Checkpointing In-Memory Data Analytics Applications with Panda

Bogdan Ghit
Joint work with Dick Epema

Delft University of Technology

About me

PhD degree from TU Delft, advised by Dick Epema

Thesis topic on scheduling data analytics frameworks

Hadoop

Fawkes

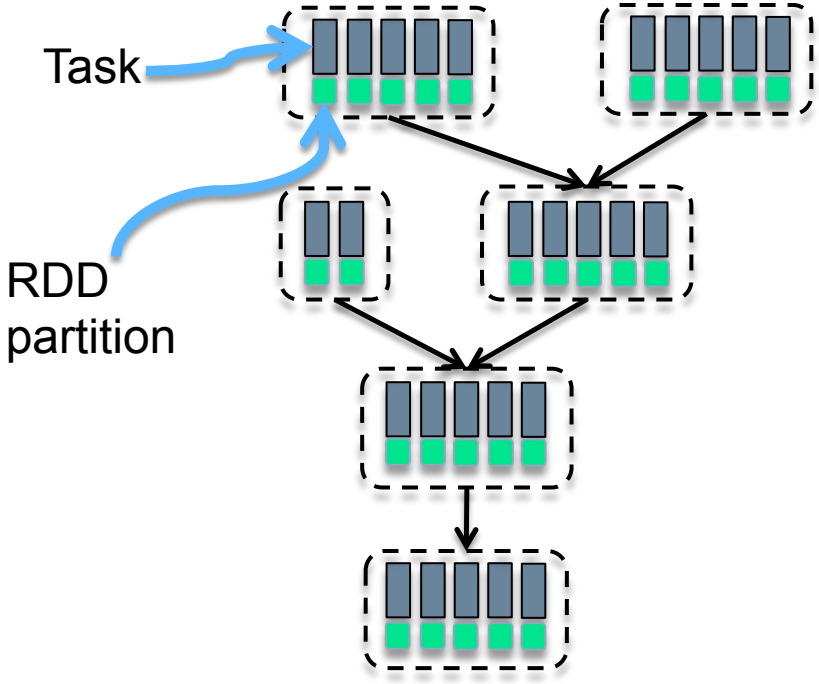
Tyrex

This talk is about Panda

Call for Efficiency

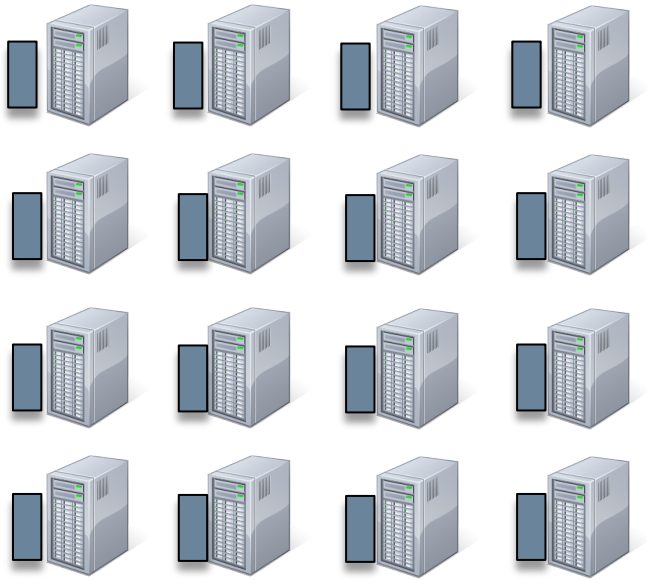
Large-scale data processing is now widespread

Spark Scheduling Model



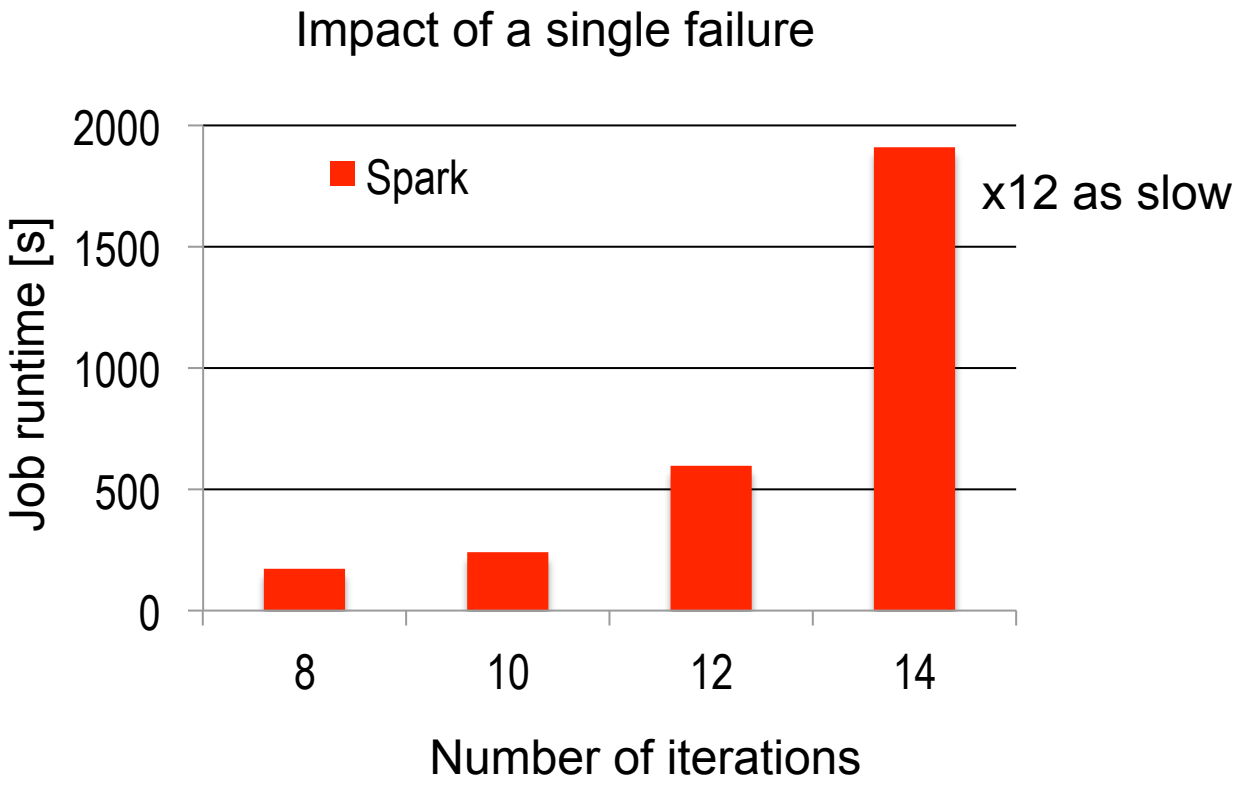
In-memory parallel computation

DAG-aware
scheduling

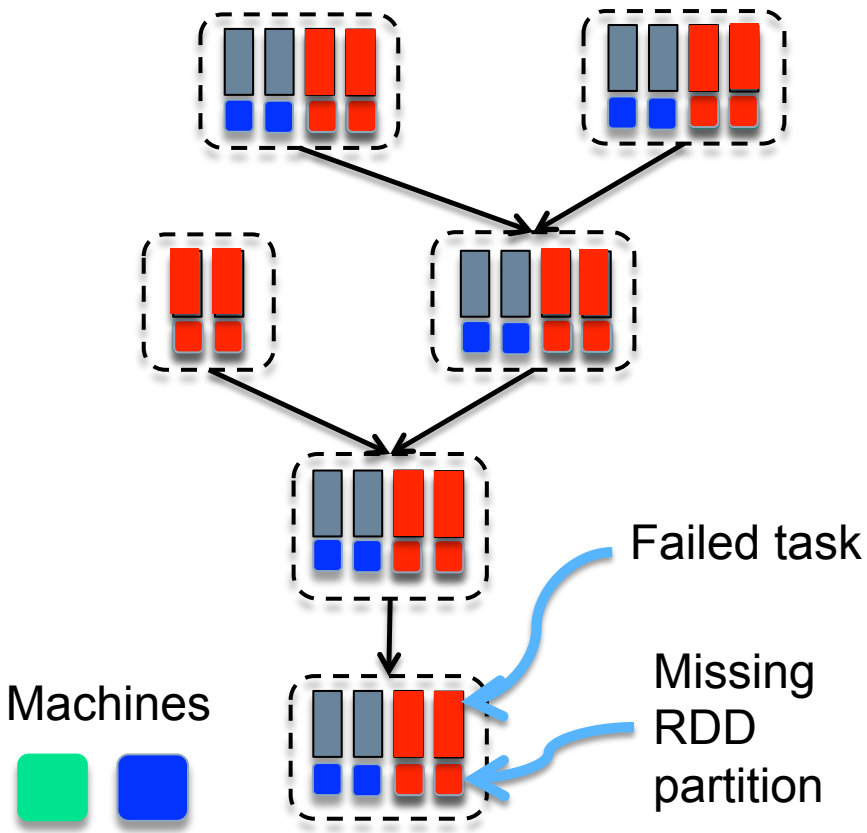


Task to slot allocation

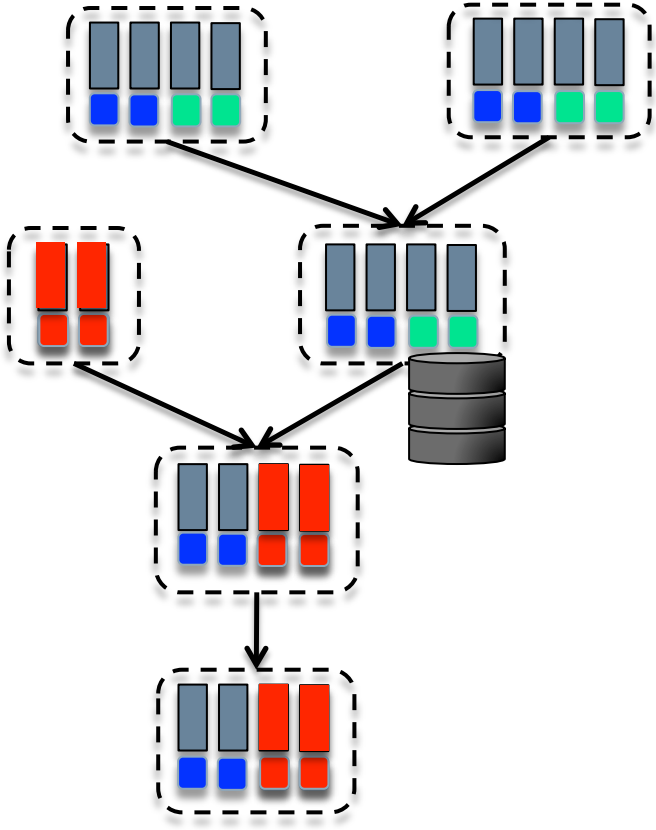
Resilient but Inefficient by Design



Recomputation vs. Checkpointing



Without checkpointing



With checkpointing

TU Delft

The Case for Checkpointing

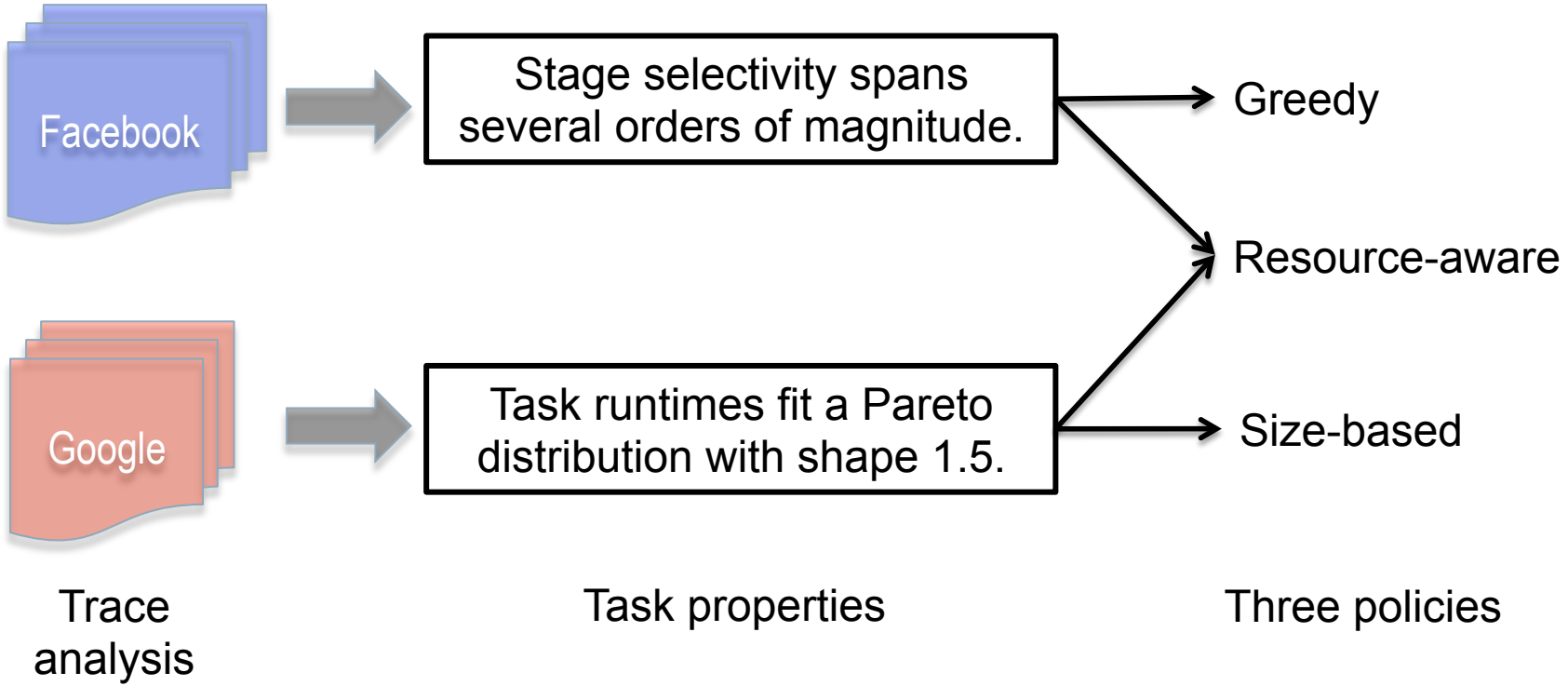
Failure	Rate (per year)	Machines lost	Downtime
Overheating	0.5	All	1-2 days
PDU	1	500-1000	6 h
Network rewiring	1	5%	2 days
Racks	20	40-80	1-6 h
Servers	1000	-	-
HDD	1000s	-	-

Jeff Dean, https://www.cs.cornell.edu/projects/ladis2009/talks/dean-keynote-ladis2009.pdf

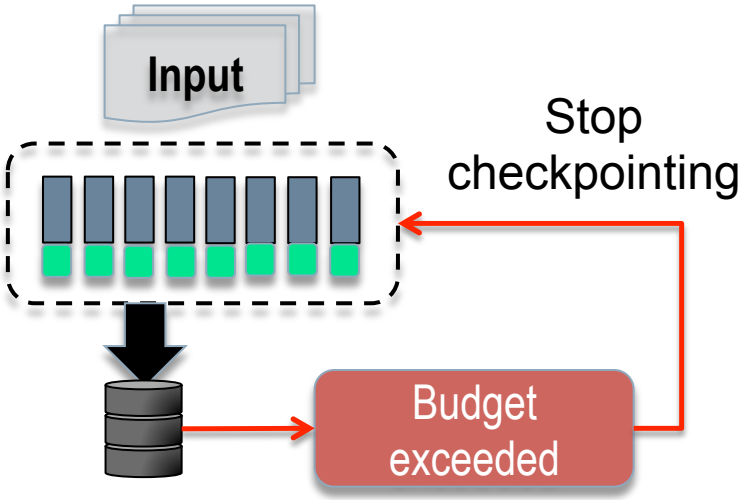
Where We Want to Go

Reduce the checkpointing problem to a task-selection problem

Policy Framework



Greedy Checkpointing



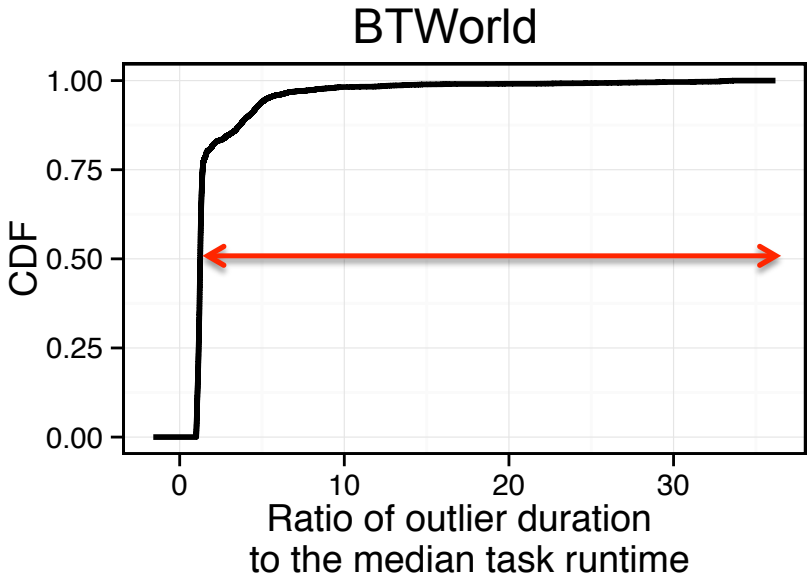
Task selection

- As many tasks as the budget allows
- Inflight checkpointing tasks are allowed to finish

The checkpointing budget

- Limits the checkpointing cost in each stage
- Set to a fraction of the total stage input

Size-based Checkpointing



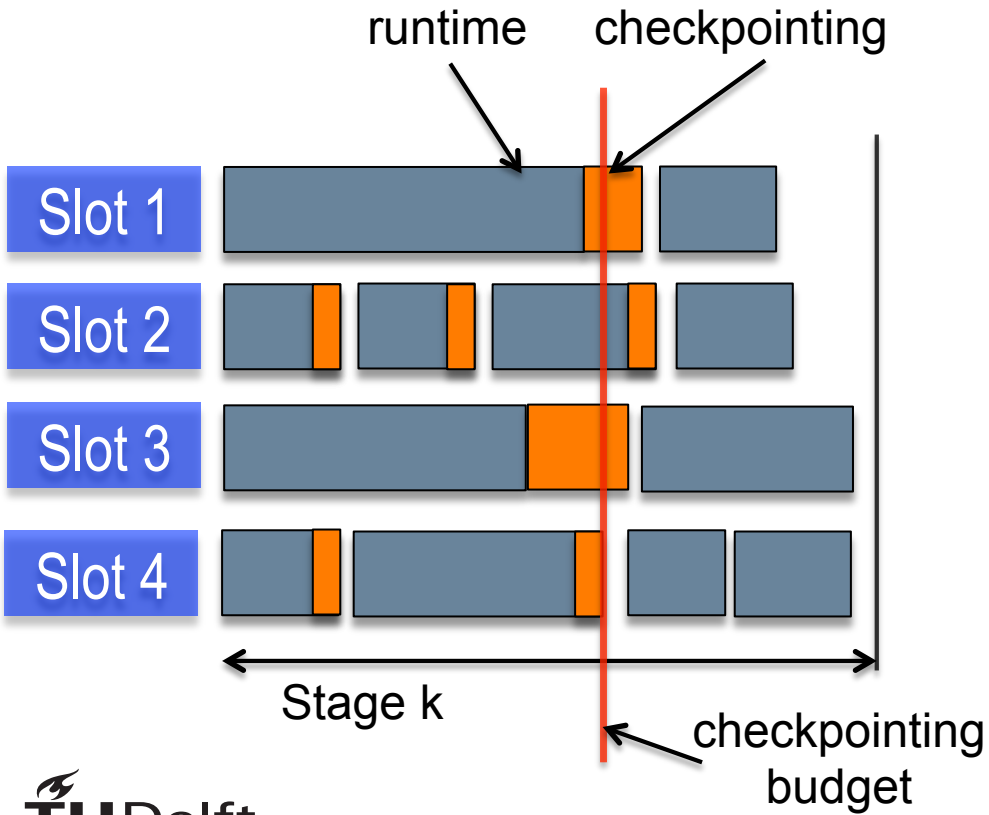
Task selection

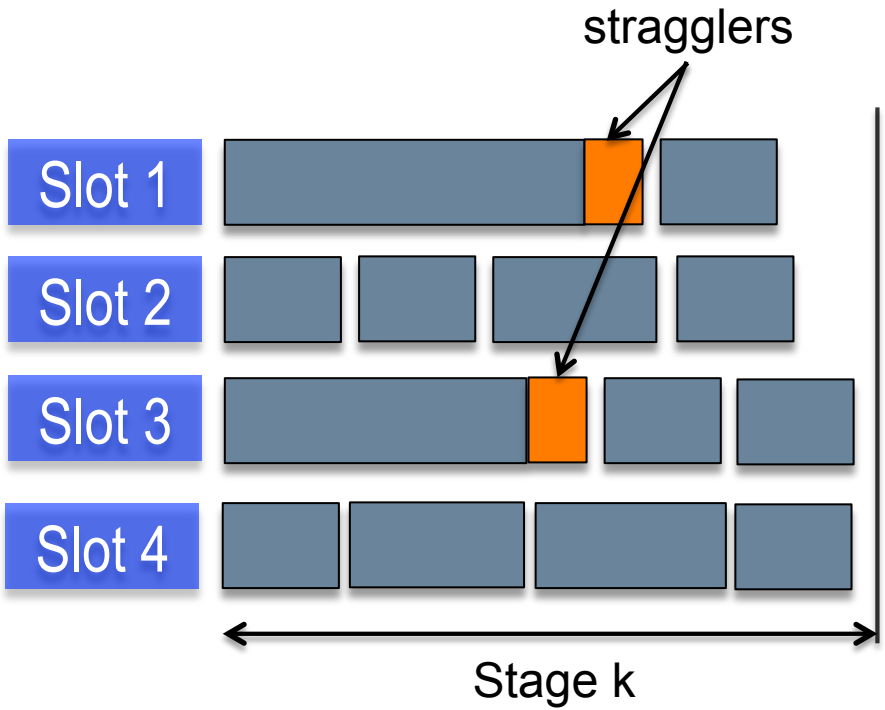
- Straggler tasks that run very slow
- Avoid recomputing time-consuming tasks

Identify stragglers

- Build up a history of task runtimes per job
- Tasks that run **m** times longer than the median

Greedy versus Size-based

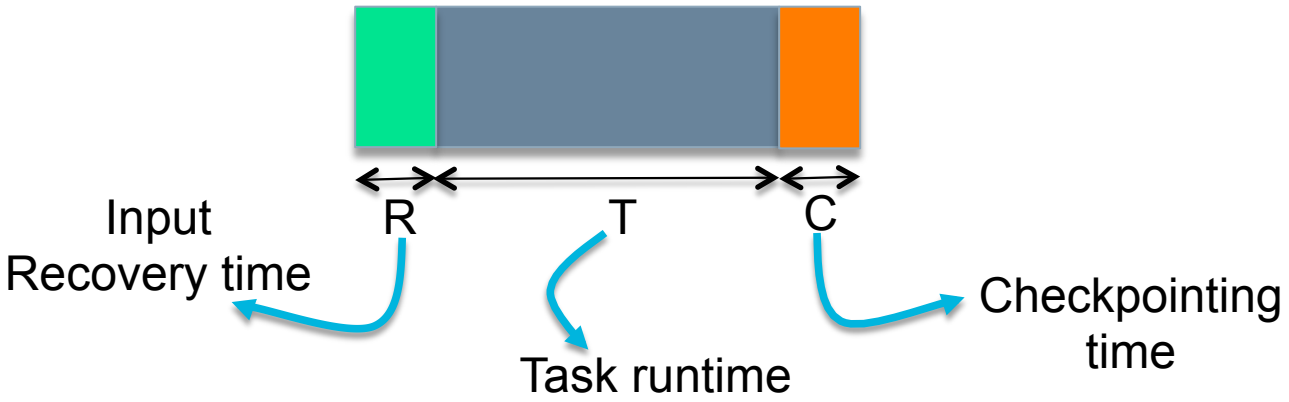




Resource-aware Checkpointing

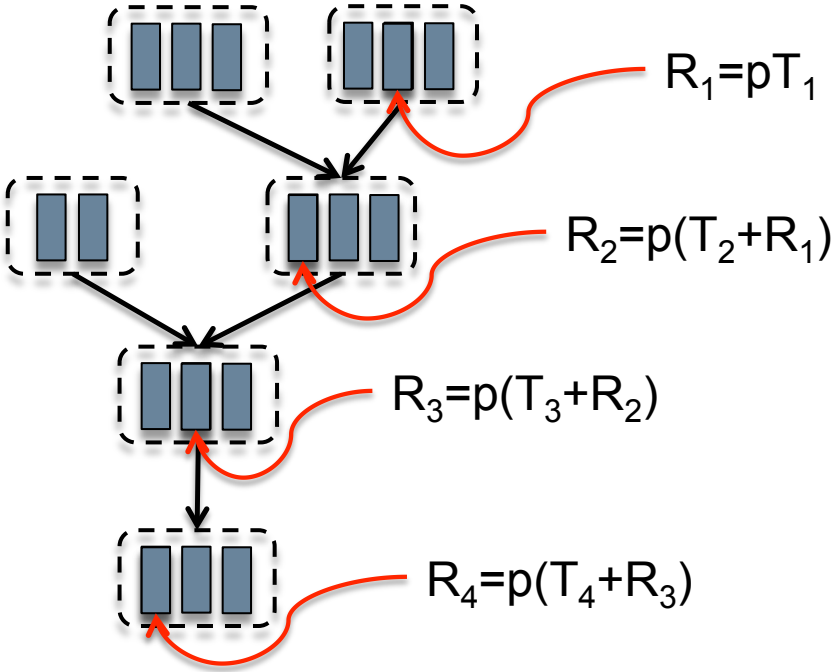
Task selection

- Estimated benefit outweighs the checkpointing cost

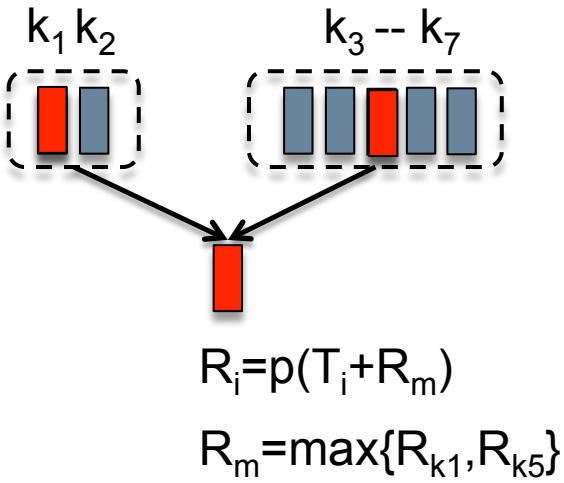


- Checkpoint tasks if: p (T + R) > C, p is the likelihood of failure

Estimating the Recomputation Cost



Single recovery path



Multiple recovery paths

Estimating the Checkpointing Cost

Checkpointing time depends on:

- Output size and write throughput
- Contention due to other tasks being checkpointed

Approximate method:

- Checkpoint the early waves of each stage
- Partial distribution of tasks checkpointing times

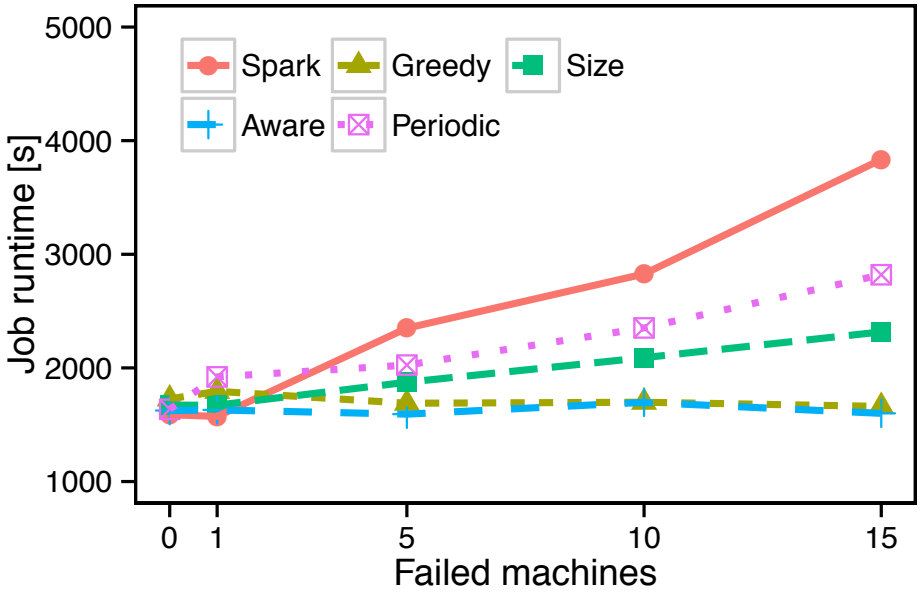
Experiment 1

How does the performance of our policies compare with periodic checkpointing?

Experiment details:

- 20-machine cluster
- BTWorld (500 GB)
- All policies

Takeaway: Greedy and Aware deliver constant job runtimes for the complete range of failures.



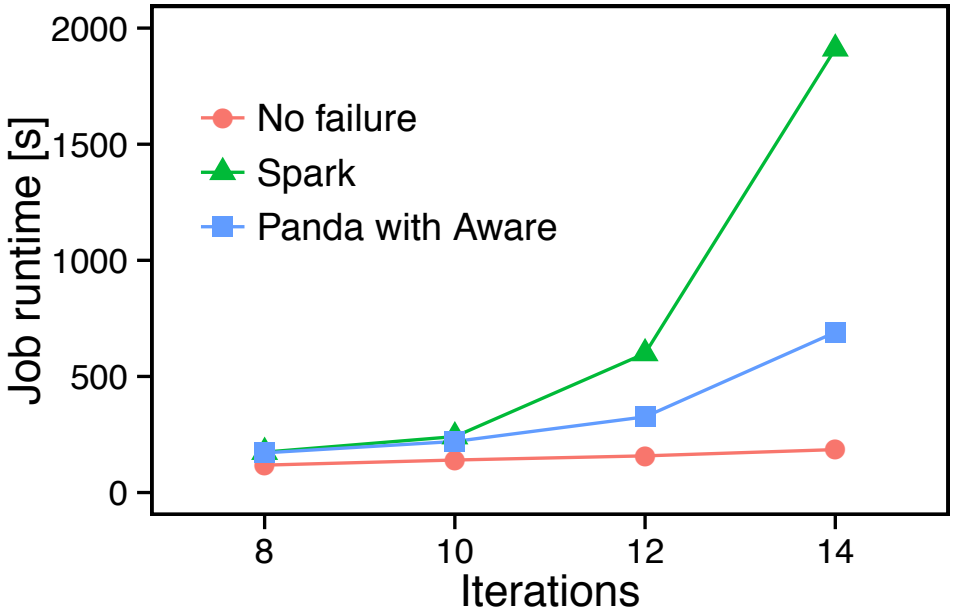
BTWorld

Experiment 2

What is the impact of the lineage length?

Experiment details:
- 5-machine cluster
- PageRank (1 GB)
- Aware policy

Takeaway: The Aware policy performs very well irrespective of the lineage length of the application.



Takeaways

*In-memory data analytics require **checkpointing**,*

*checkpointing is worthwhile **if you do it right**,*

using Panda is the right way to do it!

Bogdan Ghit and Dick Epema, "Better Safe than Sorry: Grappling with Failures of In-Memory Data Analytics Frameworks", ACM HPDC 2017