

Comment trouver un sujet de recherche?

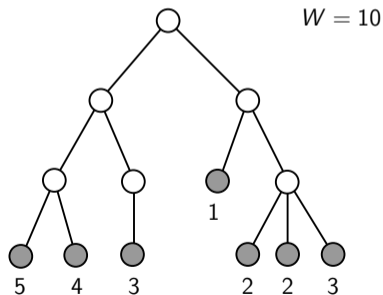


Anne Benoit
Le Pleynet, France

Semaine sport-études des L3 – 18 janvier 2023

Un premier problème: Placement de répliques dans des arbres

- On nous donne un arbre; les feuilles de l'arbre sont des clients
- Un certain nombre de requêtes par unité de temps par client
- But: placer des serveurs pour satisfaire toutes les requêtes
- **Le problème est-il bien posé ?**



Un premier problème: Placement de répliques dans des arbres

- On nous donne un arbre; les feuilles de l'arbre sont des clients
- Un certain nombre de requêtes par unité de temps par client
- But: placer des serveurs pour satisfaire toutes les requêtes
- Le problème est-il bien posé ?

Il faut clarifier certains points

- Combien de requêtes sont traitées par un serveur ?
- On commence par un cas simple, W requêtes max par serveur
- Quelle est la fonction objective ?
- Minimiser le nombre de serveurs à placer
- Mais encore ?
- Quelle est la politique de service des clients par les serveurs ?

Un premier problème: Placement de répliques dans des arbres

- On nous donne un arbre; les feuilles de l'arbre sont des clients
- Un certain nombre de requêtes par unité de temps par client
- But: placer des serveurs pour satisfaire toutes les requêtes
- **Le problème est-il bien posé ?**

Politiques au plus près, plus haut dans l'arbre, plusieurs serveurs par clients, on peut rajouter de l'énergie, de la résilience, ça fait plein de problèmes rigolos !

Premier exemple (rapide) illustrant la recherche de sujet de recherche: broder sur les règles du jeu, établir la complexité des problèmes... 😊

Un sujet plus détaillé: Replication Is More Efficient Than You Think

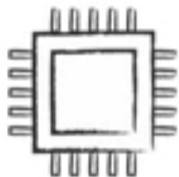


Anne Benoit¹, Thomas Herault², Valentin Le Fèvre¹, Yves Robert^{1,2}

1. LIP, Ecole Normale Supérieure de Lyon, France
2. ICL, University of Tennessee Knoxville, USA

Semaine sport-études des L3 – 18 janvier 2023

Scale Is The Enemy

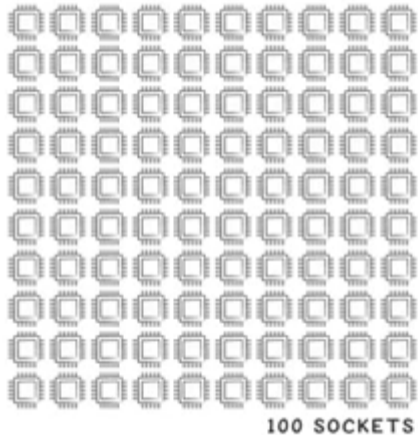


100
YEARS

MEAN TIME
BETWEEN FAILURES

Chouette, mon ordi est fiable !

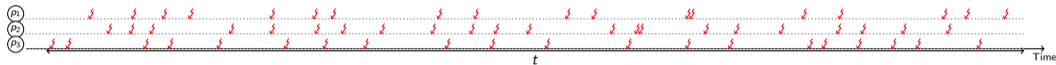
Scale Is The Enemy



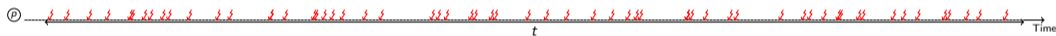
1
YEAR
MEAN TIME
BETWEEN FAILURES

Pas tant que ça ?

Scale Is The Enemy



If three processors have around 20 faults during a time t ($\mu = \frac{t}{20}$)...



...during the same time, the platform has around 60 faults ($\mu_N = \frac{t}{60}$)

$$\mu_N = \frac{\mu}{N}$$

Scale Is The Enemy



36
DAYS

MEAN TIME
BETWEEN FAILURES

Scale Is The Enemy



8
HOURS
—
MEAN TIME
BETWEEN FAILURES

Scale Is The Enemy

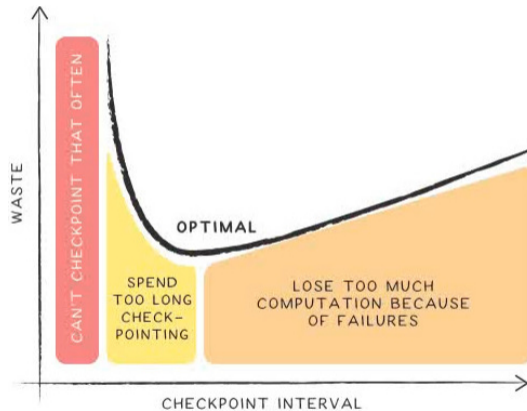
Need to checkpoint!

... i.e., periodically save the application so you don't have to restart from scratch when a failure occurs

But when should we checkpoint?

Scheduling matters 😊

Optimal Checkpointing Interval



Par exemple, Bob qui rédige sa thèse...

The Young/Daly Formula

Period T , minimize overhead $\mathbb{H}(T) = \frac{\mathbb{E}(T+C)}{T} - 1$

Theorem

$$T_{opt} = \sqrt{\frac{2C}{\lambda_N}} = \sqrt{2C\mu_N} = \Theta(\lambda^{-\frac{1}{2}}) \quad (1)$$

$$\mathbb{H}_{opt} = \sqrt{2C\lambda_N} + o(\lambda^{\frac{1}{2}}) = \Theta(\lambda^{\frac{1}{2}}) \quad (2)$$

Recall that $\lambda_N = N\lambda = \frac{1}{\mu_N} = \frac{N}{\mu}$

... mais où est le problème alors ? Sur quoi peut-on chercher ?

Outline

- 1 Replication
- 2 New Strategy
- 3 Digression
- 4 Experiments

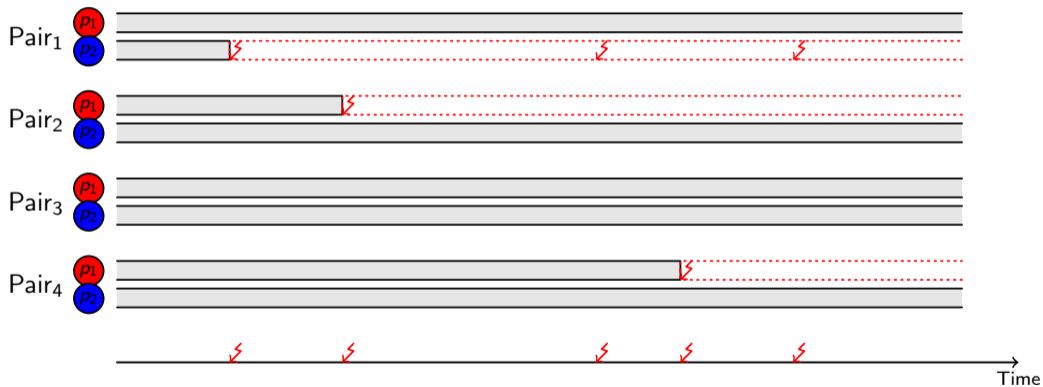
Replication

- Full replication: efficiency $< 50\%$
- Can replication+checkpointing be more efficient than checkpointing alone?
- Study by Ferreira et al. [SC'2011]: **yes**
- Revisited by Hussain, Znati and Melhem [SC'2018]: **yes**

Model by Ferreira et al. [SC'2011]

- Platform with $N = 2b$ processors arranged into b pairs
- Parallel application with b processes, each replicated
- When a replica is hit by a failure, it is not restarted
- Application fails when both replicas in one pair have been hit

Example



Why Replication?

With $\mu = 5$ years, time to reach 90% chance of fatal failure:

No replication 24 minutes for $N = 100,000$

No replication 12 minutes for $N = 200,000$

Replication 85 hours for $N = 200,000$ ($b = 100,000$ pairs)

Checkpointing Period

- Replication combined with periodic checkpoint-restart à la Young/Daly
- Restart after **interruption** instead of after **first failure**
- Many failures needed to interrupt the application
⇒ checkpointing period much larger than without replication
- **Optimal period?**

Mean Time To Interruption

- $N = 2b$, b processor pairs
- $n_{\text{fail}}(2b)$ expected number of failures to interrupt the applications
- MTTI $M_N = M_{2b} = \text{Mean Time to Interruption}$
⇒ replaces MTBF from the application perspective

$$M_N = M_{2b} = n_{\text{fail}}(2b) \times \mu_{2b} = n_{\text{fail}}(2b) \times \frac{\mu}{2b} = \frac{n_{\text{fail}}(2b)}{2\lambda b} \quad (3)$$

Mean Time To Interruption

- $N = 2b$, b processor pairs
- $n_{\text{fail}}(2b)$ expected number of failures to interrupt the applications
- MTTI $M_N = M_{2b} = \text{Mean Time to Interruption}$
⇒ replaces MTBF from the application perspective

$$M_N = M_{2b} = n_{\text{fail}}(2b) \times \mu_{2b} = n_{\text{fail}}(2b) \times \frac{\mu}{2b} = \frac{n_{\text{fail}}(2b)}{2\lambda b} \quad (3)$$

Theorem

$$n_{\text{fail}}(2b) = 1 + 4^b / \binom{2b}{b} \approx \sqrt{\pi b}$$

A Little Bragging

$$M_{2b} = \int_0^{\infty} (1 - (1 - e^{-\lambda t})^2)^b dt$$

$$x = \frac{e^{-\lambda t}}{2} \Rightarrow n_{\text{fail}}(2b) = 2b4^b \int_0^{\frac{1}{2}} x^{b-1} (1-x)^b dx = 2b4^b B\left(\frac{1}{2}, b, b+1\right)$$

$$B(z, u, v) = \int_0^z x^{u-1} (1-x)^{v-1} dx = \frac{z^u}{u} \times {}_2F_1\left[\begin{matrix} u, 1-v \\ u+1 \end{matrix}; z\right]$$

$${}_2F_1\left[\begin{matrix} u, v \\ w \end{matrix}; z\right] = \sum_{n=0}^{\infty} \frac{\langle u \rangle_n \langle v \rangle_n}{\langle w \rangle_n} \frac{z^n}{n!} = 1 + \frac{uv}{1!w} z + \frac{u(u+1)v(v+1)}{2!w} z^2 + \dots$$

$$B\left(\frac{1}{2}, b, b+1\right) = \frac{1}{b2^b} \times {}_2F_1\left[\begin{matrix} b, -b \\ b+1 \end{matrix}; \frac{1}{2}\right]$$

$${}_2F_1\left[\begin{matrix} b, -b \\ b+1 \end{matrix}; \frac{1}{2}\right] = \frac{\sqrt{\pi} \Gamma(b+1)}{2^{b+1}} \left[\frac{1}{\Gamma(b+1)\Gamma(\frac{1}{2})} + \frac{1}{\Gamma(b+\frac{1}{2})\Gamma(1)} \right]$$

$$\Gamma(z) = \int_0^{\infty} x^{z-1} e^{-x} dx, \Gamma(1) = 1, \Gamma(b+1) = b!, \Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}, \Gamma\left(b+\frac{1}{2}\right) = \frac{\sqrt{\pi}(2b)!}{4^b b!}$$

Maple helped find the formula

$$M_{2b} = \int_0^{\infty} (1 - (1 - e^{-\lambda t})^2)^b dt$$

$$M_{2b} = \int_0^{\infty} e^{-\lambda bt} (2 - e^{2\lambda t})^b dt$$

$$M_{2b} = \sum_{i=1}^b 2^i \binom{b}{i} (-1)^{b-i} \int_0^{\infty} e^{-\lambda(2b-i)t} dt$$

$$M_{2b} = \sum_{i=1}^b 2^i \binom{b}{i} (-1)^{b-i} \frac{1}{\lambda(2b-i)}$$

$$n_{\text{fail}}(2b) = 2\lambda b M_{2b} = \sum_{i=1}^b 2^i \binom{b}{i} (-1)^{b-i} \frac{2b}{2b-i}$$

Luck matters too



Checkpointing

$$\text{No Replication} \quad T_{opt} = \sqrt{2\mu_N C} \quad (4)$$

$$\text{Full Replication} \quad T_{opt} = \sqrt{2M_N C} \quad (5)$$

What's Wrong?

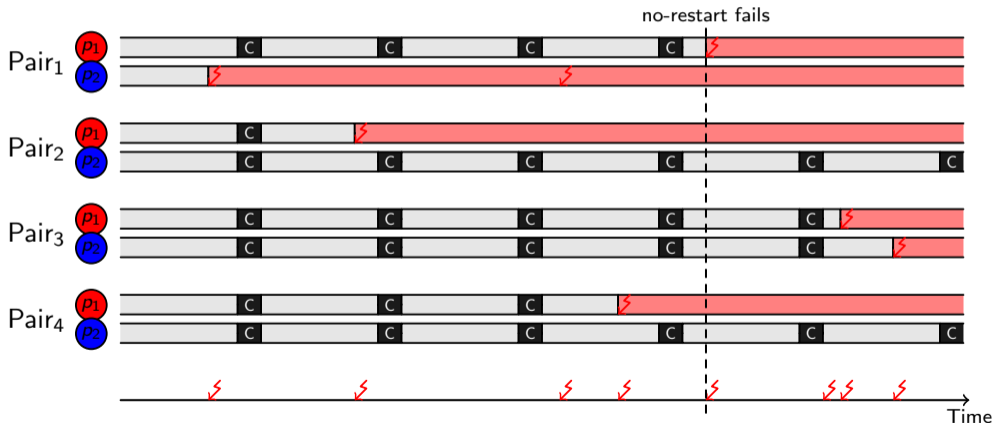
$$T_{opt} = \sqrt{2M_N C}$$

- Just an approximation. How accurate?
- Risk is increasing as more and more processors die until application crash
⇒ Periodic checkpointing (most likely) not optimal 😞

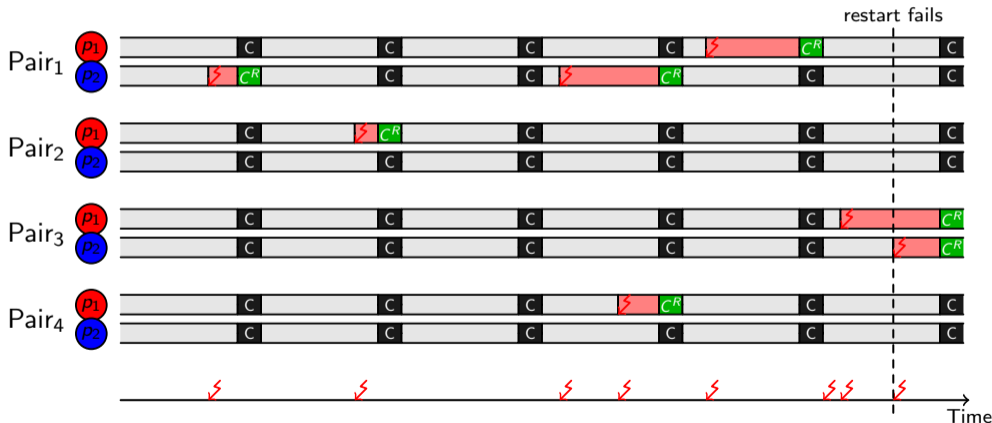
Outline

- 1 Replication
- 2 New Strategy**
- 3 Digression
- 4 Experiments

no-restart vs. restart



no-restart vs. restart



restart

- Restart all failed processors (if any) **after each checkpoint** instead of **only after interruption**
- What is the additional cost?
- What is the optimal checkpointing period?

Combined Checkpoint-Restart

Cost of a checkpoint and restart wave C^R

- one instance of each surviving process saves state (checkpoint)
- processes for missing replicas of the replicas allocated
- new processes load current (checkpointed) state and join system

In-memory checkpoint replication

- the buddy process and the replica are the same process
 - surviving processes upload their checkpoint directly onto memory of newly spawned replicas
- ⇒ no exchange of checkpoints between pair of surviving buddies

Worst case: sequential approach, $C^R = 2C$

Best-case: buddy checkpointing, negligible overhead, $C^R \approx C$

Checkpointing Period

Periodic checkpointing is optimal for *restart*

$$T_{opt}^{rs} = \left(\frac{3C^R}{4b\lambda^2} \right)^{\frac{1}{3}} = \Theta(\lambda^{-\frac{2}{3}}). \quad (6)$$

$$\mathbb{H}^{rs}(T_{opt}^{rs}) = \left(\frac{3C^R \sqrt{b\lambda}}{\sqrt{2}} \right)^{\frac{2}{3}} + o(\lambda^{\frac{2}{3}}) = \Theta(\lambda^{\frac{2}{3}}) \quad (7)$$

An order of magnitude longer!

Outline

- 1 Replication
- 2 New Strategy
- 3 Digression**
- 4 Experiments

Single Processor Pair

One processor: $T_{YD} = \sqrt{2\mu C}$

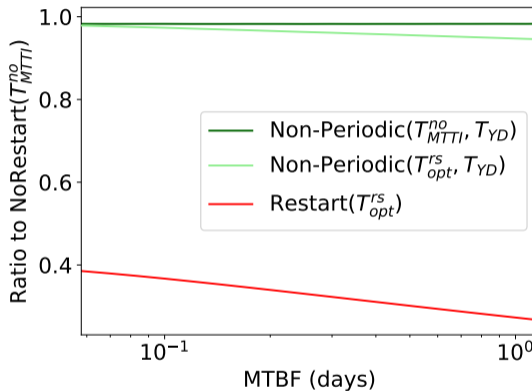
One replica pair, *no-restart*: $\mu_2 = \frac{\mu}{2}$, $n_{\text{fail}}(2) = 3$, $T_{MTTI}^{\text{no}} = \sqrt{3\mu C}$

One replica pair, *restart*: $T_{\text{opt}}^{\text{rs}} = \left(\frac{3}{4}C\mu^2\right)^{\frac{1}{3}}$

Four variants:

- *no-restart* (T_{MTTI}^{no}): baseline
- *restart* ($T_{\text{opt}}^{\text{rs}}$): check how good we are
- *no-restart* NonPeriodic(T_1, T_2):
 - use T_1 while both processors are alive
 - switch to T_2 after first failure
 - Variant 1: $T_1 = T_{MTTI}^{\text{no}}$, $T_2 = T_{YD}$
 - Variant 2: $T_1 = T_{\text{opt}}^{\text{rs}}$, $T_2 = T_{YD}$
- 100,000 simulations, each with 10,000 periods

Single Processor Pair



Ratio of time to solution of two non-periodic strategies and *restart* over time-to-solution of *no-restart* ($C = C^R = 60$ seconds)

Outline

- 1 Replication
- 2 New Strategy
- 3 Digression
- 4 Experiments**
 - Overhead
 - Time To Solution
 - When To Restart

Outline

- 1 Replication
- 2 New Strategy
- 3 Digression
- 4 Experiments**
 - **Overhead**
 - Time To Solution
 - When To Restart

Notations

- *restart*

$\text{Restart}(T)$ and overhead $\mathbb{H}^{\text{rs}}(T)$

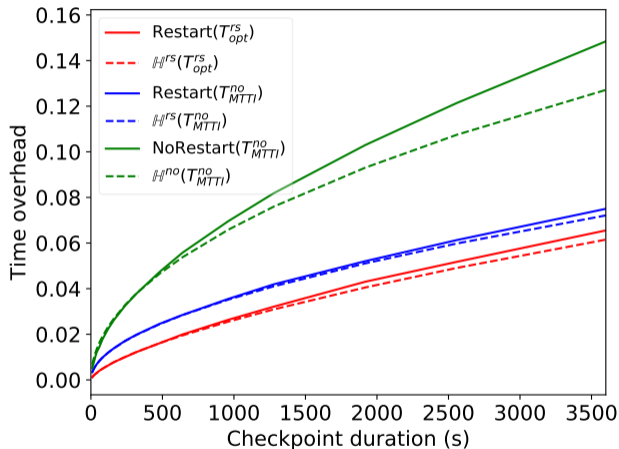
$T_{\text{opt}}^{\text{rs}}$ optimal period

- *no-restart*

$\text{NoRestart}(T)$ and overhead $\mathbb{H}^{\text{no}}(T)$

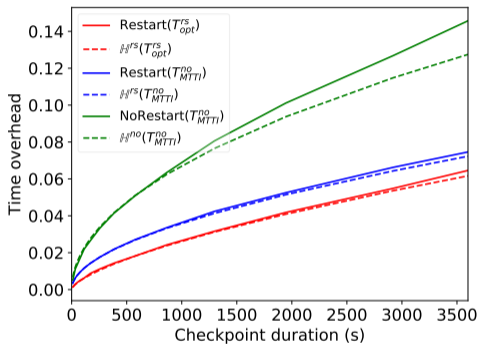
$T_{\text{MTTI}}^{\text{no}}$ used as 'optimal' period (analogy with Young/Daly)

Model Accuracy

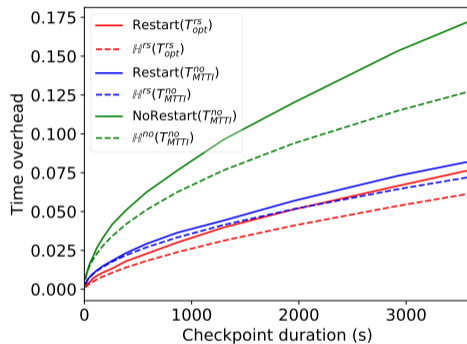


$\mu = 5$ years, $b = 10^5$ processor pairs, $C^R = C$

Model Accuracy With Trace Logs



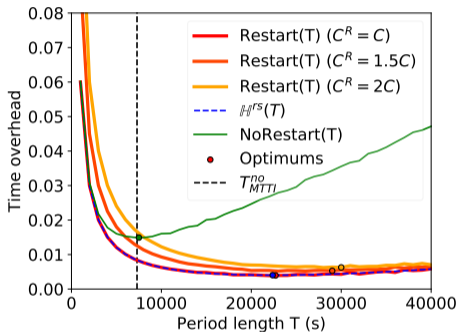
LANL#18



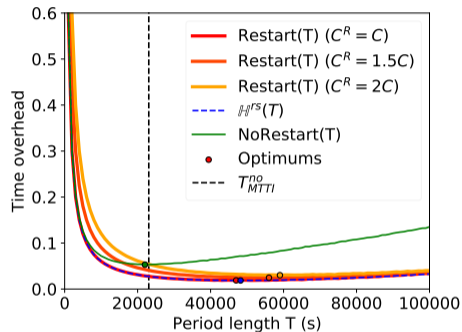
LANL#2

$$\mu = 5 \text{ years}, b = 10^5 \text{ processor pairs}, C^R = C$$

Impact of Checkpointing Period



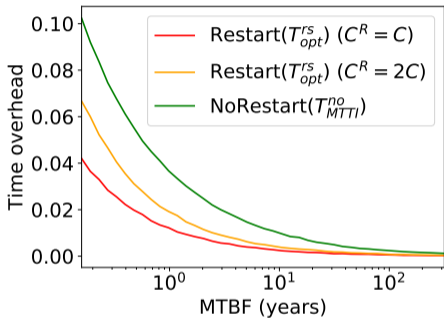
$C = 60$ seconds



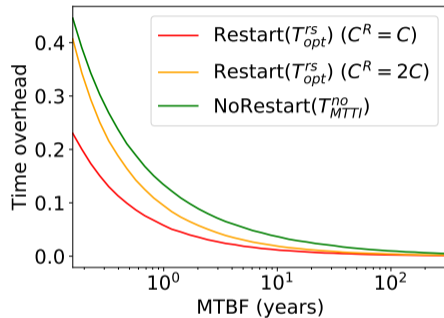
$C = 600$ seconds

$\mu = 5$ years, $b = 10^5$ processor pairs

Impact of MTBF



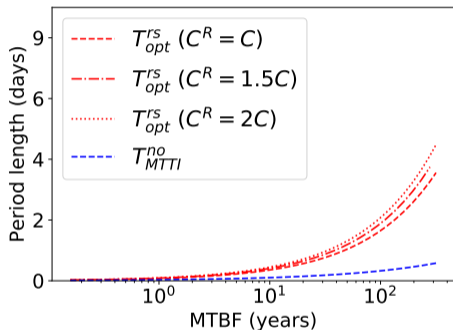
$C = 60$ seconds



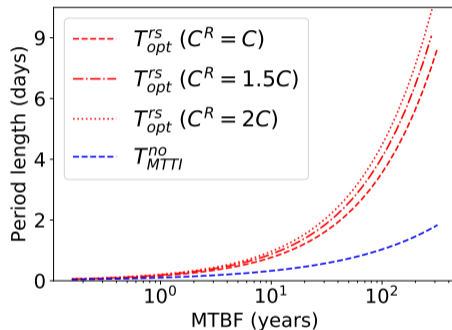
$C = 600$ seconds

$b = 10^5$ processor pairs

I/O Pressure



$C = 60$ seconds



$C = 600$ seconds

$b = 10^5$ processor pairs

Outline

- 1 Replication
- 2 New Strategy
- 3 Digression
- 4 Experiments**
 - Overhead
 - Time To Solution**
 - When To Restart

Time To Solution

No replication, N parallel processors

$$T_{final} = (\mathbb{H}_{opt} + 1) \left(\gamma + \frac{1 - \gamma}{N} \right) T_{seq}, \quad \mathbb{H}_{opt} = \sqrt{\frac{2C}{\mu N}}$$

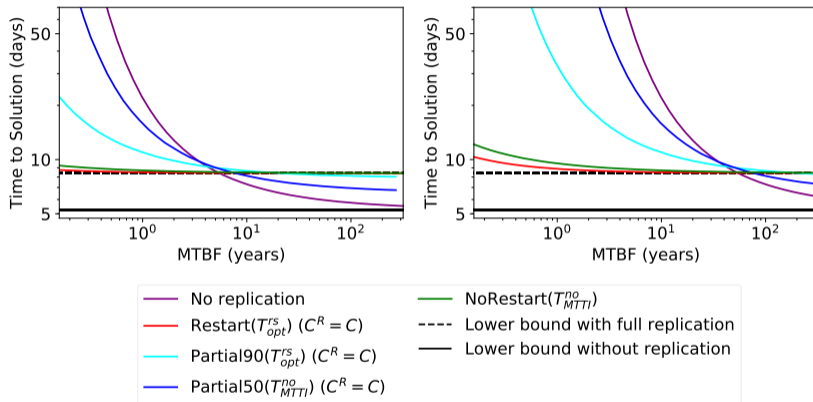
Replication, $N = 2b$, b replica pairs

$$T_{final} = (\mathbb{H}_{opt} + 1)(1 + \alpha) \left(\gamma + \frac{2(1 - \gamma)}{N} \right) T_{seq}$$

$$\text{no-restart} \quad \mathbb{H}_{opt} = \sqrt{\frac{2C}{M_N}}$$

$$\text{restart} \quad \mathbb{H}_{opt} = \left(\frac{3C^R \sqrt{N} \lambda}{2\mu} \right)^{\frac{2}{3}}$$

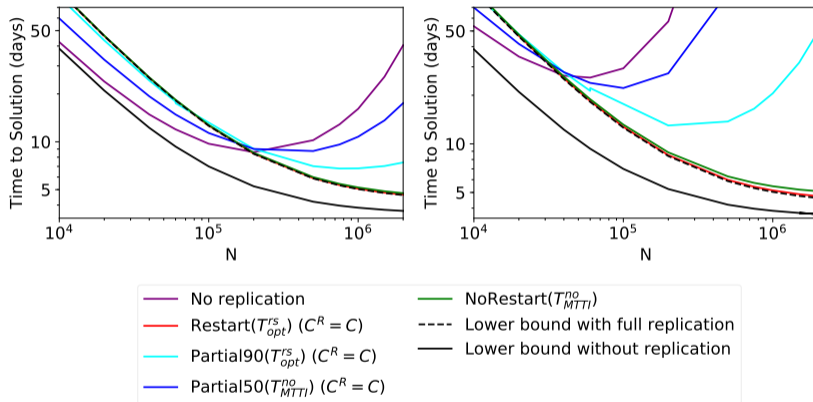
Time To Solution



$$C^R = C = 60 \text{ seconds} \quad C^R = C = 600 \text{ seconds}$$

$$N = 200,000, \gamma = 10^{-5}, \alpha = 0.2$$

Replication Useful?



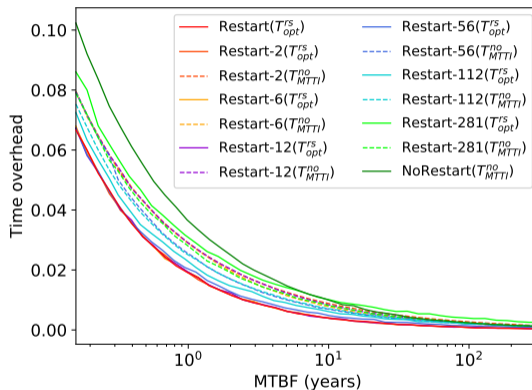
$$C^R = C = 60 \text{ seconds} \quad C^R = C = 600 \text{ seconds}$$

$$\mu = 5 \text{ years}, \gamma = 10^{-5}, \alpha = 0.2$$

Outline

- 1 Replication
- 2 New Strategy
- 3 Digression
- 4 Experiments**
 - Overhead
 - Time To Solution
 - When To Restart**

When To Restart



$$C^R = C = 60 \text{ seconds, } b = 10^5 \text{ and } n_{\text{fail}}(2b) = 561$$

Summary

- Model is realistic 😊
- *restart* with T_{opt}^{rs} is indeed optimal 😊
- Smaller time overheads than *no-restart* with T_{MTTI}^{no} , longer periods, less I/O pressure 😊 😊 😊

- Opinion is divided about replication
- Checkpoint/restart alone cannot ensure full reliability in heavily failure-prone environments
- When replication is needed (large C , short μ , large γ),
magic recipe:
 - use full replication
 - *restart* dead processors at each checkpoint (overlap if possible)
 - use T_{opt}^{rs}

Future Work

Solve the non-periodic problem with one replica pair

... so that we can sleep again !!!!!!!!!!! 😊 😊 😊

Experimentally evaluate non-periodic checkpointing strategies that rejuvenate failed processors after a given number of failures is reached or after a given time interval is exceeded

Revisit partial replication for heterogeneous platforms

Comment trouver des problèmes de recherche rigolos ?

- Broder sur les règles du jeu, en cherchant à avoir un modèle réaliste
- Trouver des instances du problème de complexité non triviale (notamment pour le placement de répliques)
- Partir de problèmes existants, de papiers trouvés dans la littérature qui traitent de sujets de recherche encore ouverts
- Discuter avec ses collègues, son directeur de thèse !