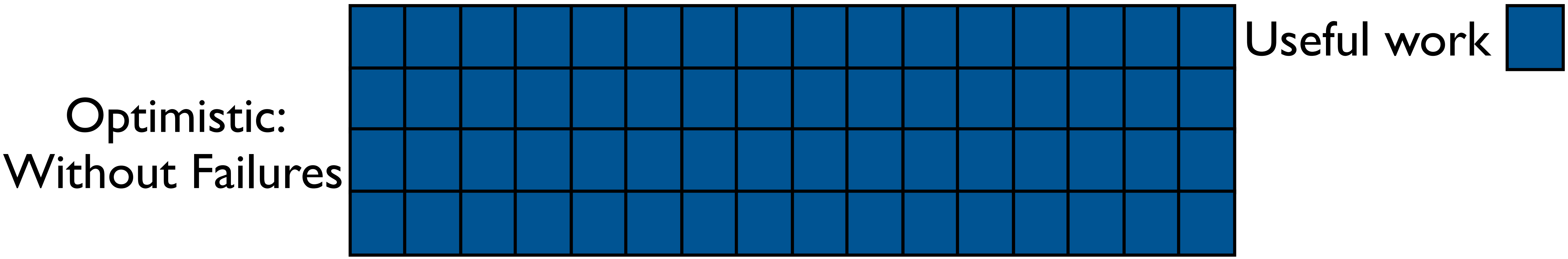
Improving the computing efficiency of HPC systems using a combination of proactive and preventive fault tolerance actions

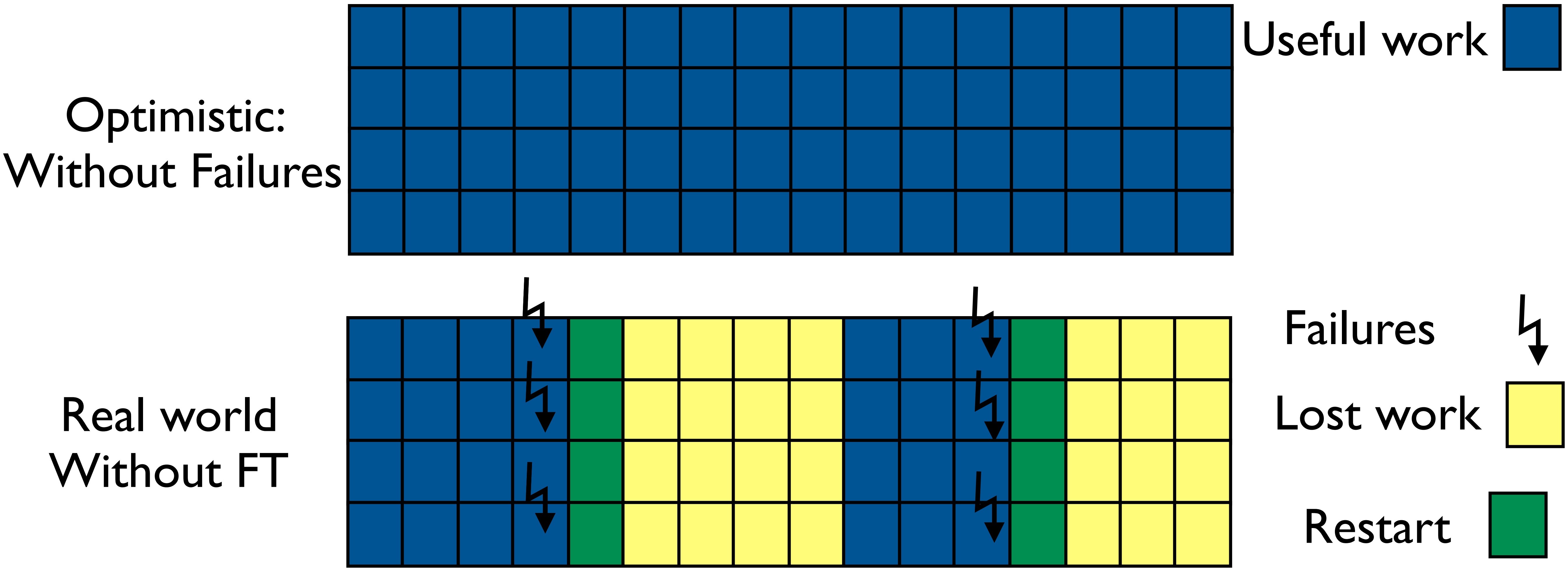
Mohmed Slim Bouguerra[1], Leonardo Bautista Gomez, Ana Gainaru, Franck Cappello

November 21, 2012

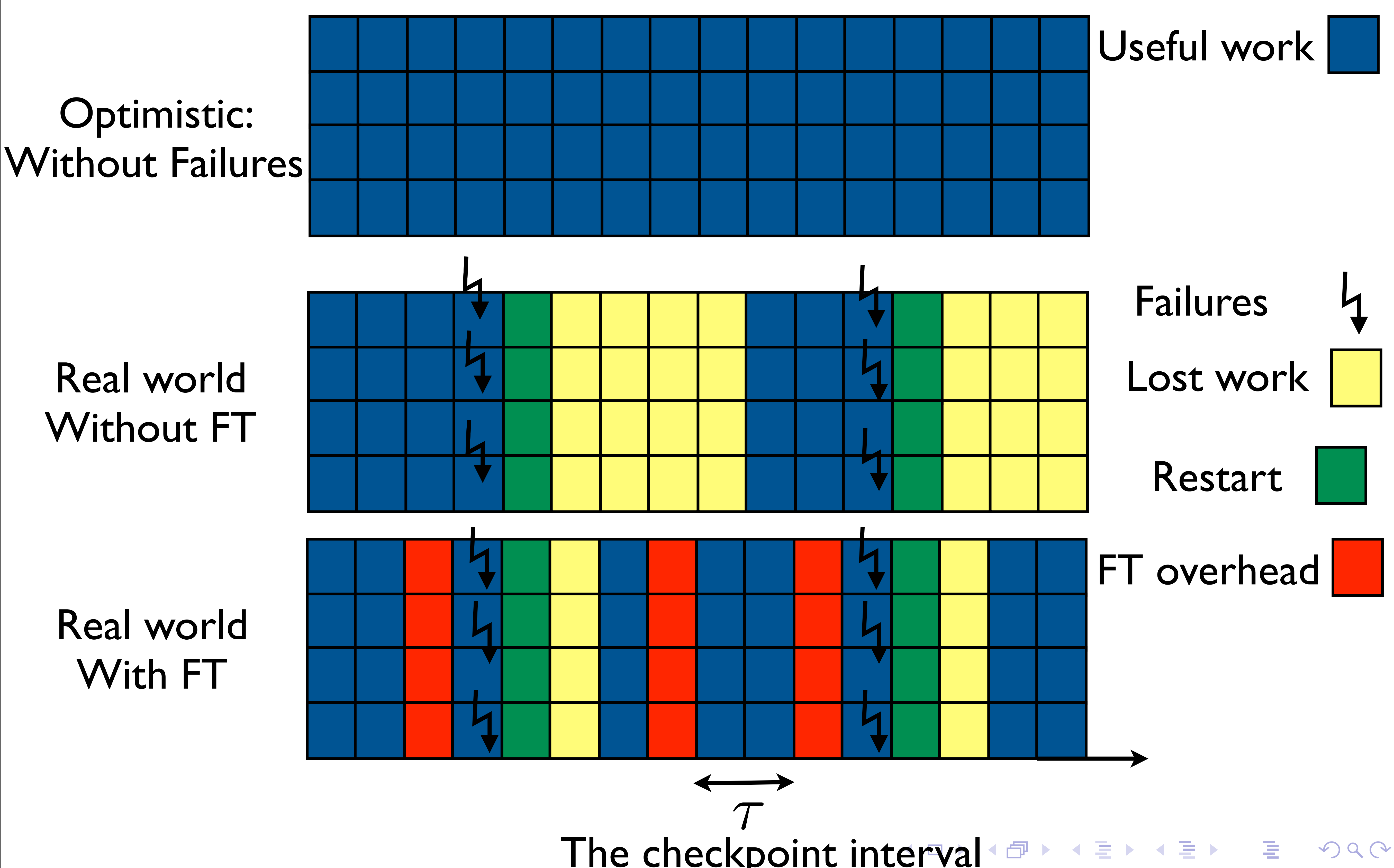
Problem statement



Problem statement



Problem statement



Classical checkpoint interval scheduling problem

The input:

- The checkpoint cost c
- The failure distribution $F(t)$
- The restart and down time cost R

Classical checkpoint interval scheduling problem

The input:

- The checkpoint cost c
- The failure distribution $F(t)$
- The restart and down time cost R

The output:

The optimal τ that minimizes the total useful work ?

Classical checkpoint interval scheduling problem

The input:

- The checkpoint cost c
- The failure distribution $F(t)$
- The restart and down time cost R

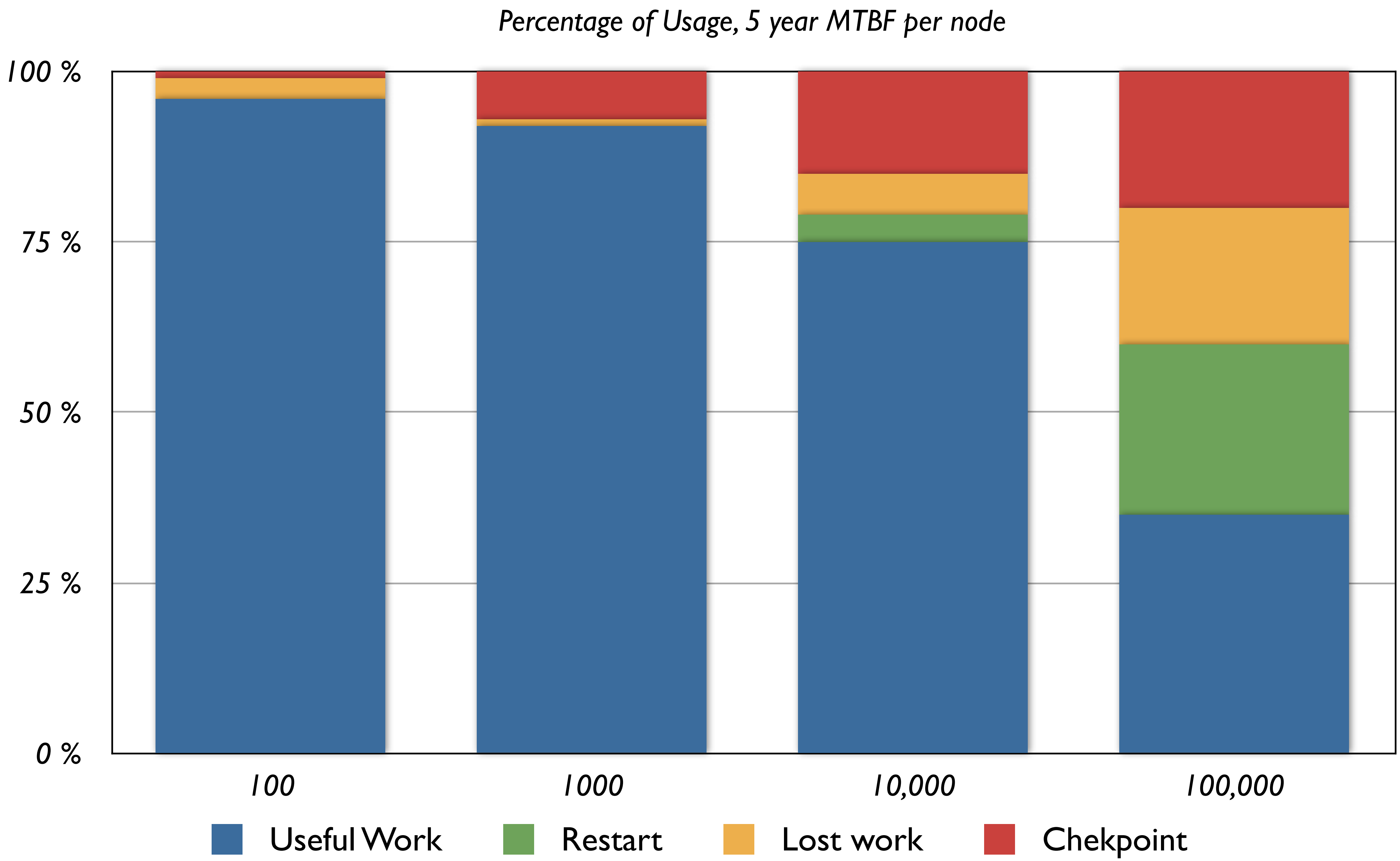
The output:

The optimal τ that minimizes the total useful work ?

Optimal solution

Young 74, Daly 2006, $\cdots$

What's wrong with Checkpoint/Restart



1. Failure Modeling
2. Fault tolerance actions scheduling
3. Simulations
4. Conclusion and future work

Failure Modeling

Estimate or predict

- The time to the next failure.
- The location of the next failure.
- What kind of failure: permanent, transient, hardware or software...

Failure Modeling

Estimate or predict

- The time to the next failure.
- The location of the next failure.
- What kind of failure: permanent, transient, hardware or software...

Probability distribution estimation

Estimate **offline** the probability distribution $F(t)$ of the time to the next failure from the previous occurrence of failures.

Failure Modeling

Estimate or predict

- The time to the next failure.
- The location of the next failure.
- What kind of failure: permanent, transient, hardware or software...

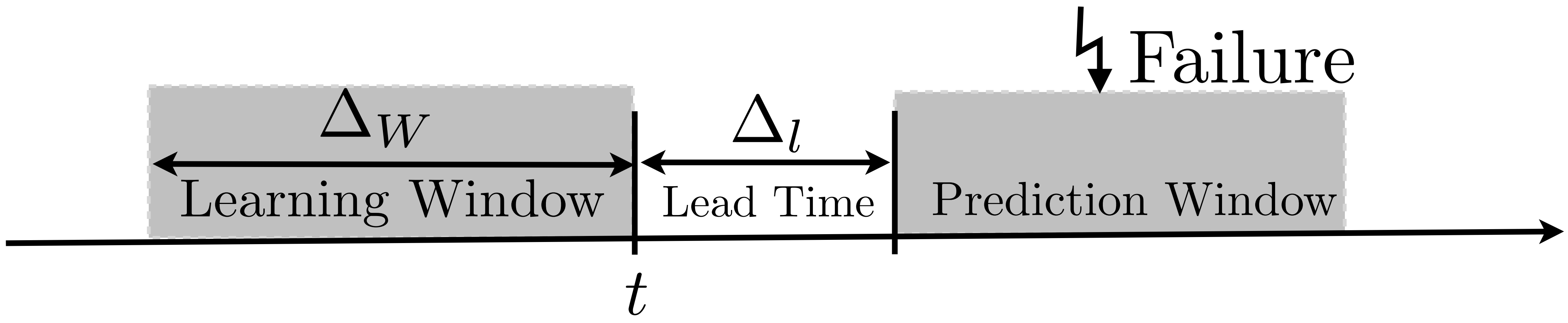
Probability distribution estimation

Estimate **offline** the probability distribution $F(t)$ of the time to the next failure from the previous occurrence of failures.

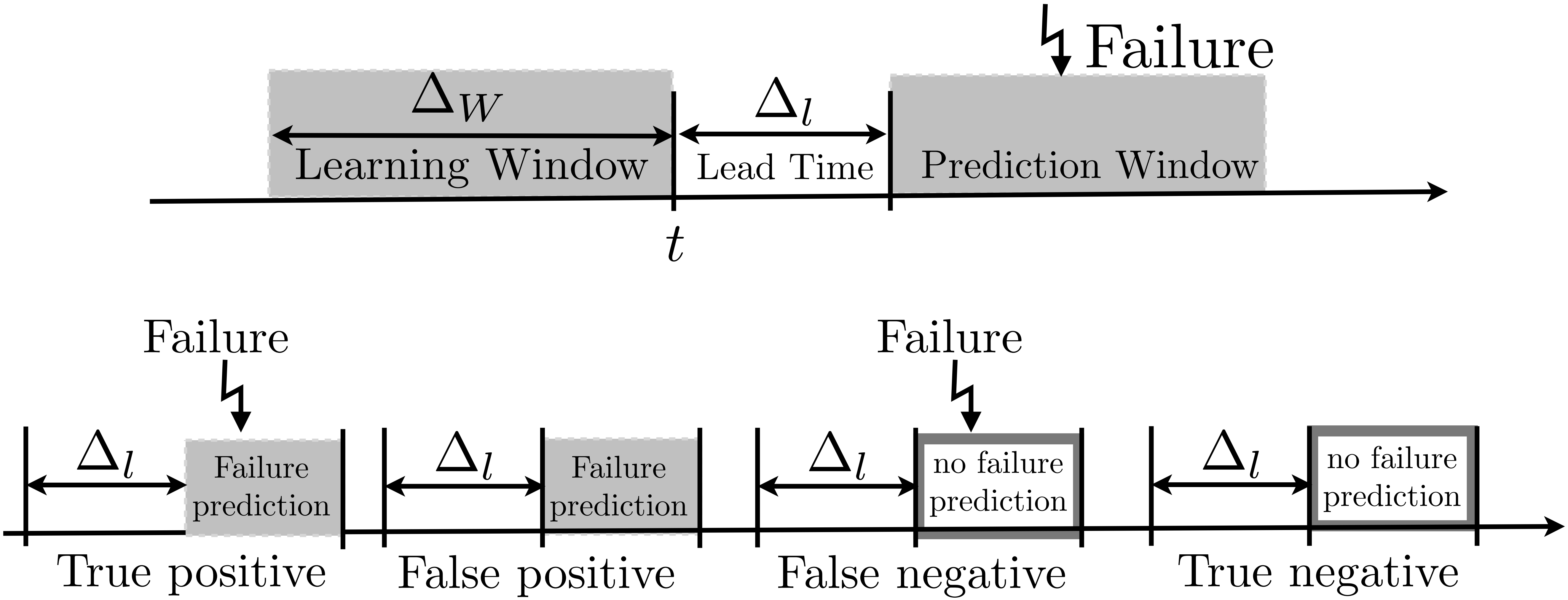
Online failure prediction

Predict during runtime whether a failure will occur in the near future based on an assessment of the monitored current system state.

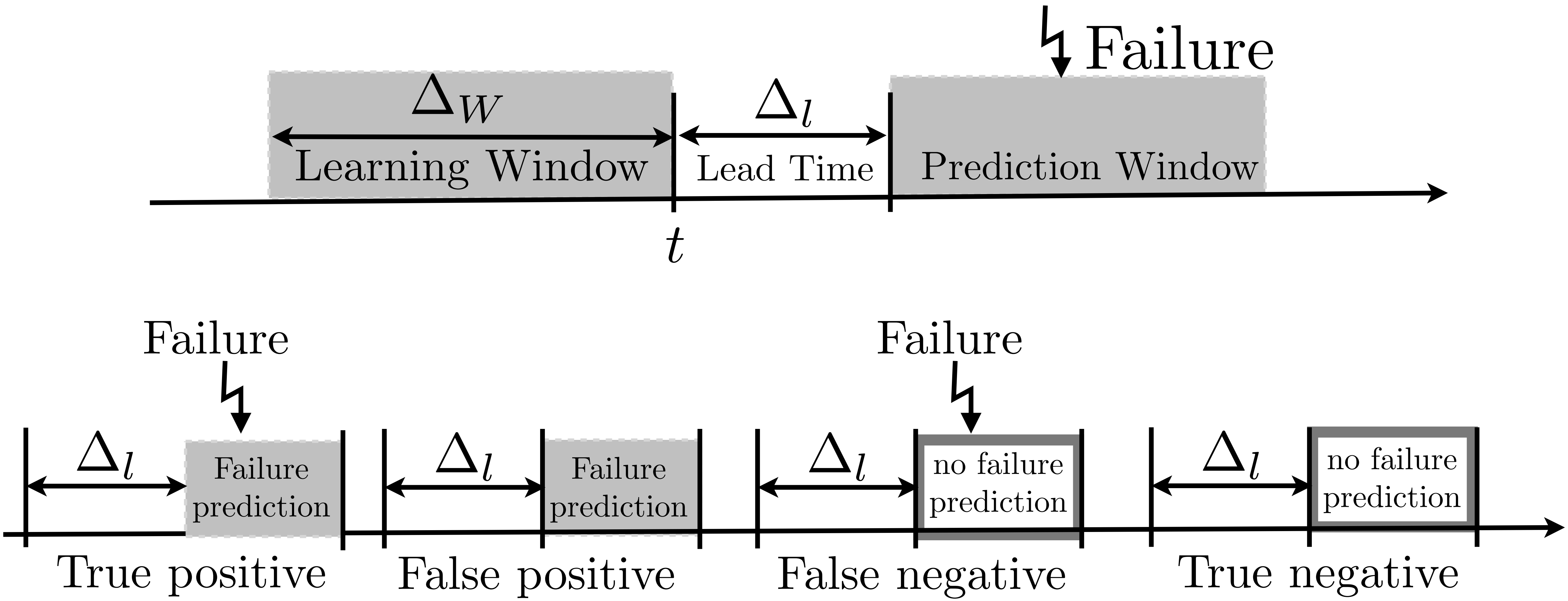
Online Failure prediction



Online Failure prediction



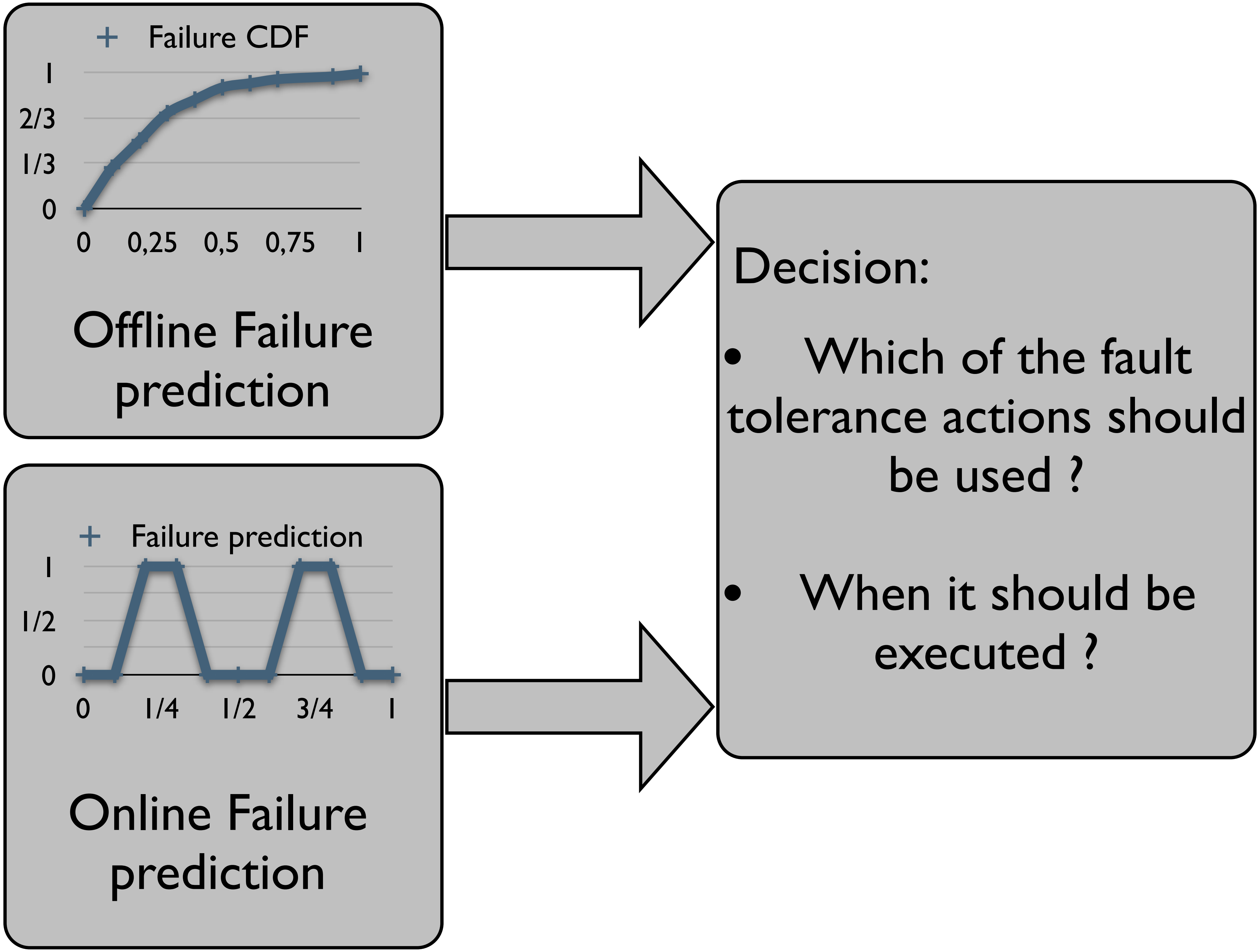
Online Failure prediction



The recall: $r = \frac{\#True\ positive}{\#True\ positive + \#False\ negative}$

The precision: $p = \frac{\#True\ positive}{\#True\ positive + \#False\ positive}$

Online Failure prediction



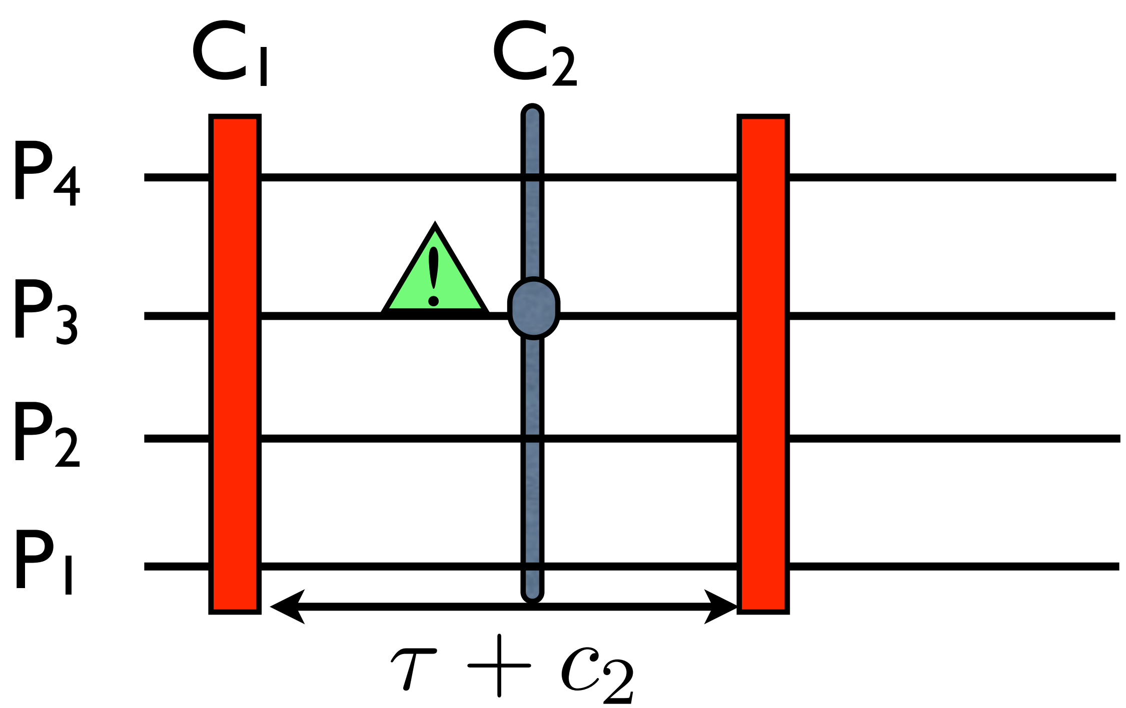
1. Failure Modeling
2. Fault tolerance actions scheduling
3. Simulations
4. Conclusion and future work

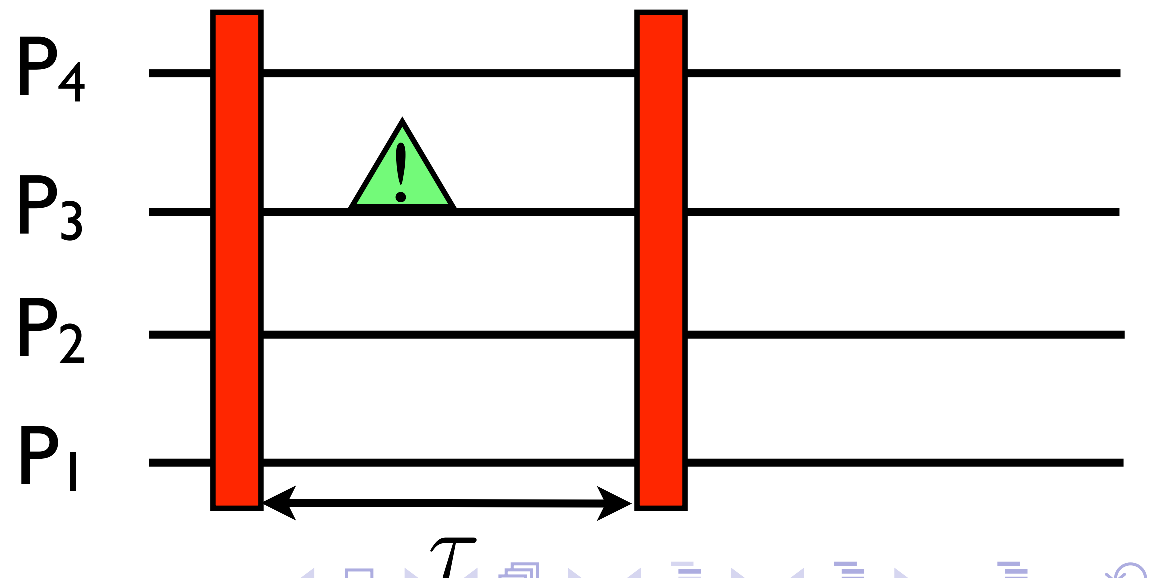
FTI: high performance Fault Tolerance Interface

- Fast proactive checkpoint (save a process context in 2-3 second)
- Global preventive checkpoint (save the entire application state in a remote storage 10 min for current petaflops systems)

The proposed combination

- Perform or not fast proactive checkpoint of one process once a we have a failure a prediction
- Periodically perform a preventive checkpoint (as the recall < 100 %).





Mathematical Modeling

Proactive decision

- To checkpoint:

$$W_p = p\left(R + c_2 + \Delta_I - c_2\right) + \overline{p}c_2$$

- To ignore:

$$W_{np} = p\left(R + t_a + \Delta_I\right)$$

Mathematical Modeling

Proactive decision

- To checkpoint:

$$W_p = p\left(R + c_2 + \Delta_l - c_2\right) + \overline{p}c_2$$

- To ignore:

$$W_{np} = p\left(R + t_a + \Delta_l\right)$$

The proactive action is performed iif

$$W_p \leq W_{np} \equiv \overline{p}c_2/p \leq t_a$$

Mathematical Modeling

Preventive period

- Assuming that failures are exponentially distributed with a mean μ.
- $t\bar{r}/\mu$ failures that we can not predict.
- $t \times r \times s/\mu$ failures predicted with a short lead time (s$=\mathbb{P}\left\{\Delta_l < c_2\right\}$).
- $t \times r \times q \times p/\mu$ Ignored true positive alerts (q is the probability that the decision is to ignore the alert).
- The preventive checkpoint cost c_1.

Mathematical Modeling

Preventive period

- Assuming that failures are exponentially distributed with a mean μ.
- $t\bar{r}/\mu$ failures that we can not predict.
- $t \times r \times s/\mu$ failures predicted with a short lead time (s$=\mathbb{P}\left\{\Delta_l < c_2\right\}$).
- $t \times r \times q \times p/\mu$ Ignored true positive alerts (q is the probability that the decision is to ignore the alert).
- The preventive checkpoint cost c_1.

The optimal interval between preventive checkpoints:

$$\tau^* = \begin{cases} \sqrt{\frac{2\mu c_1 - srh^2}{1-sr}} & \text{if } h < \sqrt{2\mu c_1} \\ \sqrt{2\mu c_1} & \text{if } h \geq \sqrt{2\mu c_1} \end{cases} \quad \text{where } h = \frac{c_2\overline{p}}{p}$$

1. Failure Modeling
2. Fault tolerance actions scheduling
3. Simulations
4. Conclusion and future work

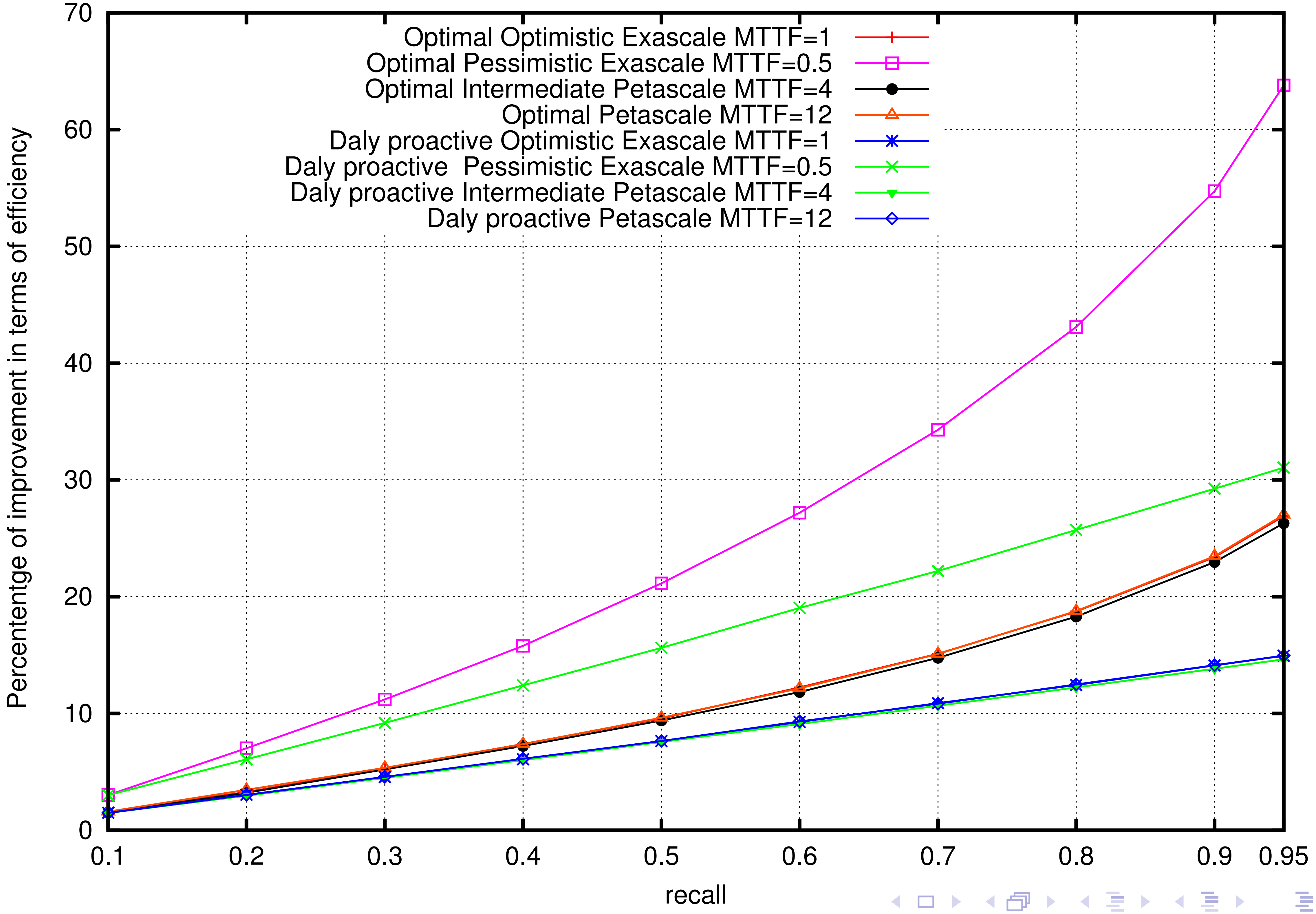
The considered configuration

Table: Computing platform configuration

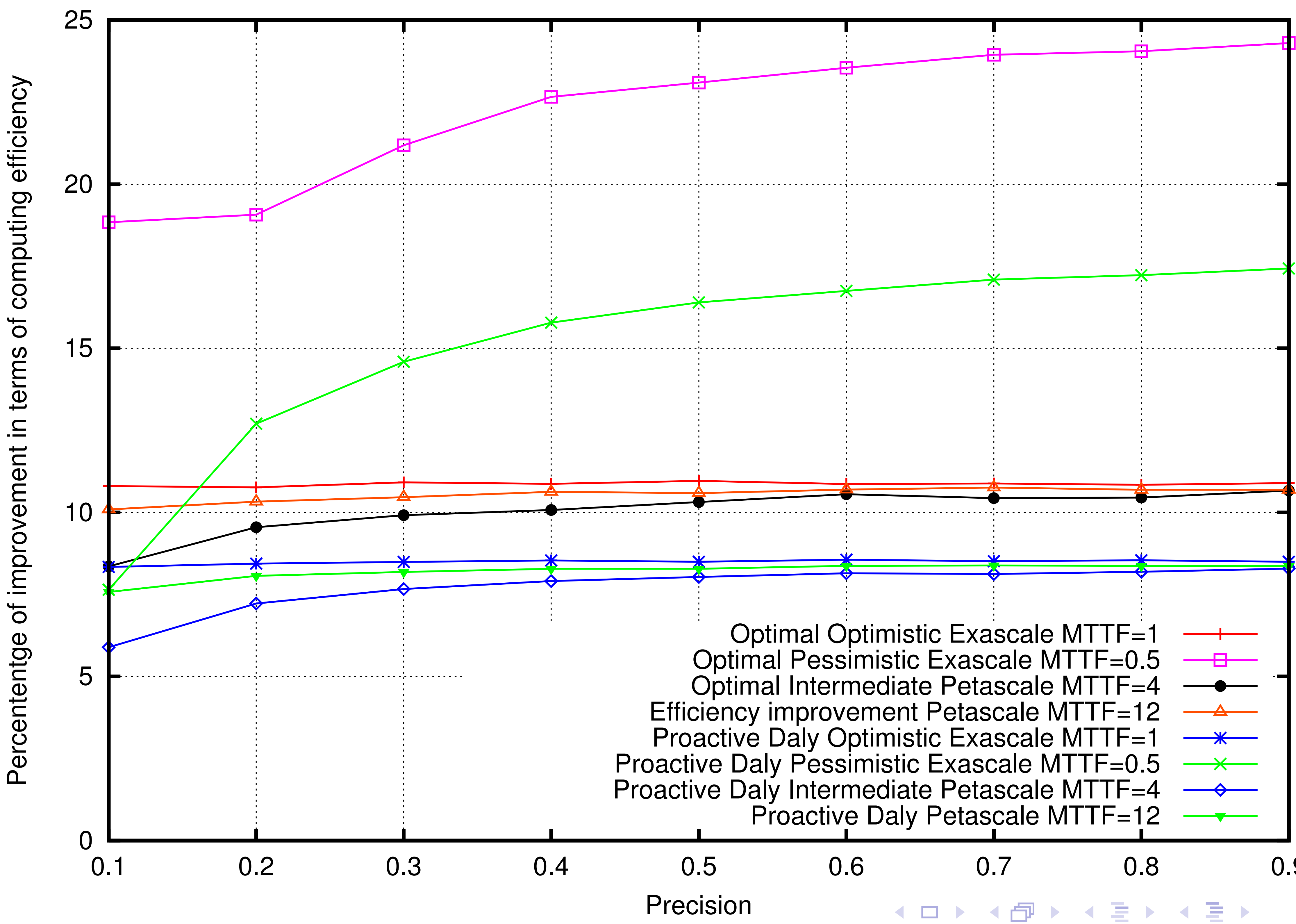
Paramters	Petascale Jaguar, 10PF	Intermediate 100PF	Exascale Optimistic	Exascale Pessimistic
MTTF	24h to 6h	6h to 4h	2h to 1h	30 min
Preventive Checkpoint time	30 min	10 min	2.5 min	10 min
Proactive Checkpoint time	10 to 5 sec	5 to 1 sec	5 to 1 sec	5 to 1 sec

- Petascale: the checkpoint size per node is between 100GBs and 200GBs and the writing speed is about 350MB/s.
- Exascale (64 petabytes of memory with 100k nodes): checkpoint size per node between 200GBs and 500GBs with a writing bandwidth of 3GB/s and 1GB/s for the pessimistic scenario (Non volatile RAM, Phase Change Memories and 3-D circuit)

Impact of the recall

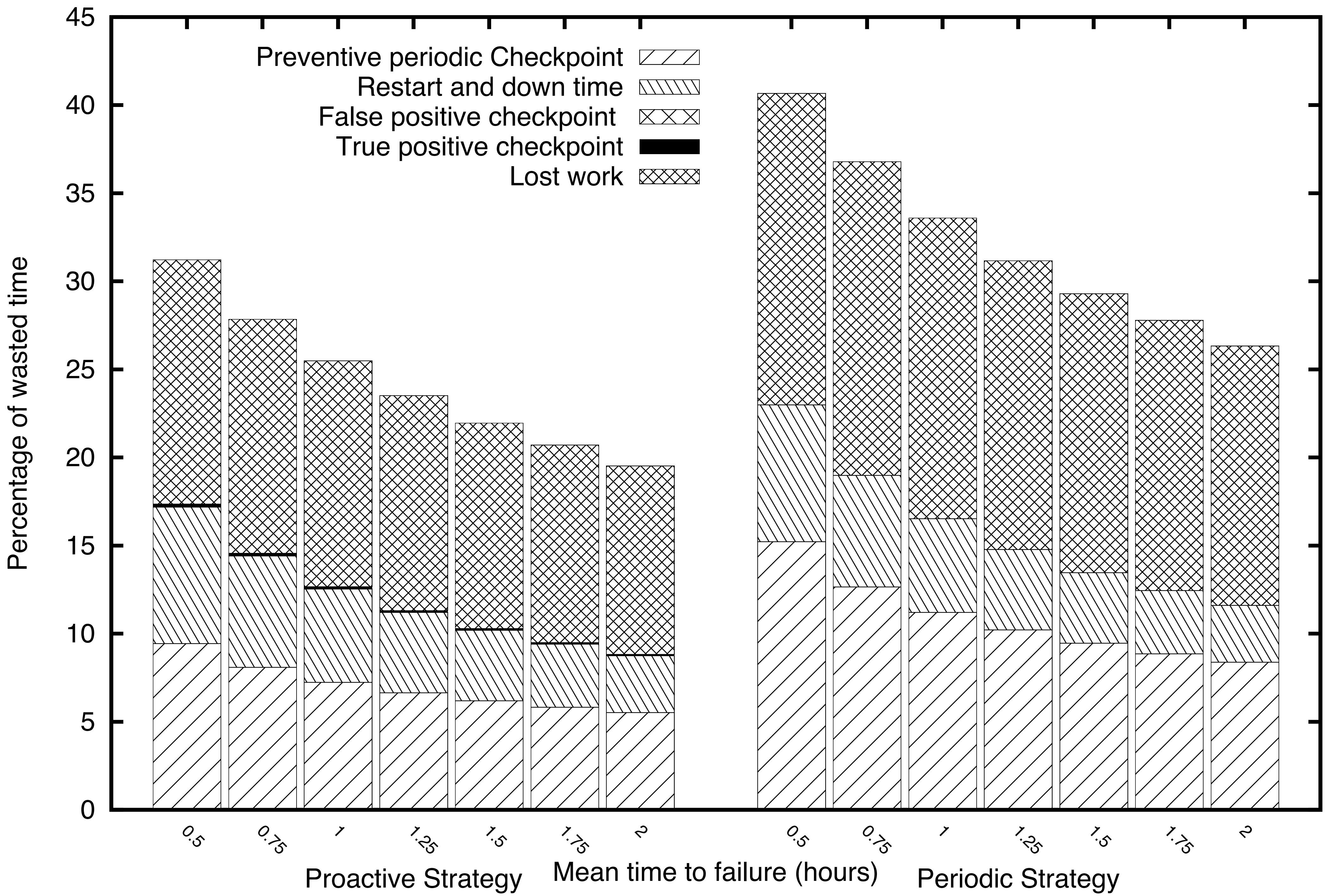


Impact of the recall



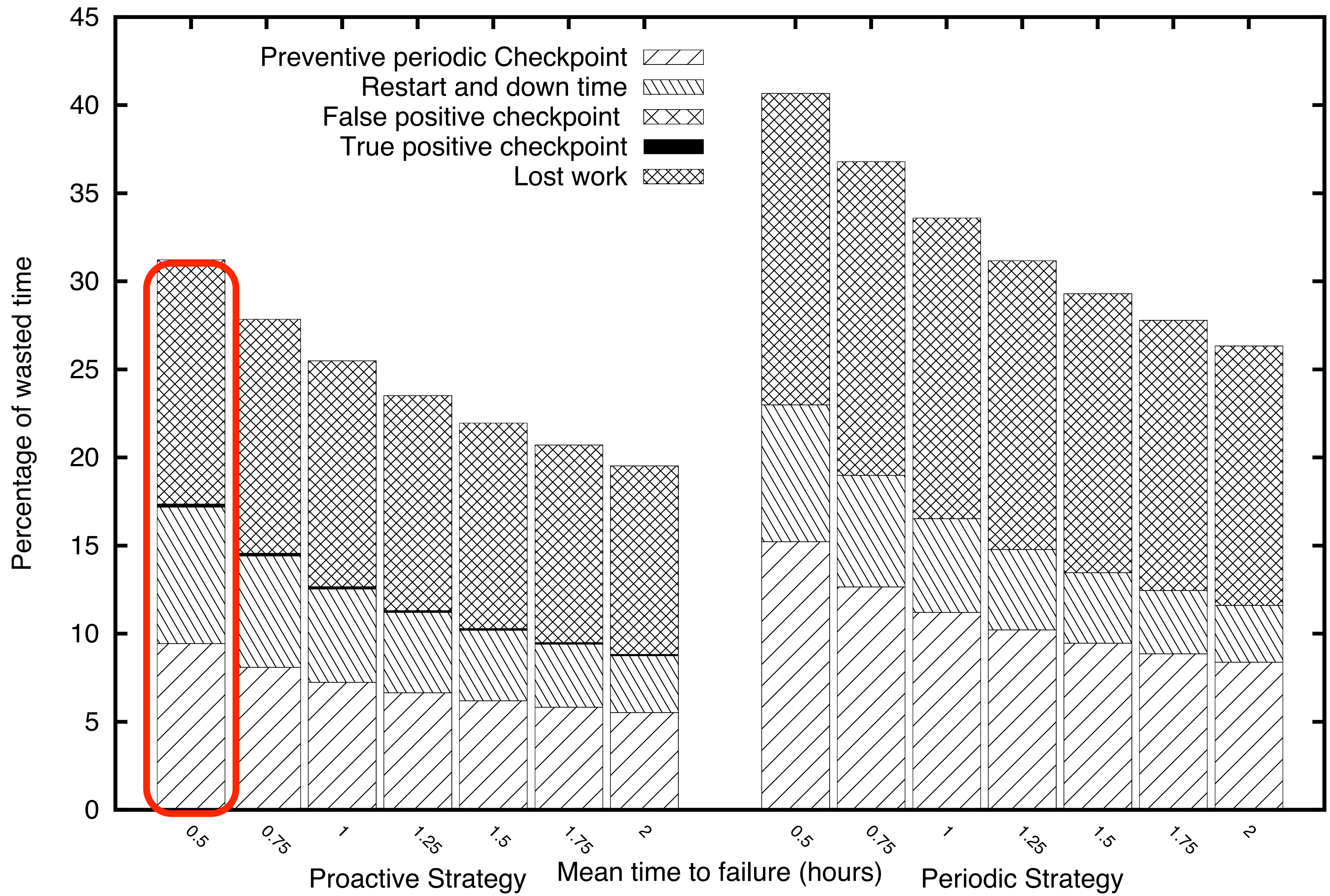
Impact of the checkpoint cost and the failure rate

Recall of 50% and a prediction precision 80%.



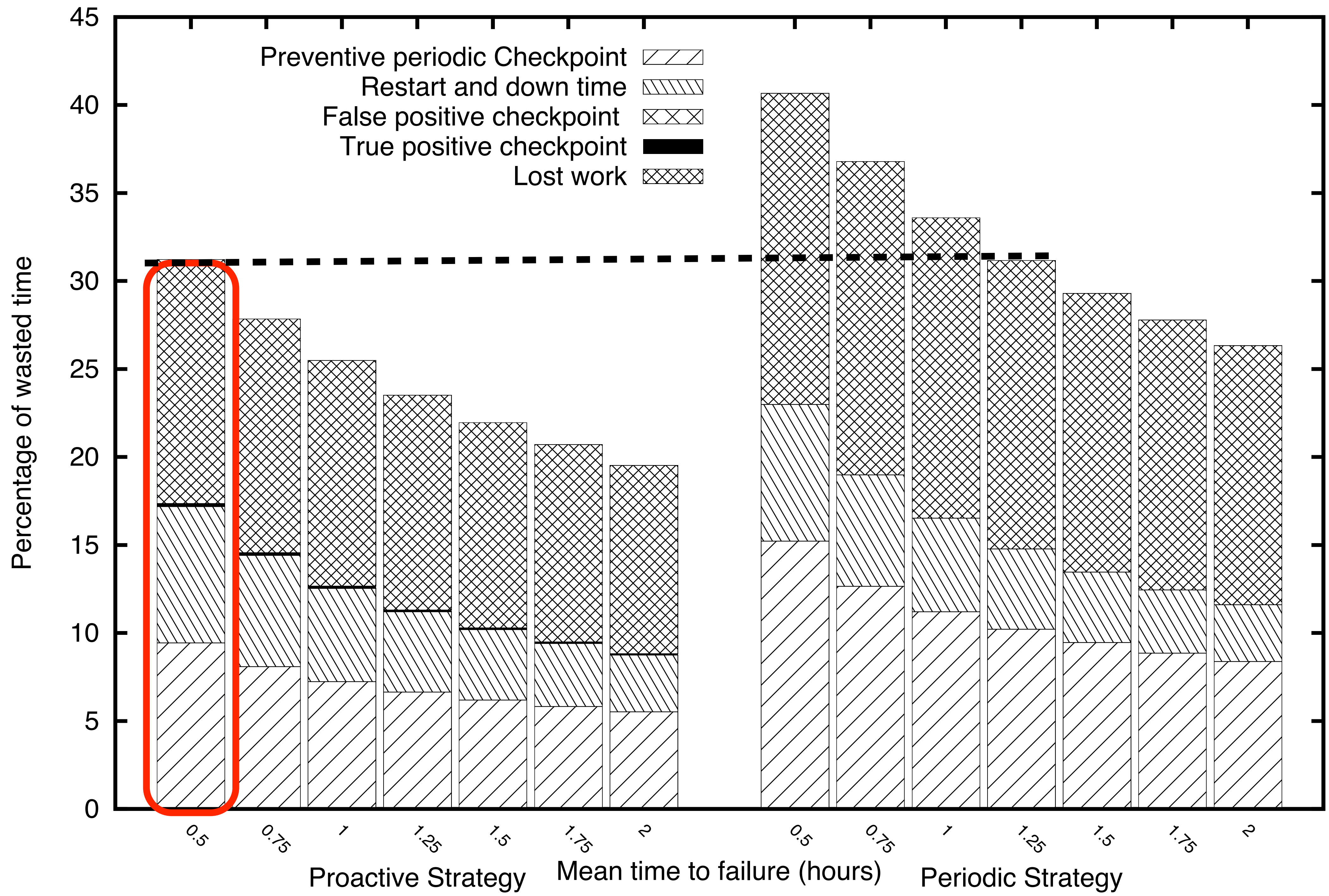
Impact of the checkpoint cost and the failure rate

Recall of 50% and a prediction precision 80%.



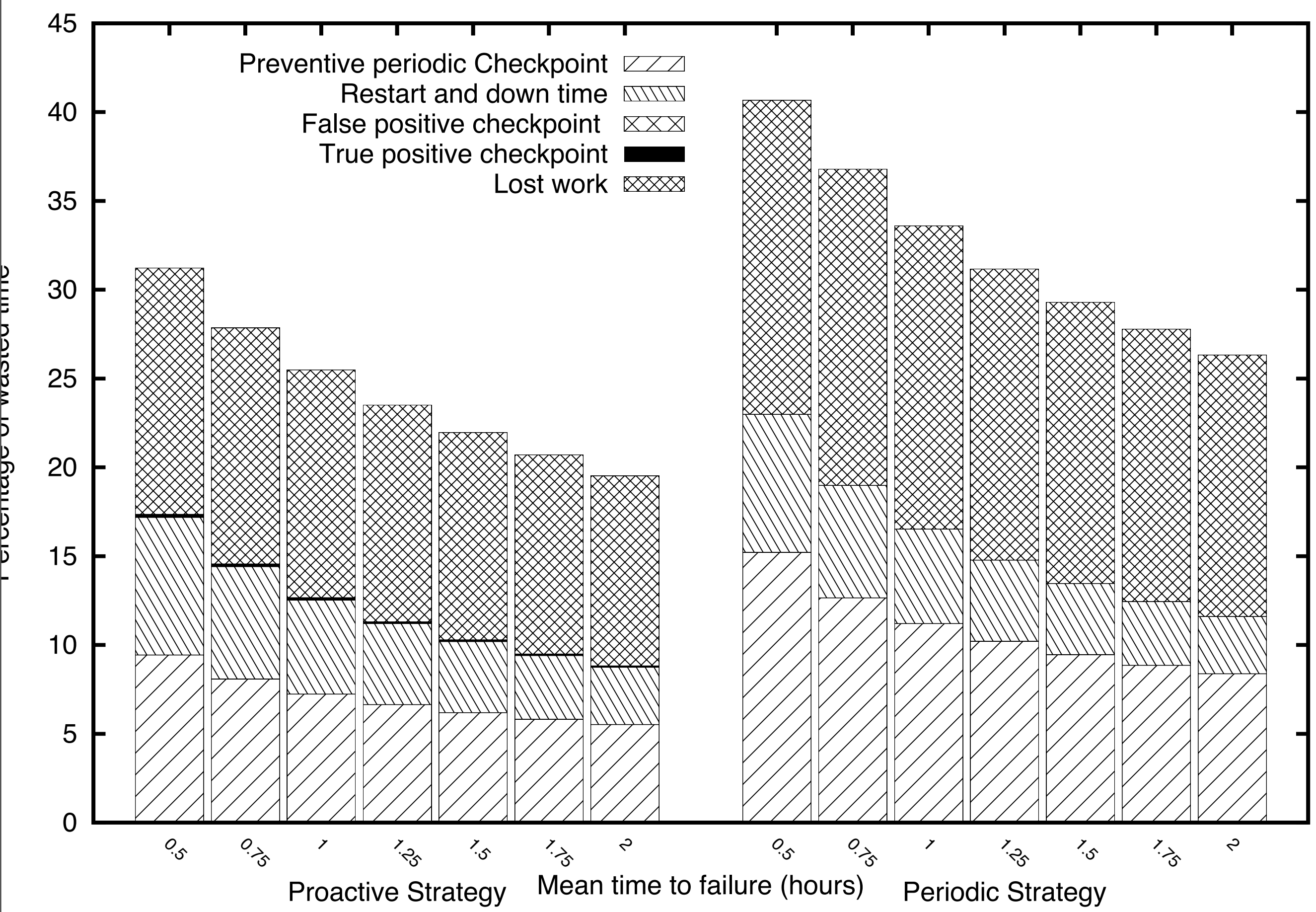
Impact of the checkpoint cost and the failure rate

Recall of 50% and a prediction precision 80%.

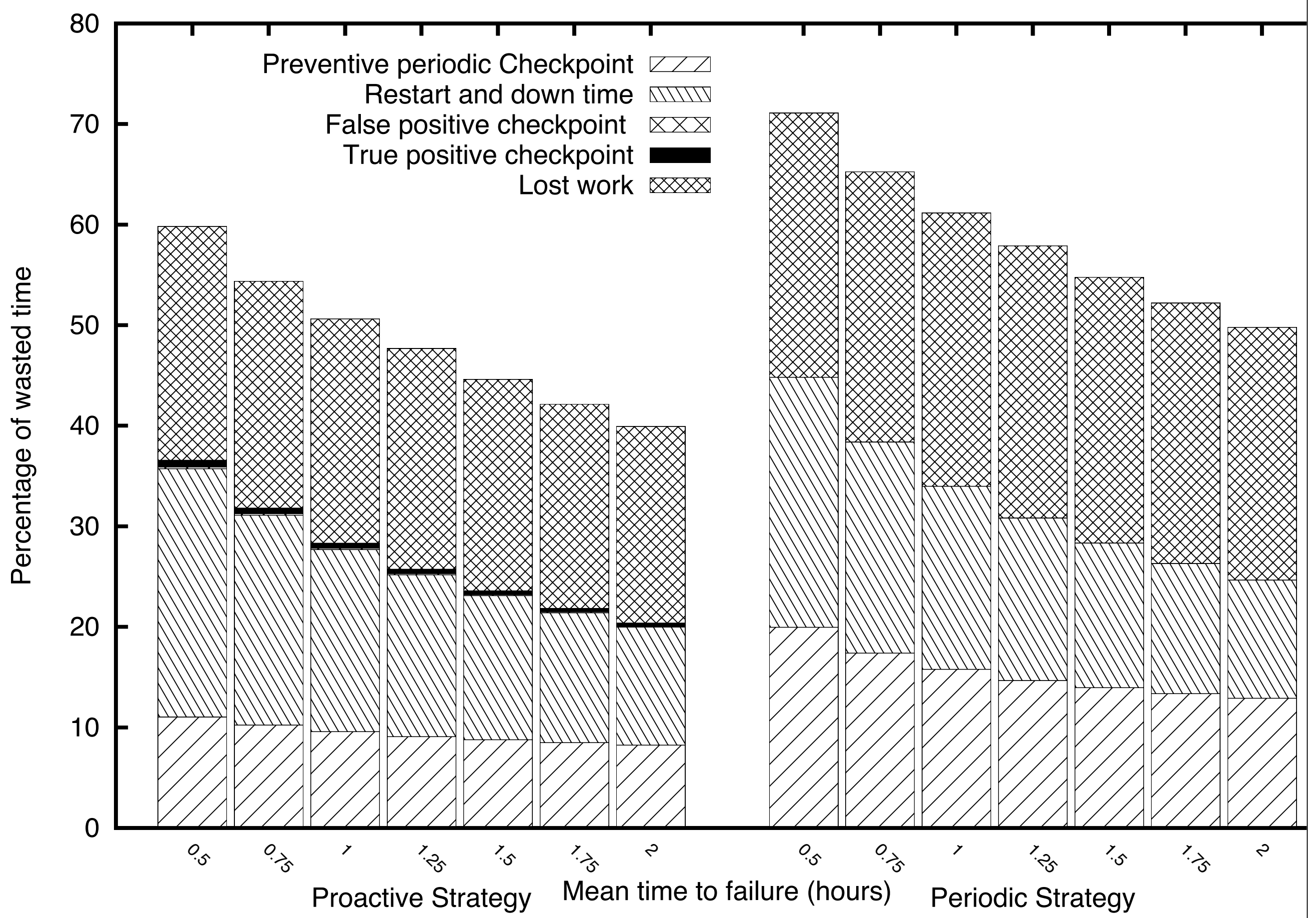


Impact of the checkpoint cost and the failure rate

Recall of 50% and a prediction precision 80%.



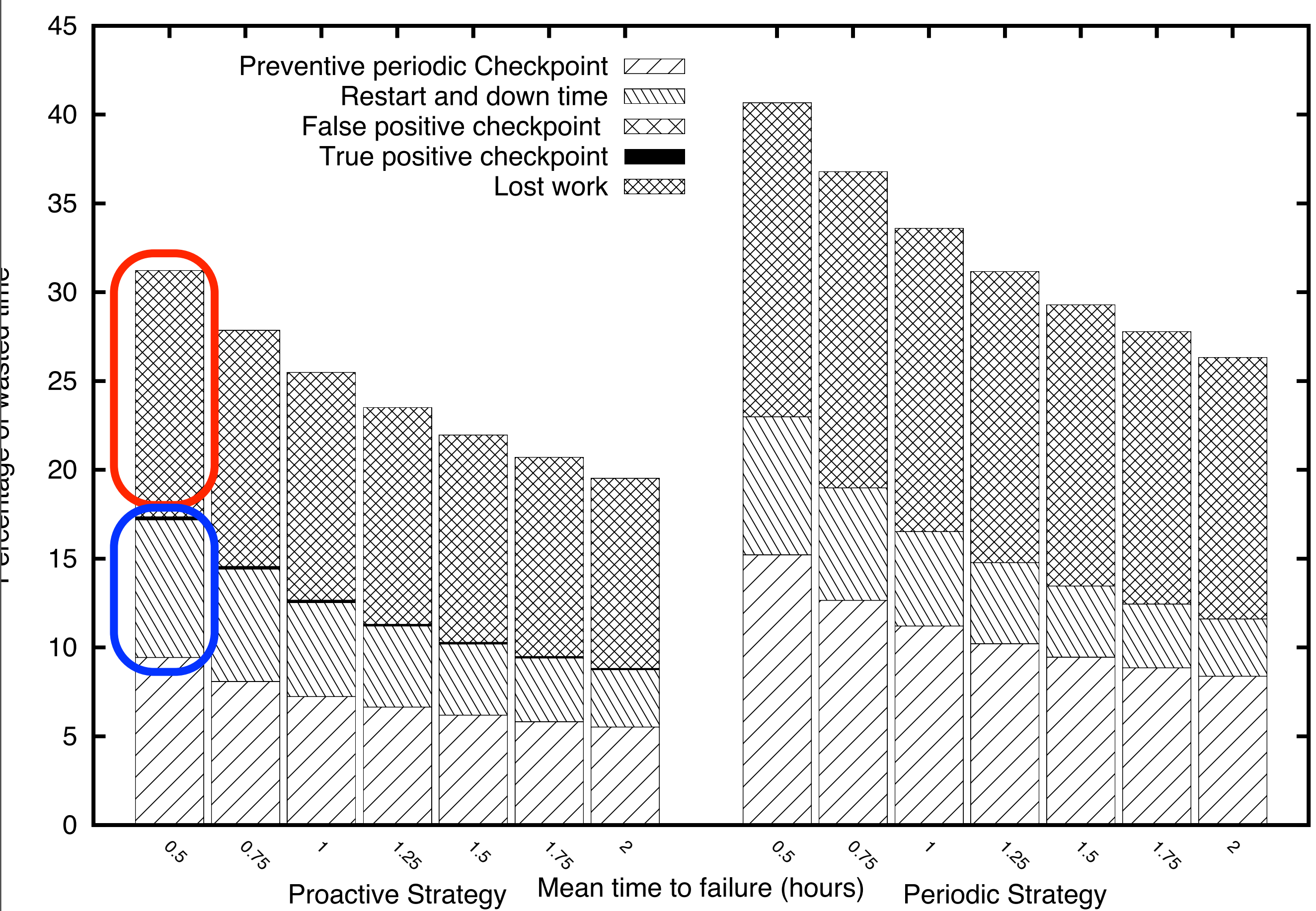
(a) Optimistic exascale configuration



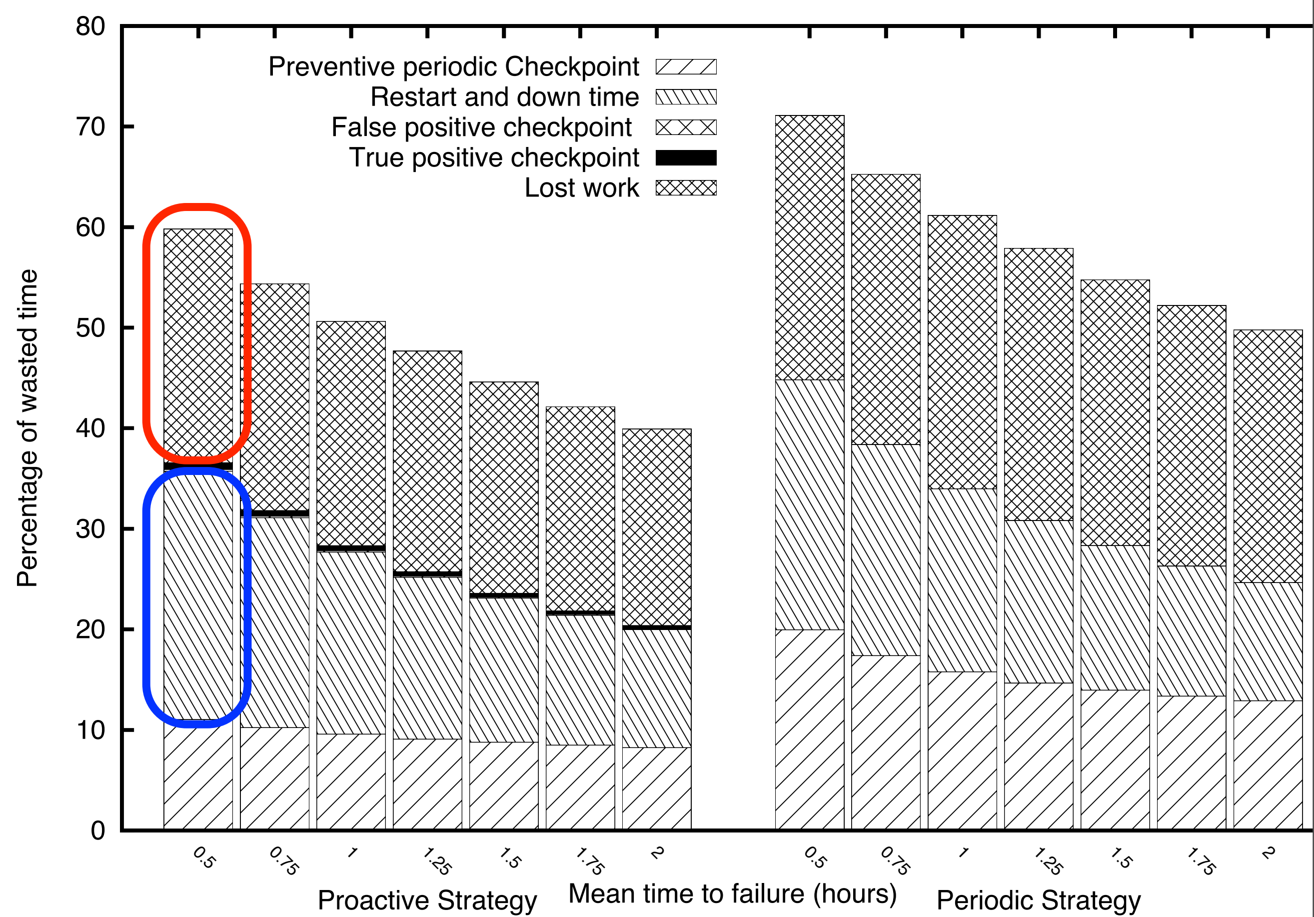
(b) Pessimistic exascale configuration

Impact of the checkpoint cost and the failure rate

Recall of 50% and a prediction precision 80%.



(a) Optimistic exascale configuration



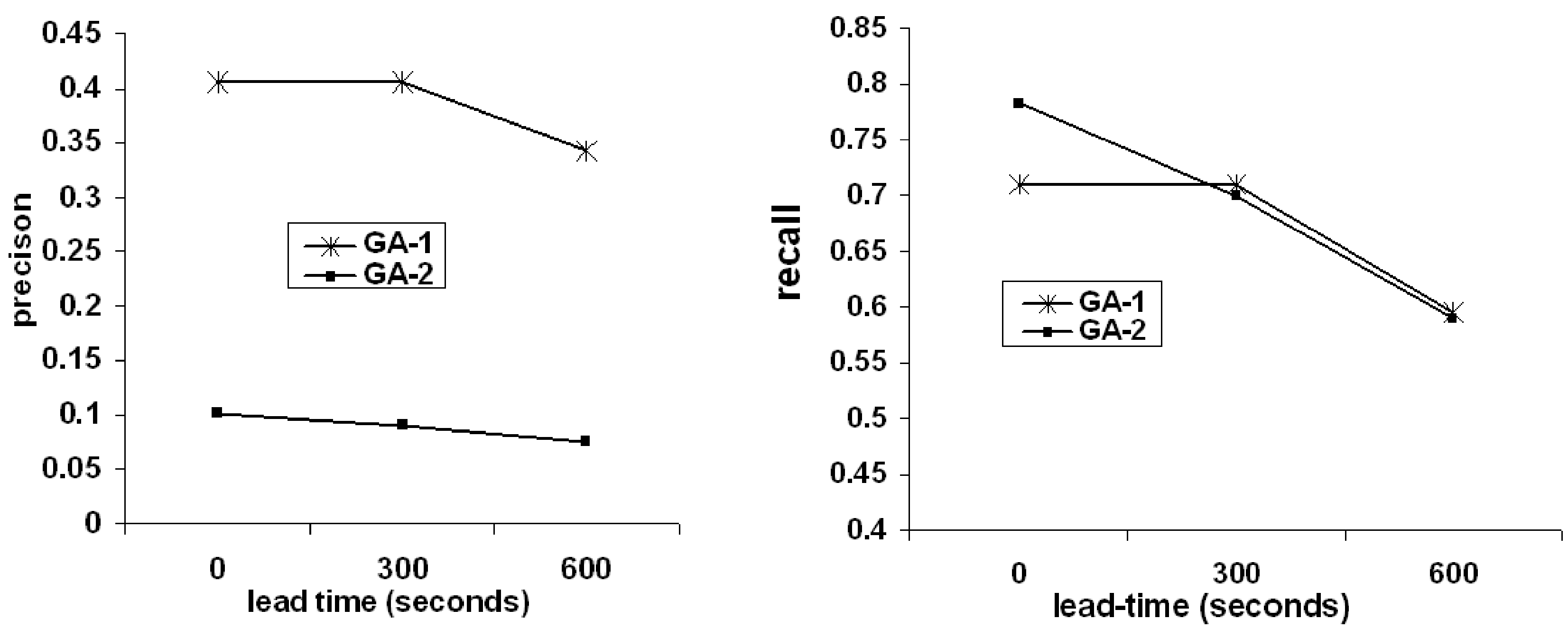
(b) Pessimistic exascale configuration

Conclusion

- Combining accurate failure prediction, fast proactive checkpointing and preventive multilevel checkpointing to mitigate the effects of failures and improve execution performance
- We developed a mathematical model that reflects the expected computing efficiency of our proposed technique.
- The prediction recall has an important impact on the overall efficiency improvement in contrary to the prediction precision, that has only a minor impact. (if failure predictors provide some flexible precision/recall trade-offs, one should favor first high recall as opposed to high precision.)
- With a 50% recall the performance achieved is equivalent to the performance of a system with an MTTBF two times higher.

Future work

- Manage the tradeoff between the lead-time and recall.
- Manage the tradeoff between the precision and the recall.
- Use different sources of failure prediction that concerns different component of the machine.



(a) (b)

Figure 2: (a) Precision (b) Re

A Practical Failure Prediction with Location and Lead Time for Blue Gene/P

Ziming Zheng , Zhiling Lan, Rinku Gupta, Susan Coghlan, Peter Beckman

Future work

- Investigate the failure distribution of the False positives prediction and its impact on the model.
- Extend the proposed protocol and the model to use different proactive actions like the replication and the migration.
- Provide more accurate model for the checkpoint cost.