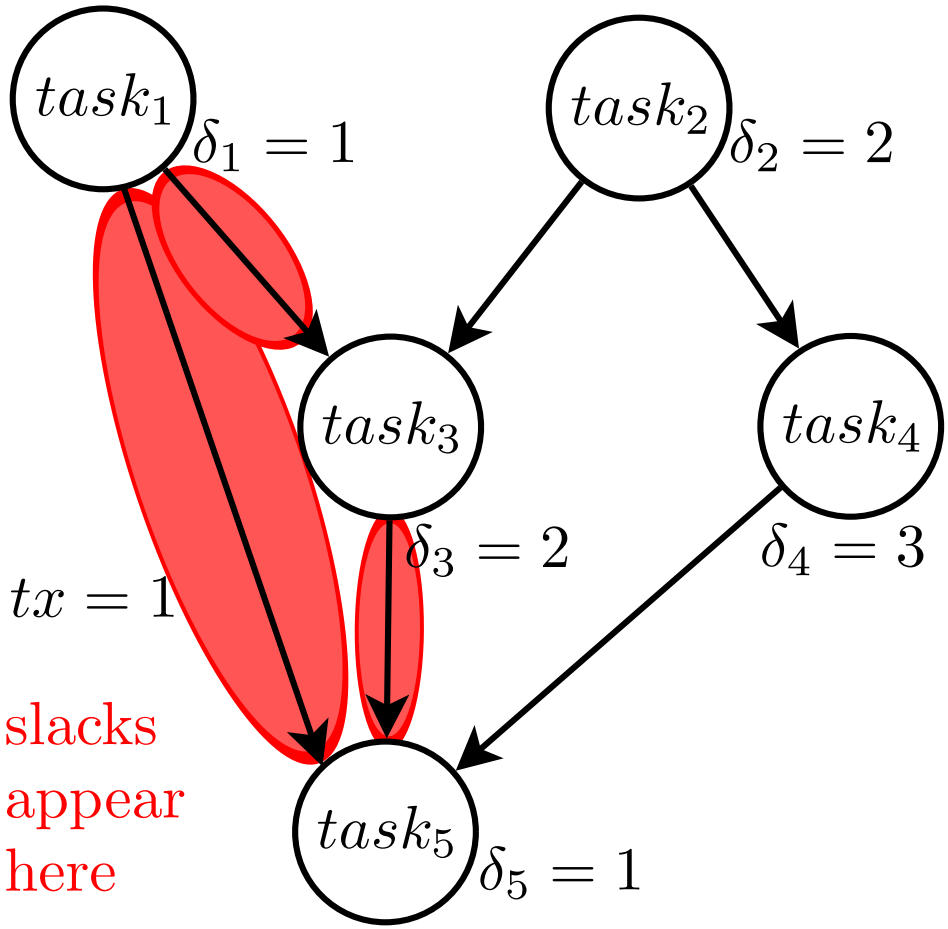
Automating Data-Throttling Analysis (I)



- Inputs: Performance estimation (i.e., DAX annotations) + PN-based model
- Outputs: Data-throttling values + Analysis results
- 4 steps
  1. Compute slack values
  2. Cluster slacks
     - Why? In the next slide!
  3. Compute data-throttling values
  4. Performance analysis
     - With and w/out data-throttling

Automating Data-Throttling Analysis (II)

The need of clustering slacks

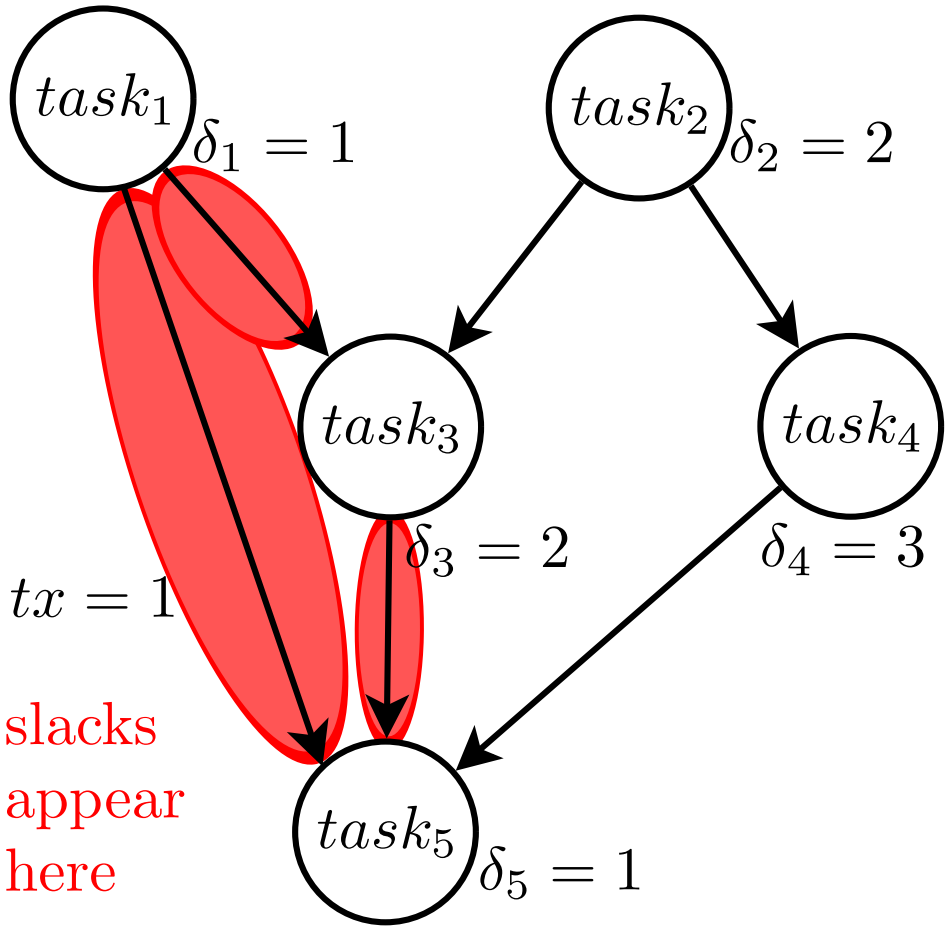


- Influence between slacks
- Order of adjusting IS IMPORTANT

Zaragoza

Automating Data-Throttling Analysis (II)

The need of clustering slacks

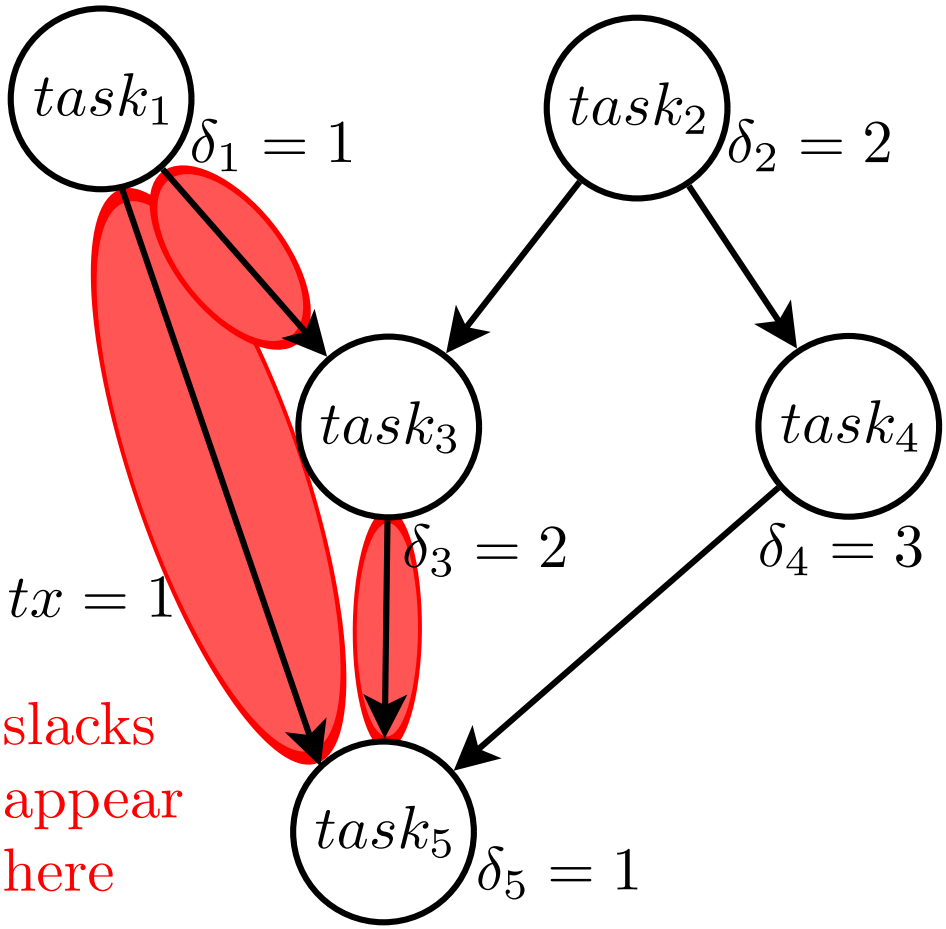


- Influence between slacks
- Order of adjusting IS IMPORTANT
- From output to input
- Cluster slacks, then:
  1. Compute data-throttling values
  2. Recompute slacks values

Zaragoza

Automating Data-Throttling Analysis (II)

The need of clustering slacks



- Influence between slacks
- Order of adjusting IS IMPORTANT
- From output to input
- Cluster slacks, then:
  1. Compute data-throttling values
  2. Recompute slacks values
- In the example:
  1. $\{\mu_{3,5}, \mu_{1,5}\}$
  2. $\mu_{1,3}$

Zaragoza

Automating Data-Throttling Analysis (III)

Deriving data-throttling values from slack

- $\mu(p) \rightarrow$ some delay may be added on the path
- $tx_{new} = tx_{old} + \alpha,\ \alpha = \frac{\mu(p)}{\Theta}$

Automating Data-Throttling Analysis (III)

Deriving data-throttling values from slack

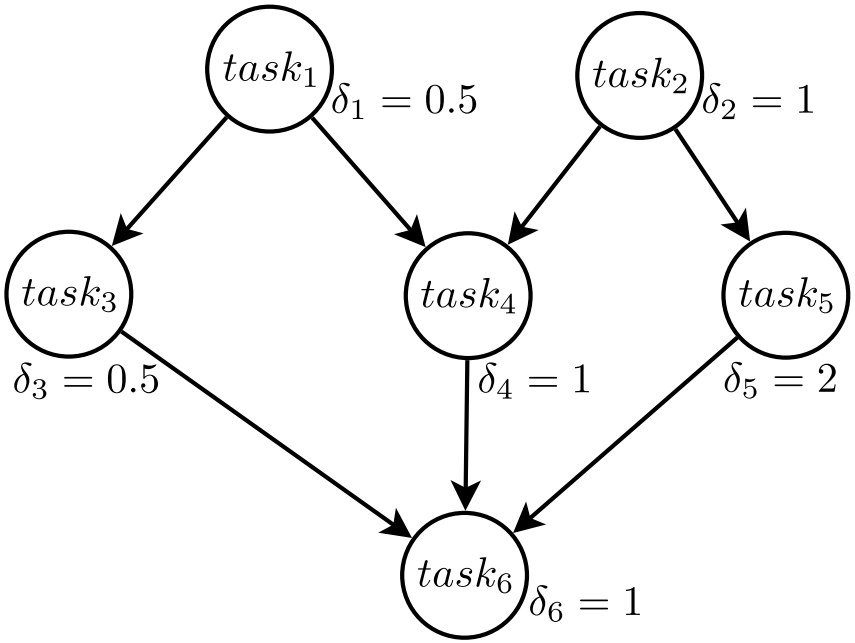
- $\mu(p) \rightarrow$ some delay may be added on the path
- $tx_{new} = tx_{old} + \alpha,\ \alpha = \frac{\mu(p)}{\Theta}$
- $tx = \frac{data\ size}{link\ bandwidth(BW)}$

$$BW_{new} = \frac{1}{\frac{1}{BW_{old}} + \frac{\mu(p)}{\Theta \cdot data\ size}} \quad (2)$$

μ slack; tx transmission time;
Θ inverse of execution time of slowest path

Automating Data-Throttling Analysis (IV)

Applying to an example



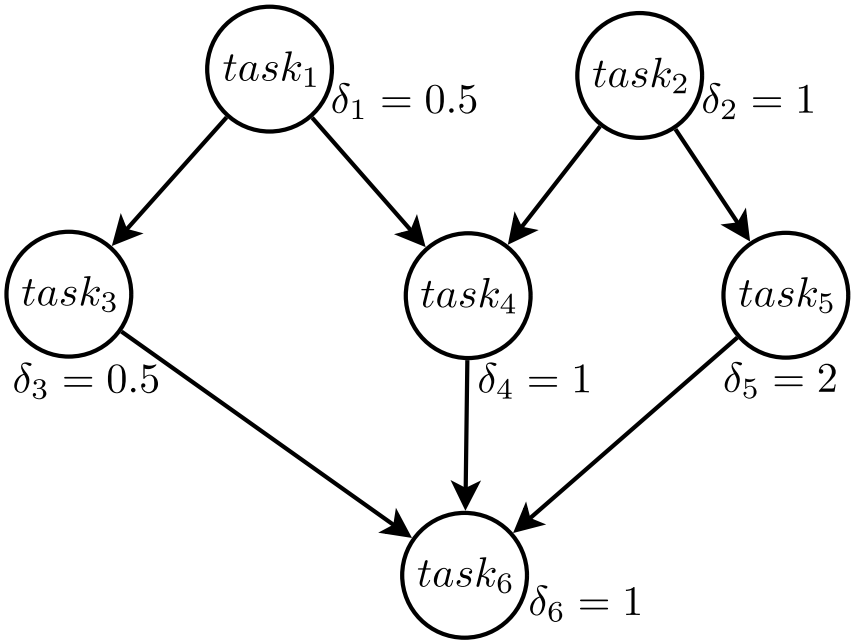
Recall: slacks on synchronisation points

Assumptions

- BW=100Mbps, latency $1e^{-4}s$
- Data-sets equal to 10MiB
- Dedicated network topology

Automating Data-Throttling Analysis (IV)

Applying to an example



Recall: slacks on synchronisation points

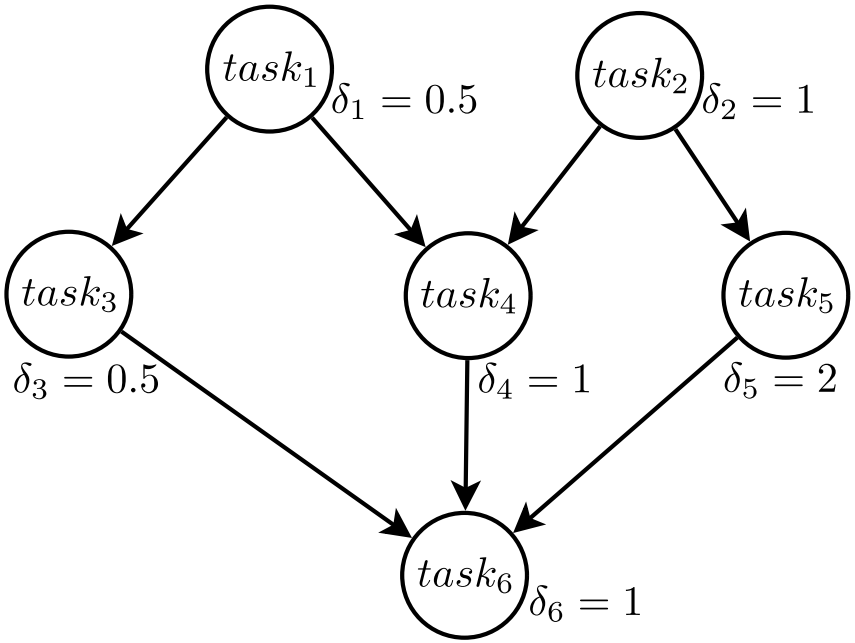
Assumptions

- BW=100Mbps, latency $1e^{-4}s$
- Data-sets equal to 10MiB
- Dedicated network topology

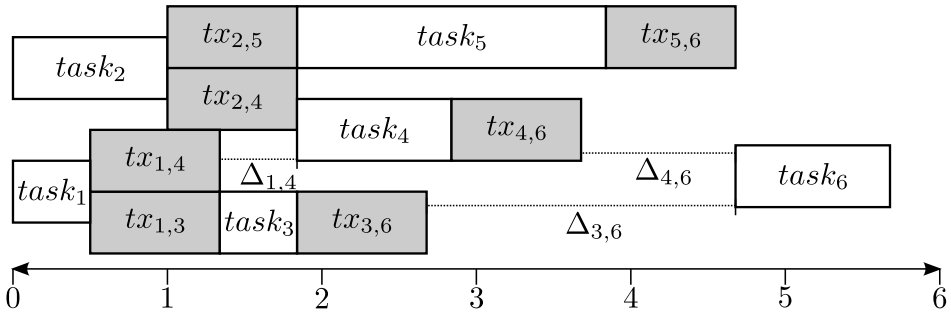
- Slowest path: $2 \rightarrow 5 \rightarrow 6$
- Slacks: $\mu_{1,4}, \mu_{3,6}, \mu_{4,6}$

Automating Data-Throttling Analysis (IV)

Applying to an example



Recall: slacks on synchronisation points



Makespan: 5.6779 seconds

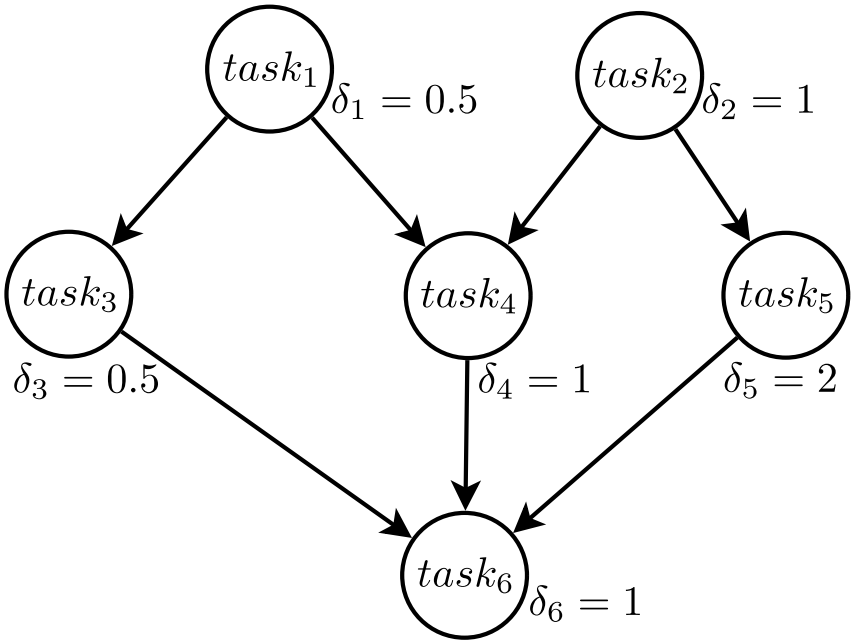
Assumptions

- BW=100Mbps, latency $1e^{-4}s$
- Data-sets equal to 10MiB
- Dedicated network topology

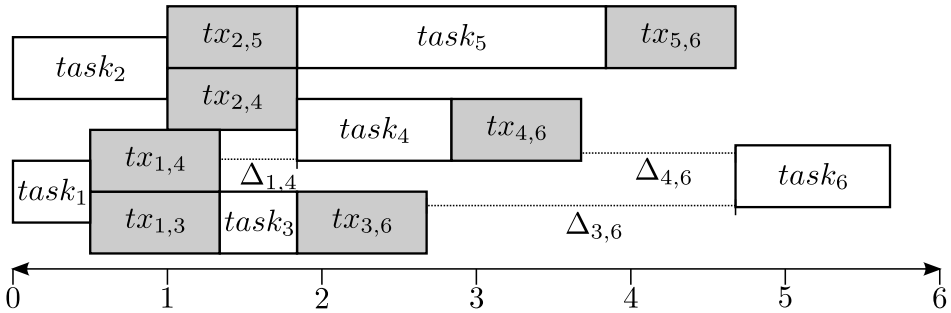
- Slowest path: $2 \rightarrow 5 \rightarrow 6$
- Slacks: $\mu_{1,4}, \mu_{3,6}, \mu_{4,6}$

Automating Data-Throttling Analysis (IV)

Applying to an example



Recall: slacks on synchronisation points



Makespan: 5.6779 seconds

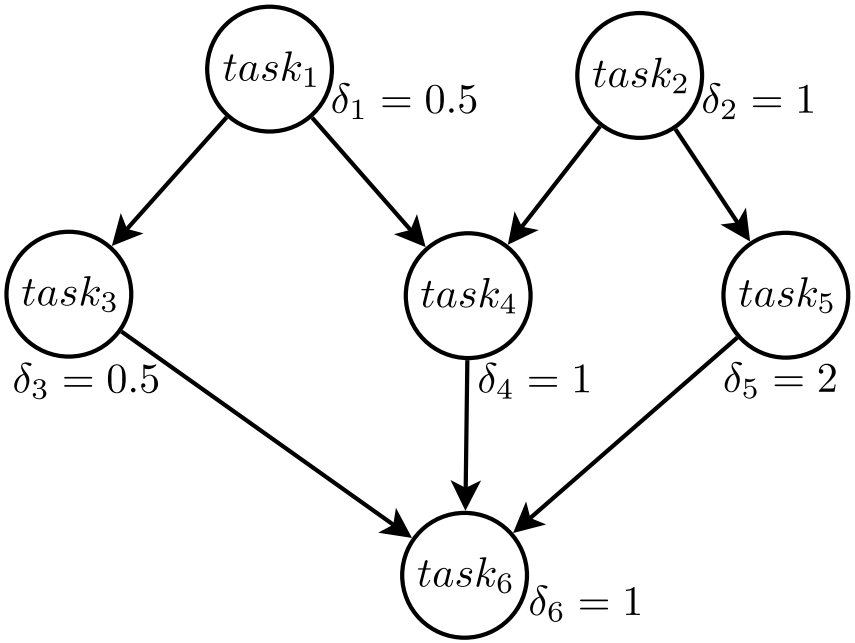
Assumptions

- BW=100Mbps, latency $1e^{-4}s$
- Data-sets equal to 10MiB
- Dedicated network topology

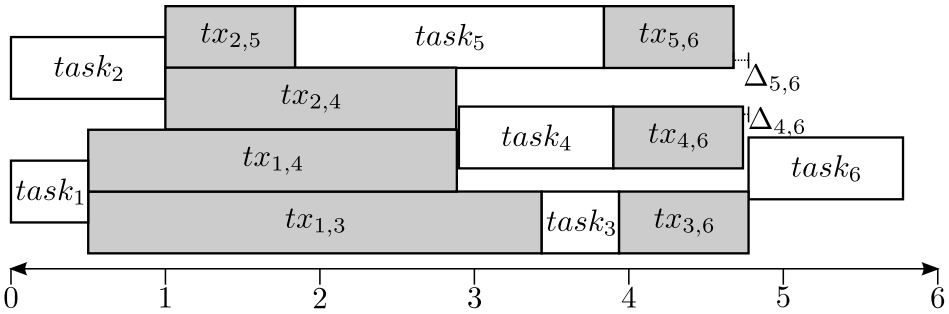
- Slowest path: $2 \rightarrow 5 \rightarrow 6$
- Slacks: $\mu_{1,4}, \mu_{3,6}, \mu_{4,6}$
  - $1 \rightarrow 4$ adjust to 28.57%
  - $3 \rightarrow 6$ adjust to 35.15%
  - $4 \rightarrow 6$ adjust to 44.55%

Automating Data-Throttling Analysis (IV)

Applying to an example



Recall: slacks on synchronisation points



Makespan: 5.7750 seconds

(negligent impact on makespan BUT input buffer waiting times reduced)

Assumptions

- BW=100Mbps, latency $1e^{-4}s$
- Data-sets equal to 10MiB
- Dedicated network topology

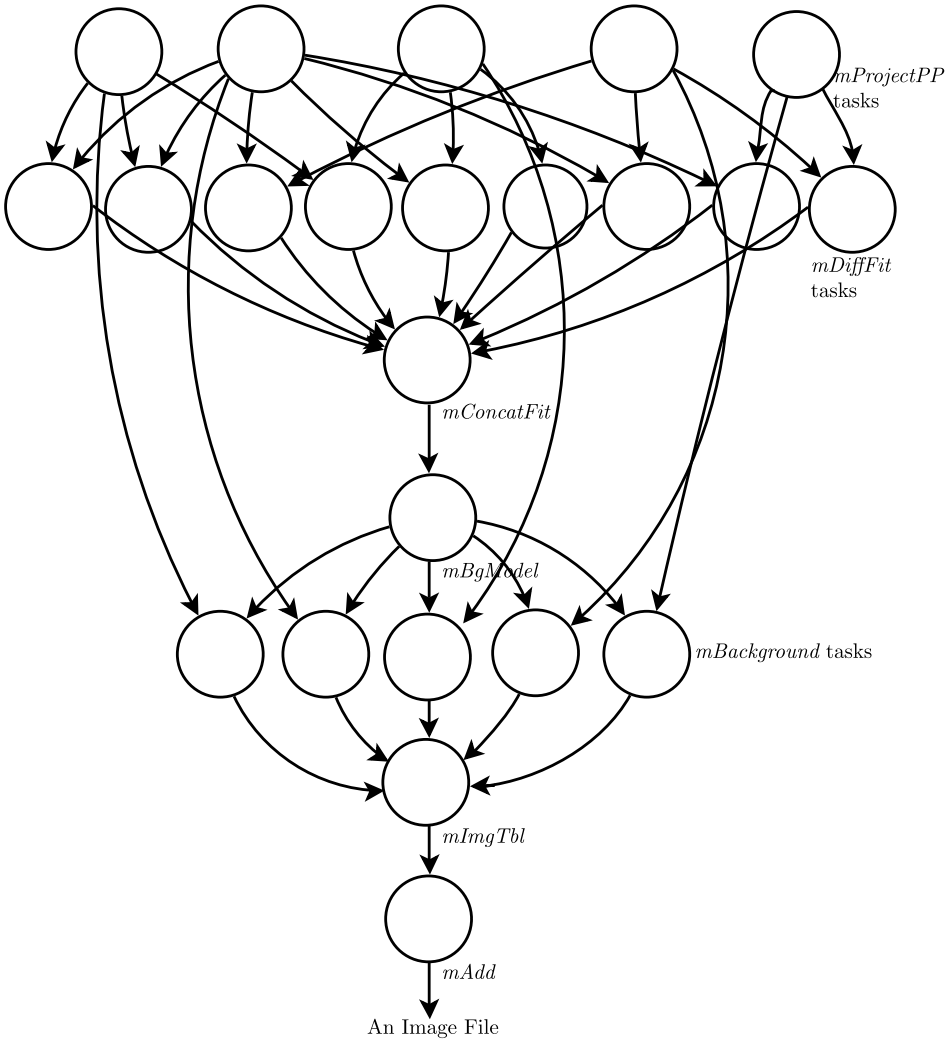
- Slowest path: $2 \rightarrow 5 \rightarrow 6$
- Slacks: $\mu_{1,4}, \mu_{3,6}, \mu_{4,6}$
  - $1 \rightarrow 4$ adjust to 28.57%
  - $3 \rightarrow 6$ adjust to 35.15%
  - $4 \rightarrow 6$ adjust to 44.55%

Outline

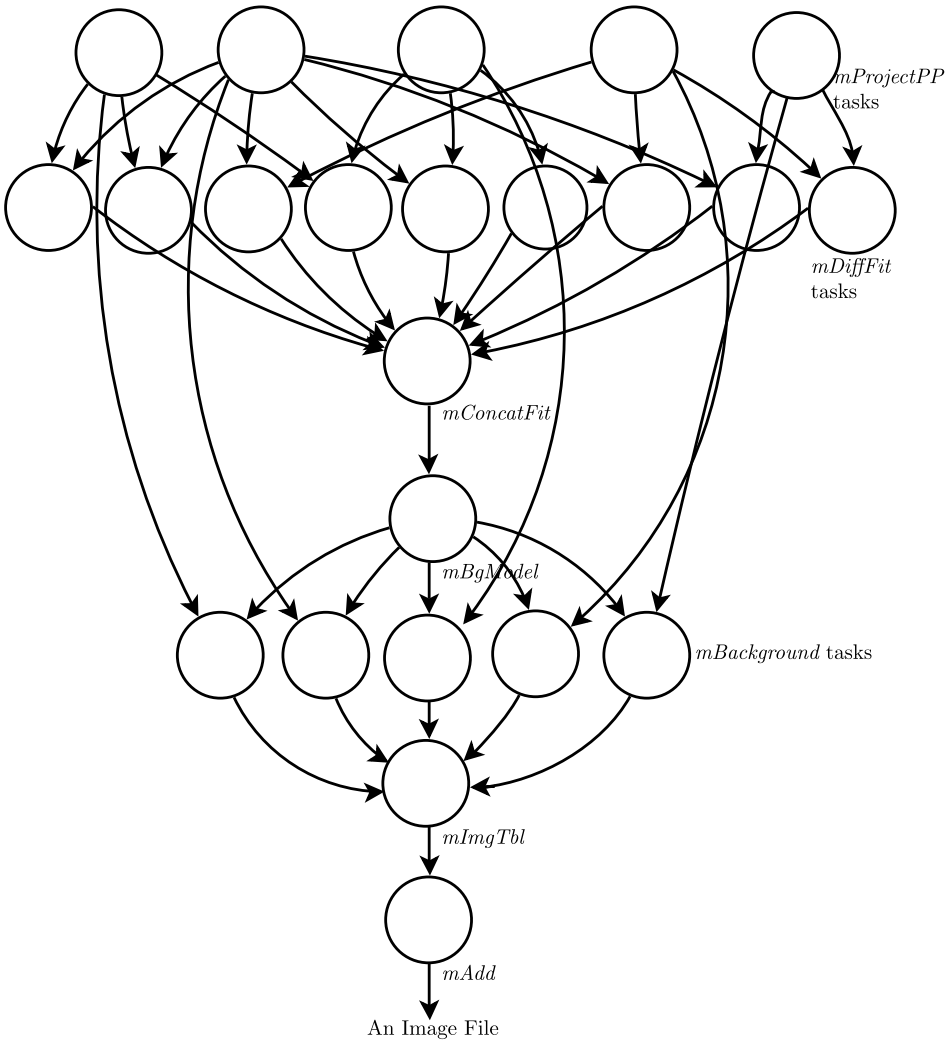
Universidad Zaragoza

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Experiments and Results (I): Description

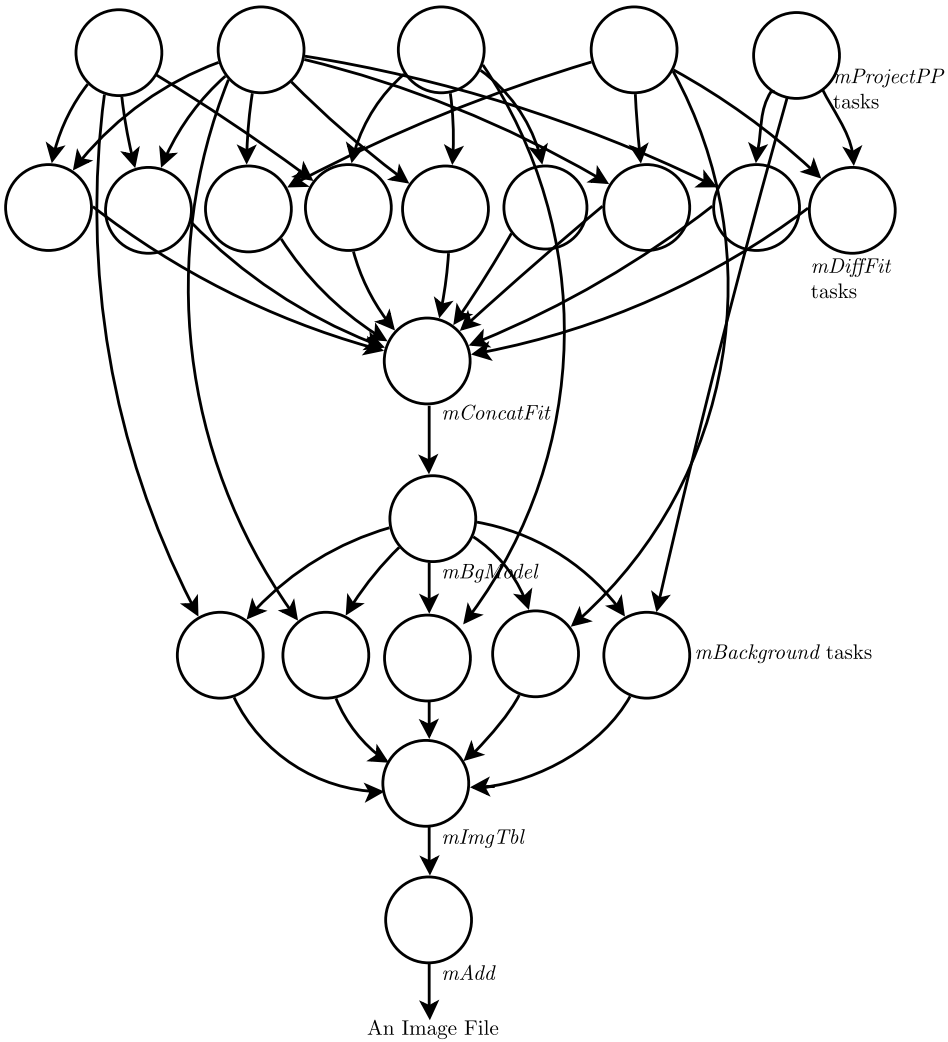


Experiments and Results (I): Description



- Montage Workflow – 5inputs
- Performance estimation from DAX (Pegasus Wf System)

Experiments and Results (I): Description

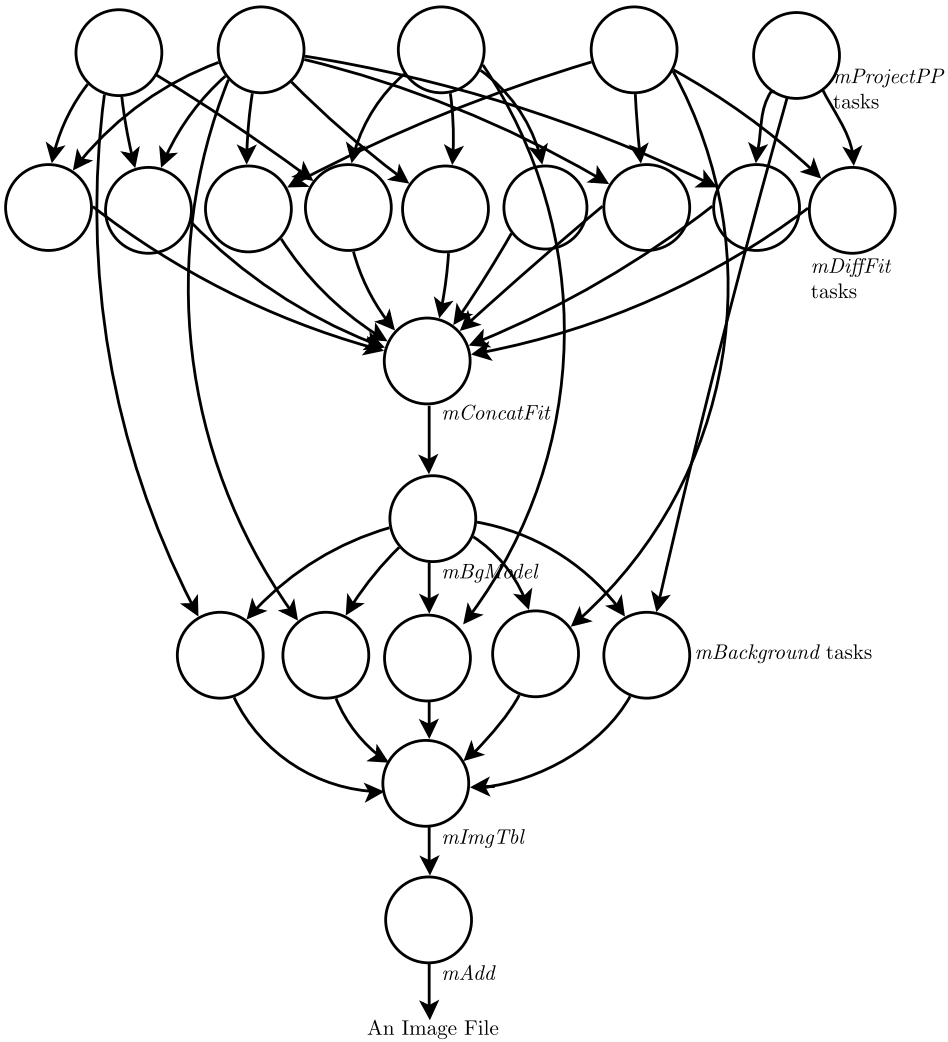


- Montage Workflow – 5inputs
- Performance estimation from DAX (Pegasus Wf System)
- Network topologies assumed:
  - Single vs. dedicated data-link (throttling in dedicated)

Universidad Zaragoza 1542

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Experiments and Results (I): Description



- Montage Workflow – 5inputs
- Performance estimation from DAX (Pegasus Wf System)
- Network topologies assumed:
  - Single vs. dedicated data-link (throttling in dedicated)

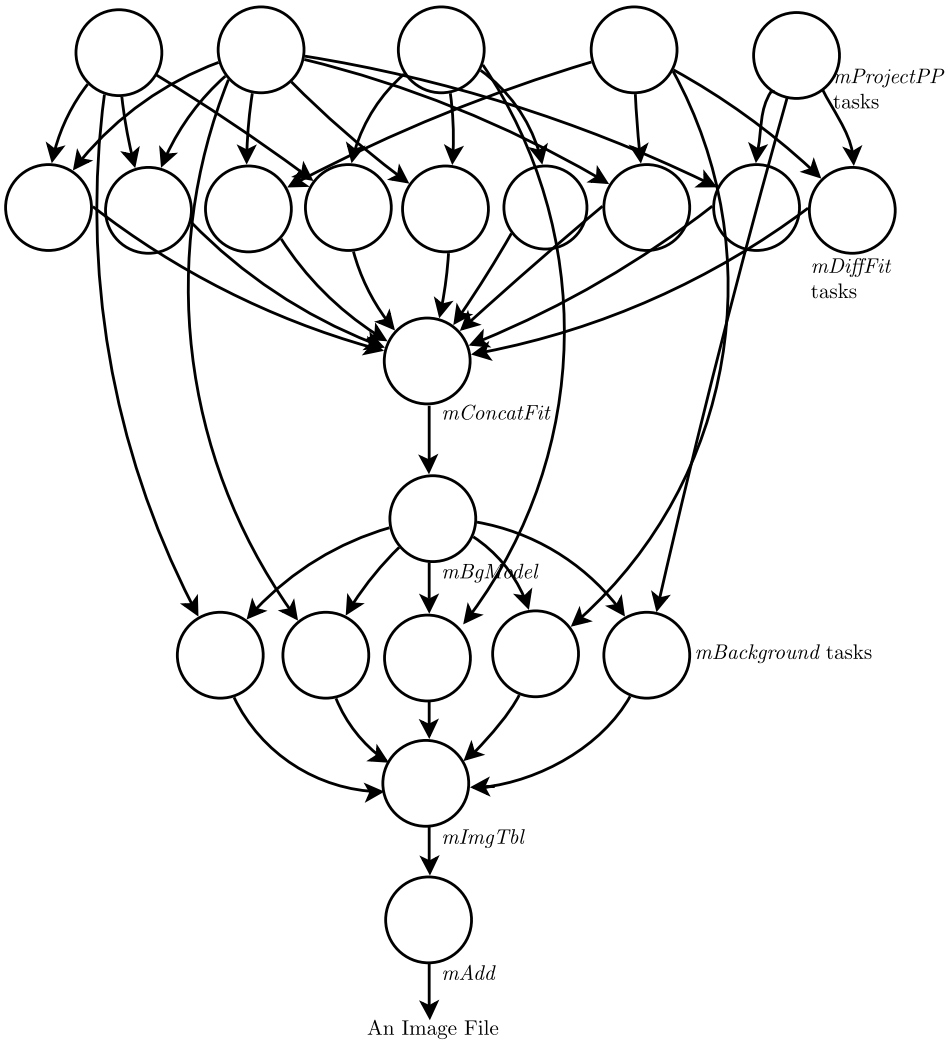
Tools used

- Slack computation: MATLAB
- Performance analysis: SimGrid

Universidad Zaragoza 1542

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Experiments and Results (I): Description



- Montage Workflow – 5inputs
- Performance estimation from DAX (Pegasus Wf System)
- Network topologies assumed:
  - Single vs. dedicated data-link (throttling in dedicated)

Tools used

- Slack computation: MATLAB
- Performance analysis: SimGrid

Experiments performed

- Impact on makespan
- Input buffers & net BW usage

Experiments and Results (II): Experiments (1)

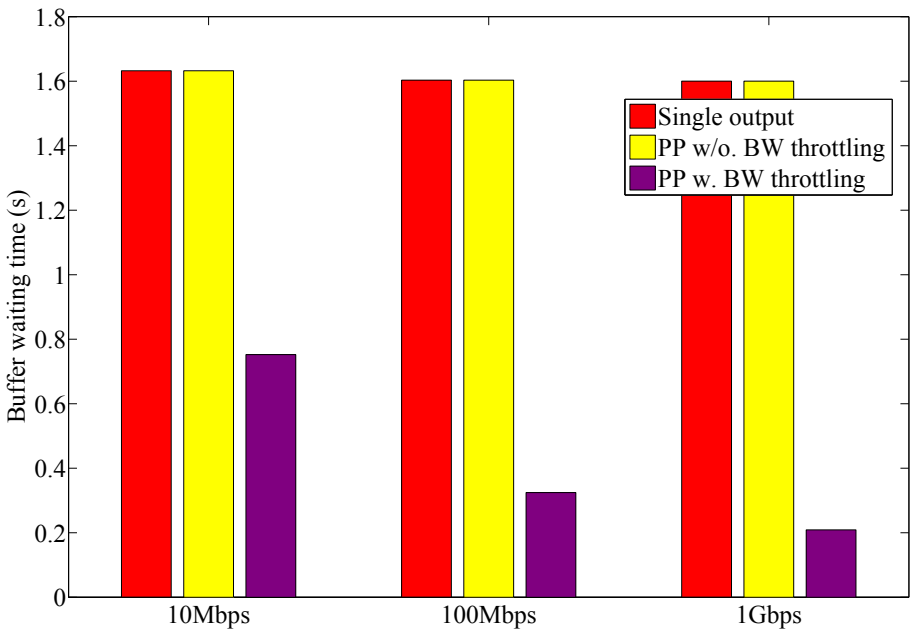
Impact on the Workflow Makespan

Network topology	Network bandwidth		
	10Mbps	**100Mbps**	**1Gbps**
Single output	193.20*s*	61.18*s*	47.98*s*
PP without BW throttling	153.15*s*	57.17*s*	47.58*s*
PP with BW throttling	153.32*s*	57.21*s*	47.58*s*

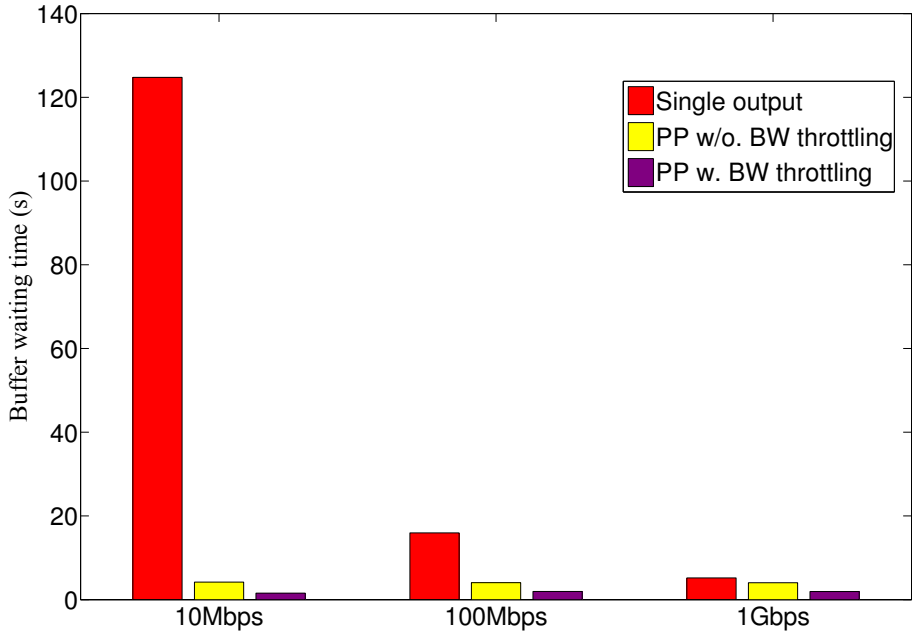
- Data-throttling looses (insignificantly)
- ↑ net. BW implies smaller impact on makespan
- Correlation $\frac{\textit{data transmission}}{\textit{computation}}$ (as indicated by Park & Humphrey)
  - Verified by our results

Experiments and Results (II): Experiments (2)

Input Buffers and Network Bandwidth Usage – some plots



(a) Workflow task *mImgTbl*



(b) Workflow task *mConcatFit*

- Data-throttling has great impact on input buffers
- Outperforms both other topologies

Outline

Universidad Zaragoza

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Related Work

- PNs already used in scientific workflow community
  - GWorkflowDL, GridFlow, FlowManager
- Overhead analysis (Nerieri et al.)
  - Load imbalance + data movement
- Data throttling issue (Park & Humphrey)

Related Work

- PNs already used in scientific workflow community
  - GWorkflowDL, GridFlow, FlowManager
- Overhead analysis (Nerieri et al.)
  - Load imbalance + data movement
- Data throttling issue (Park & Humphrey)
- Performance analysis
  - Hybrid Bayesian-neural network (Duan et al., predicts execution time of tasks)
  - Parametrised PN-based model (Tolosana-Calasanz et al.)
- Structural analysis (Van der Alst & Van Hee)
  - WF-nets
    - Operations: sequence, choice, synchroniser, fork, merge
    - Analysis: correctness, deadlock analysis, liveness

How can I use this *fancy* approach?

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

How can I use this *fancy* approach?

Workflows characteristics

- Synchronisation points/merge tasks
- DAX (DAG in XML format) $\rightarrow$ PNML (PN in XML format)
  - Automatic transformation

How can I use this *fancy* approach?

Workflows characteristics

- Synchronisation points/merge tasks
- DAX (DAG in XML format) → PNML (PN in XML format)
  - Automatic transformation

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- generated: 2008-09-24T14:28:09-07:00 -->
<!-- generated by: shishir [??] -->
<adag xmlns="http://pegasus.isi.edu/schema/DAX" xmlns:xsi="htt
xsi:schemaLocation="http://pegasus.isi.edu/schema/DAX http://p
index="0" name="test" jobCount="25" fileCount="0" childCount="
<!-- part 1: list of all referenced files (may be empty) -->
<!-- part 2: definition of all jobs (at least one) -->
  <job id="ID00000" namespace="Montage" name="mProjectPP" vers
    <uses file="region.hdr" link="input" register="true" trans
    <uses file="2mass-atlas-ID00000s-jID00000.fits" link="inpu
type="data" size="4222080"/>
    <uses file="p2mass-atlas-ID00000s-jID00000.fits" link="out
type="data" size="4167312"/>
    <uses file="p2mass-atlas-ID00000s-jID00000_area.fits" link
type="data" size="4167312"/>
  </job>
```

```
<?xml version="1.0" encoding="iso-8859-1"?>
<pnml>
<net id="Net-One" type="P/T net">
<place id="P0">
<graphics>
<position x="180.0" y="50.0"/>
</graphics>
<name>
<value>P0</value>
<graphics>
<offset x="-15.0" y="15.0"/>
</graphics>
</name>
<place-transitiontype>
<value>D</value>
</place-transitiontype>
<showLabel>
<position1>1</position1>
<position2>0</position2>
<position3>0</position3>
```

Outline

Universidad Zaragoza

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Conclusions and Future Work

Conclusions

- Transfer as-fast-as-possible is not always the best option
  - Network bandwidth misuse
  - Input buffer misuse

Conclusions and Future Work

Conclusions

- Transfer as-fast-as-possible is not always the best option
  - Network bandwidth misuse
  - Input buffer misuse
- Strategy for computing data-throttling values proposed
  - Main drawbacks
    - Needs previous performance information
    - Scalability

Future Work

- Test in more realistic environments
- Extend to other workflows
- Improve strategy computation (reduce its complexity)

Automating Data-Throttling Analysis for Data-Intensive Workflow

Ricardo J. Rodríguez, Rafael Tolosana-Calasanz, Omer F. Rana
{rjrodriguez, rafaelt}@unizar.es, o.f.rana@cs.cardiff.ac.uk

Universidad de Zaragoza
Zaragoza, Spain

Cardiff University
Cardiff, United Kingdom

May 15^{th}, 2012

CCGrid'12: 12^{th} IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing
Ottawa, Canada