Trade-offs between performance, reliability, and energy consumption

Anne Benoit

LIP, Ecole Normale Supérieure de Lyon, France
Georgia Institute of Technology, Atlanta, USA

Anne.Benoit@ens-lyon.fr
http://graal.ens-lyon.fr/~abenoit/

HPPAC workshop @ IPDPS'18, Vancouver, May 21, 2018

Energy: a crucial issue

- Data centers ("Cloud Begins with Coal", M. Mills)
  - $250 - 350\,TWh$ in 2013
    $\approx$ consumption of Turkey (242), Spain (267), or Italy (309)
  - $\approx 530\,Mt$ of CO_2 (carbontrust) $\rightarrow$ Canada
- Nowadays: more than 90 billion kilowatt-hours of electricity a year; requires 34 giant (500 megawatt) coal-powered plants
- Explosion of artificial intelligence; AI is hungry for processing power! Need to double data centers in next four years $\rightarrow$ how to get enough power?
- Energy and power awareness $\rightsquigarrow$ crucial for both environmental and economical reasons
- Workshop on High-Performance, Power-Aware Computing!

Performance: Exascale platforms

- Hierarchical
  - 10^5 or 10^6 nodes
  - Each node equipped with 10^4 or 10^3 cores

- Failure-prone

MTBF – one node	1 year	10 years	120 years
MTBF – platform of 10^6 nodes	30sec	5mn	1h

More nodes $\Rightarrow$ Shorter MTBF (Mean Time Between Failures)

Exascale $\neq$ Petascale $\times 1000$

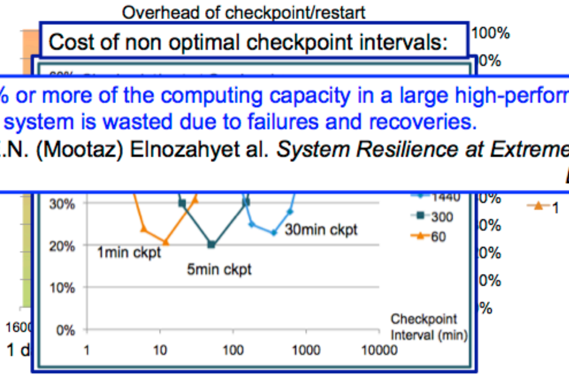
Even at Petascale (courtesy F. Cappello)

Joint Laboratory for Petascale Computation

Also an issue at Petascale

INRIA NCSA

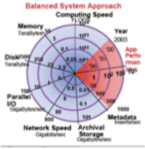
Fault tolerance becomes critical at Petascale (MTTI <= 1day)
Poor fault tolerance design may lead to huge overhead



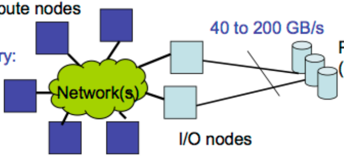
Today, 20% or more of the computing capacity in a large high-performance computing system is wasted due to failures and recoveries.
Dr. E.N. (Mootaz) Elnozahyet al. *System Resilience at Extreme Scale, DARPA*

Even at Petascale (courtesy F. Cappello)

Classic approach for FT: Checkpoint-Restart

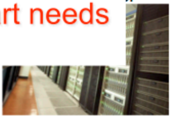


Typical "Balanced Architecture" for PetaScale Computers



RoadRunner

TACC Ranger



⇨ Without optimization, Checkpoint-Restart needs about 1h! (~30 minutes each)

Systems	Perf.	Ckpt time	Source
RoadRunner	1PF	~20 min.	Panasas
LLNL BG/L	500 TF	>20 min.	LLNL
LLNL Zeus	11TF	26 min.	LLNL
YYY BG/P	100 TF	~30 min.	YYY

LLNL BG/L

An inconvenient truth

Top ranked supercomputers in the US (June 2017)

Rank	Name	Laboratory	Technology	Processors	PFlops/s	MTBF
4	Titan	ORNL	Cray XK7	37,376	17.59	$\approx$ 1 day
5	Sequoia	LLNL	BG/Q	98,304	17.17	$\approx$ 1 day
6	Cori	LBNL	Cray XC40	11,308	14.01	$\approx$ 1 day
9	Mira	ANL	BG/Q	49,152	8.59	$\approx$ 1 day

The first exascale computer (10^{18} FLOPS) is expected by 2020:

- Larger processors count: millions of processors
- MTBF is expected to drop dramatically
- Down to **the hour** or even worse

Coping with faults:

- Make applications more fault tolerant, design better resilience techniques...
- ... And don't forget to be green!

An inconvenient truth

Top ranked supercomputers in the US (June 2017)

Rank	Name	Laboratory	Technology	Processors	PFlops/s	MTBF
4	Titan	ORNL	Cray XK7	37,376	17.59	≈ 1 day
5	Sequoia	LLNL	BG/Q	98,304	17.17	≈ 1 day
6	Cori	LBNL	Cray XC40	11,308	14.01	≈ 1 day
9	Mira	ANL	BG/Q	49,152	8.59	≈ 1 day

The first exascale computer (10^{18} FLOPS) is expected by 2020:

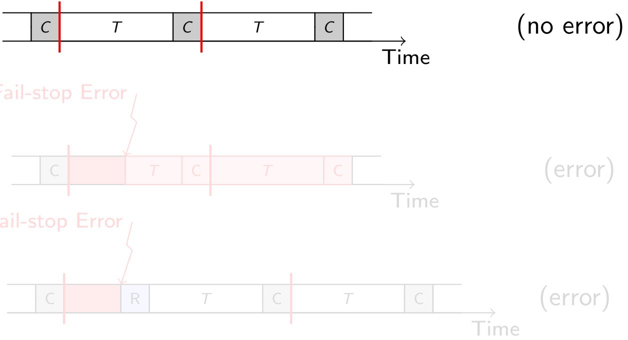
- Larger processors count: millions of processors
- MTBF is expected to drop dramatically
- Down to **the hour** or even worse

Coping with faults:

- Make applications more fault tolerant, design better resilience techniques...
- **... And don't forget to be green!**

Coping with fail-stop errors

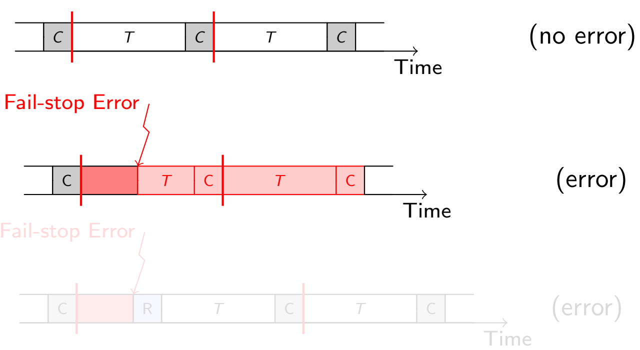
Periodic checkpoint, rollback, and recovery:



- Coordinated checkpointing (the platform is a giant macro-processor)
- Assume instantaneous interruption and detection
- Rollback to last checkpoint and re-execute

Coping with fail-stop errors

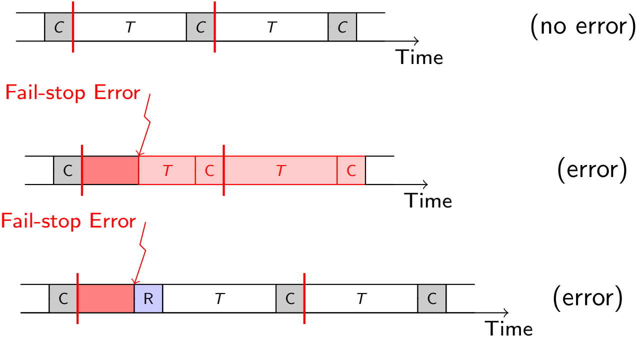
Periodic checkpoint, rollback, and recovery:



- Coordinated checkpointing (the platform is a giant macro-processor)
- Assume instantaneous interruption and detection
- Rollback to last checkpoint and re-execute

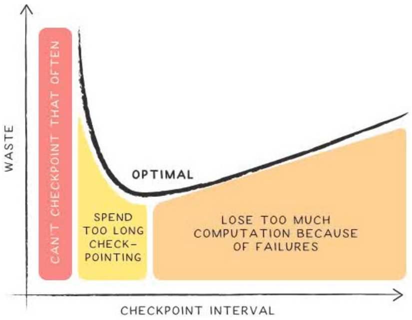
Coping with fail-stop errors

Periodic checkpoint, rollback, and recovery:



- Coordinated checkpointing (the platform is a giant macro-processor)
- Assume instantaneous interruption and detection
- Rollback to last checkpoint and re-execute

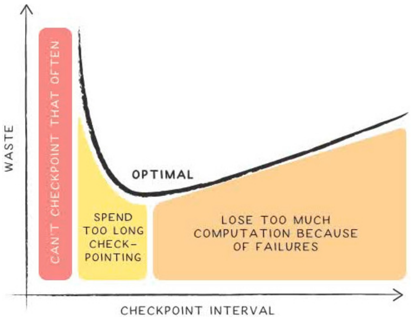
Optimal checkpoint interval (for time)



Theorem. [Young 1974, Daly 2006]

- $T^* = \sqrt{2\mu C}$
- μ: Platform MTBF, C: Checkpointing time
- Is this optimal for energy consumption?

Optimal checkpoint interval (for time)



Theorem. [Young 1974, Daly 2006]

- $T^* = \sqrt{2\mu C}$
- μ: Platform MTBF, C: Checkpointing time

- Is this optimal for energy consumption?

Outline

1. Optimal checkpointing period: time vs. energy
   - Framework
   - Optimal period for execution time
   - Optimal period for energy
   - Experiments
2. A different re-execution speed can help
   - Model and optimization problem
   - Optimal pattern size and speeds
   - Simulations
   - Extensions: both fail-stop and silent errors
3. Summary and need for trade-offs

Motivation

- Coordinated *periodic* checkpointing: what is the optimal checkpointing period if you optimize for Energy consumption?
- Is there a tradeoff between optimizing for Energy and optimizing for Time?

Outline

1. Optimal checkpointing period: time vs. energy
   - Framework
   - Optimal period for execution time
   - Optimal period for energy
   - Experiments
2. A different re-execution speed can help
   - Model and optimization problem
   - Optimal pattern size and speeds
   - Simulations
   - Extensions: both fail-stop and silent errors
3. Summary and need for trade-offs

Power model

- $\mathcal{P}_{\textbf{Static}}$: base power (platform switched on)
  - Trend: goes down (w.r.t. other powers)
- $\mathcal{P}_{\textbf{Cal}}$: overhead due to CPU (computations)
- $\mathcal{P}_{\textbf{I/O}}$: overhead due to file I/O (checkpoint or recovery)
- $\mathcal{P}_{\textbf{Down}}$: overhead when one machine is down (rebooting)

Meneses, Sarood and Kalé:

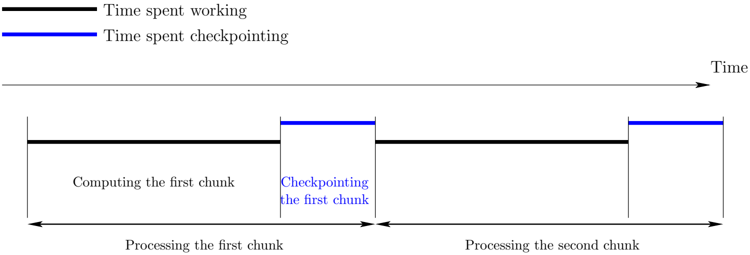
E. Meneses, O. Sarood, and L.V. Kalé, "Assessing Energy Efficiency of Fault Tolerance Protocols for HPC Systems," in Proceedings of the 2012 IEEE 24th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD 2012), New York, USA, October 2012.

- Base power $L = \mathcal{P}_{\text{Static}}$
- Maximum power $H = \mathcal{P}_{\text{Static}} + \mathcal{P}_{\text{Cal}}$
- $\mathcal{P}_{\text{I/O}} = 0$ (and $\mathcal{P}_{\text{Down}} = 0$)

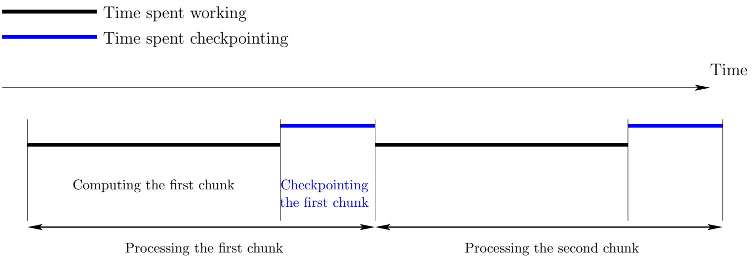
Coordinated checkpointing

- Periodic checkpointing policy of period T
- Independent and identically distributed failures
- Applies to a single processor with MTBF $\mu = \mu_{ind}$
- Applies to a platform with p processors with MTBF $\mu = \frac{\mu_{ind}}{p}$
  - tightly-coupled application
  - progress $\Leftrightarrow$ all processors available

Cost of checkpointing

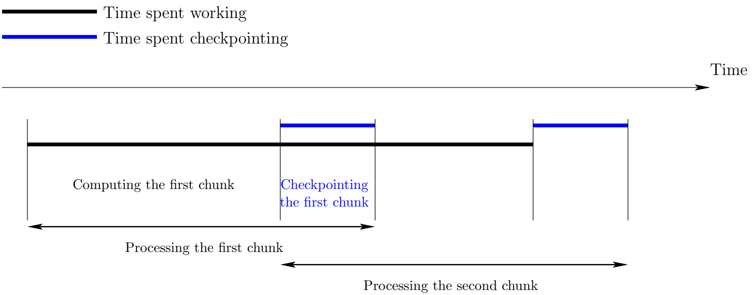


Cost of checkpointing



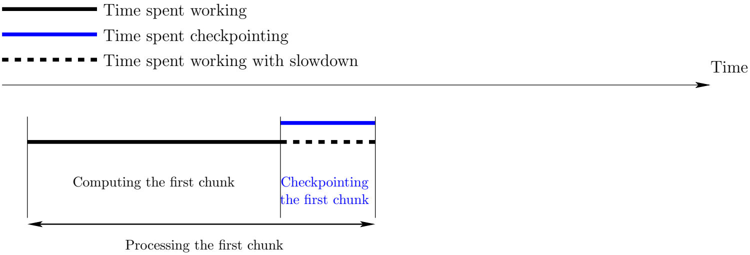
Blocking model: while a checkpoint is taken, no computation can be performed

Cost of checkpointing



Non-blocking model: while a checkpoint is taken, computations are not impacted (e.g., first copy state to RAM, then copy RAM to disk)

Cost of checkpointing



General model: while a checkpoint is taken, computations are slowed-down: during a checkpoint of duration C, the same amount of computation is done as during a time ωC without checkpointing $(0 \leq \omega \leq 1)$.

Outline

1. Optimal checkpointing period: time vs. energy
   - Framework
   - Optimal period for execution time
   - Optimal period for energy
   - Experiments
2. A different re-execution speed can help
   - Model and optimization problem
   - Optimal pattern size and speeds
   - Simulations
   - Extensions: both fail-stop and silent errors
3. Summary and need for trade-offs

Expected execution time

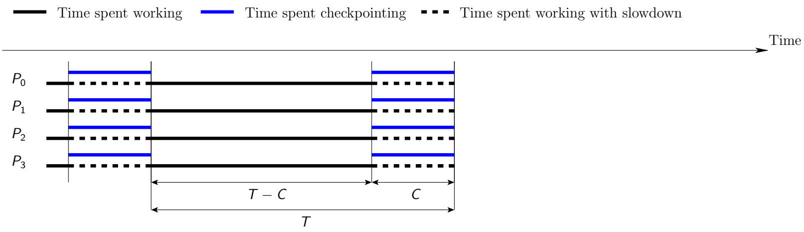
- $\mathcal{T}_{\text{base}}$: execution time without any overhead
- $\mathcal{T}_{\text{final}} = \mathcal{T}_{\text{ff}} + \mathcal{T}_{\text{fails}}$: total execution time
  - Time for fault-free execution

$$\mathcal{T}_{\text{ff}} = \mathcal{T}_{\text{base}} \frac{T}{T - (1-\omega)C}$$

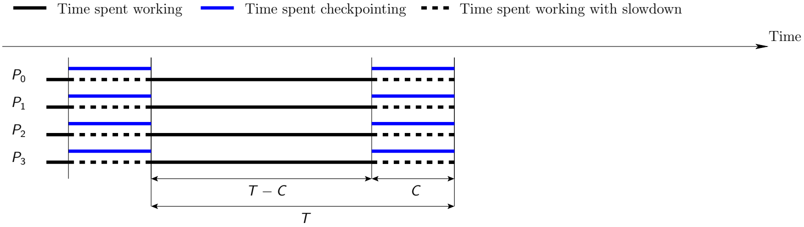
  - Time lost due to failures

$$\mathcal{T}_{\text{fails}} = \frac{\mathcal{T}_{\text{final}}}{\mu}(D + R + \text{RE-EXEC})$$

Computing Waste



Waste in the absence of failure

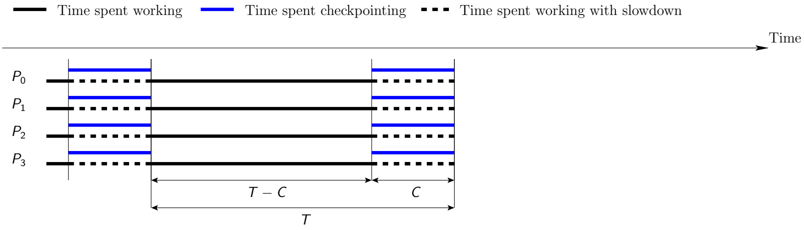


Time elapsed since last checkpoint: T

Amount of computation saved: $(T - C) + \omega C$

$\mathcal{T}_{\text{ff}} = \mathcal{T}_{\text{base}} \frac{T}{T-(1-\omega)C}$

Waste due to failures

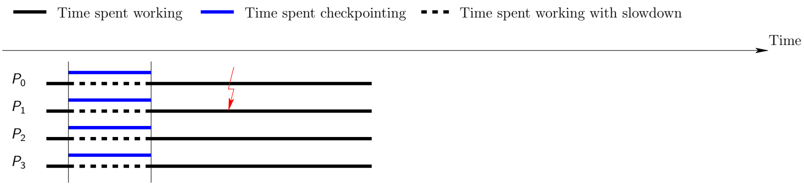


Failure can happen

1. During computation phase
2. During checkpointing phase

RE-EXEC: Time needed for re-execution

Waste due to failures

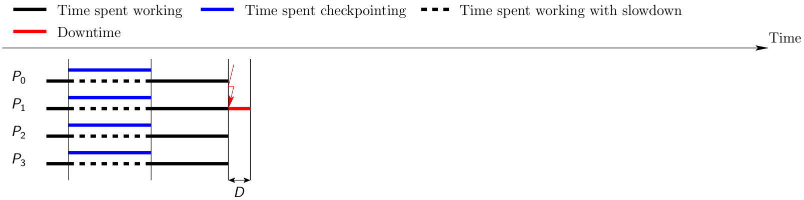


Failure can happen

1. During computation phase
2. During checkpointing phase

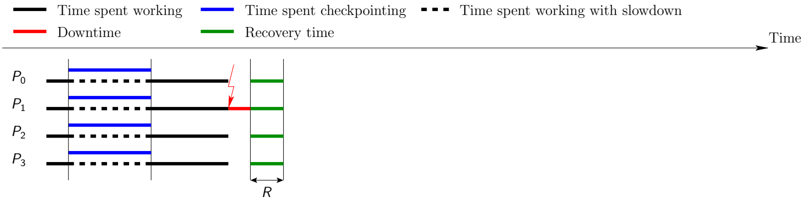
RE-EXEC: Time needed for re-execution

Waste due to failures in computation phase



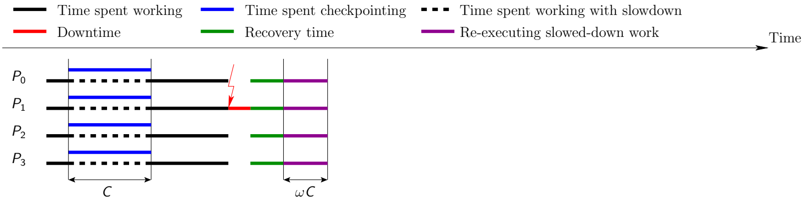
Coordinated checkpointing protocol: when one processor is victim of a failure, all processors lose their work and must roll-back to last checkpoint

Waste due to failures in computation phase



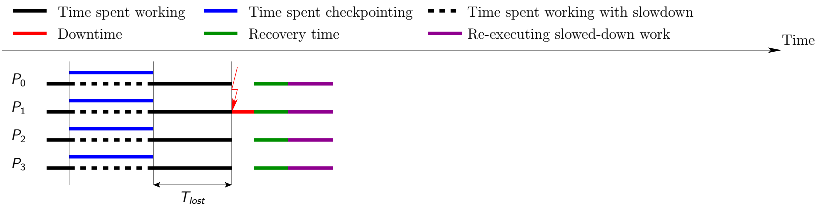
Coordinated checkpointing protocol: All processors must recover from last checkpoint

Waste due to failures in computation phase



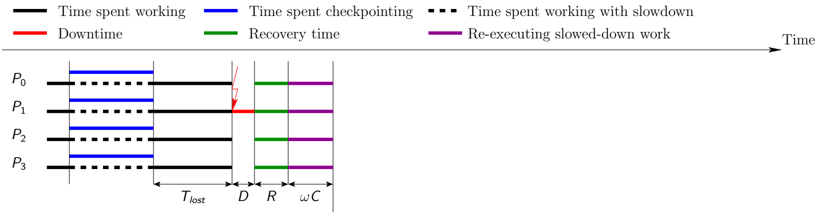
Redo the work destroyed by the failure, that was done in the checkpointing phase before the computation phase

Waste due to failures in computation phase



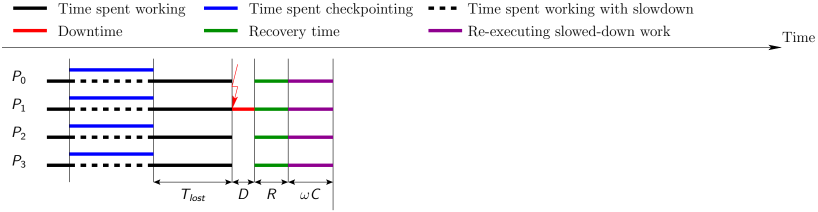
But no checkpoint is taken in parallel, hence this re-computation is faster than the original computation

Waste due to failures in computation phase



Re-execute the computation phase

Waste due to failures in computation phase



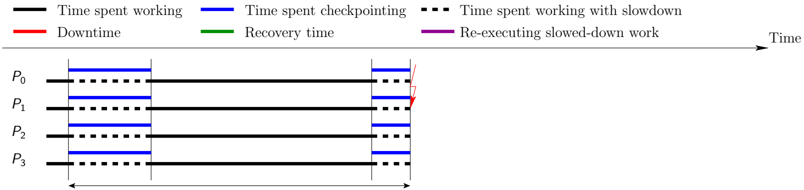
Re-execute the computation phase

RE-EXEC: $\text{RE-EXEC}_{coord-fail-in-work} = T_{lost} + \omega C$

Expectation: $T_{lost} = \frac{1}{2}(T - C)$

$$\text{RE-EXEC}_{coord-fail-in-work} = \frac{T-C}{2} + \omega C$$

Waste due to failures

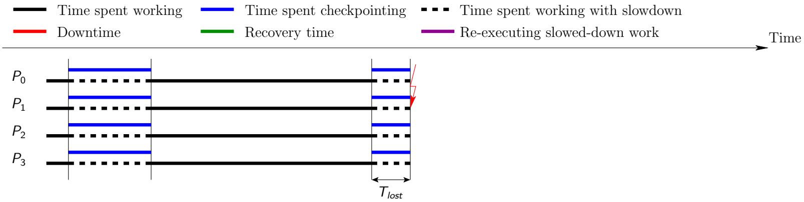


Failure can happen

1. During computation phase
2. During checkpointing phase

RE-EXEC: Time needed for re-execution

Waste due to failures in checkpointing phase



$$\text{RE-EXEC}_{coord-fail-in-checkpoint} = (T - C) + T_{lost} + \omega C$$

Expectation: $T_{lost} = \frac{1}{2}C$

$$\begin{aligned}\text{RE-EXEC}_{coord-fail-in-checkpoint} &= (T - C) + \frac{C}{2} + \omega C \\ &= T - \frac{C}{2} + \omega C\end{aligned}$$

Re-Exec

- Failure in the computation phase (probability: $\frac{T-C}{T}$)

$$\text{Re-Exec}_{coord-fail-in-work} = \frac{T-C}{2} + \omega C$$

- Failure in the checkpointing phase (probability: $\frac{C}{T}$)

$$\text{Re-Exec}_{coord-fail-in-checkpoint} = T - \frac{C}{2} + \omega C$$

$$\text{Re-Exec} = \frac{T-C}{T}\left(\frac{T-C}{2} + \omega C\right) + \frac{C}{T}\left(T - \frac{C}{2} + \omega C\right)$$

$$\text{Re-Exec} = \omega C + \frac{T}{2}$$

AlgoT: Strategy with $\mathcal{T}^{\text{opt}}_{\text{Time}}$

$$\mathcal{T}_{\text{final}} = \mathcal{T}_{\text{base}} \frac{T}{T-(1-\omega)C} + \frac{\mathcal{T}_{\text{final}}}{\mu}\left(D+R+\omega C+\frac{T}{2}\right)$$

$$= \frac{T}{(T-a)\left(b-\frac{T}{2\mu}\right)}\mathcal{T}_{\text{base}}$$

$$a = (1-\omega)C \text{ and } b = 1-\frac{D+R+\omega C}{\mu}$$

$$\mathcal{T}^{\text{opt}}_{\text{Time}} = \sqrt{2(1-\omega)C(\mu-(D+R+\omega C))}$$

Outline

1. Optimal checkpointing period: time vs. energy
   - Framework
   - Optimal period for execution time
   - Optimal period for energy
   - Experiments
2. A different re-execution speed can help
   - Model and optimization problem
   - Optimal pattern size and speeds
   - Simulations
   - Extensions: both fail-stop and silent errors
3. Summary and need for trade-offs

Consumed energy

$$
\begin{aligned}
\mathcal{E}_{\text{final}} &= \mathcal{T}_{\text{Cal}}\mathcal{P}_{\text{Cal}} + \mathcal{T}_{\text{I/O}}\mathcal{P}_{\text{I/O}} + \mathcal{T}_{\text{Down}}\mathcal{P}_{\text{Down}} + \mathcal{T}_{\text{final}}\mathcal{P}_{\text{Static}} \\
&= \left(\mathcal{T}_{\text{base}} + \frac{\mathcal{T}_{\text{final}}}{\mu}\left(\omega C + \frac{T^2 - C^2}{2T} + \frac{\omega C^2}{2T}\right)\right)\mathcal{P}_{\text{Cal}} \\
&\quad + \left(\frac{\mathcal{T}_{\text{final}}}{\mu}\left(R + \frac{C^2}{2T}\right) + C\frac{\mathcal{T}_{\text{base}}}{T - (1-\omega)C}\right)\mathcal{P}_{\text{I/O}} \\
&\quad + \frac{\mathcal{T}_{\text{final}}}{\mu} D\mathcal{P}_{\text{Down}} + \mathcal{T}_{\text{final}}\mathcal{P}_{\text{Static}}
\end{aligned}
$$

$\mathcal{T}_{\text{final}} \neq \mathcal{T}_{\text{Cal}} + \mathcal{T}_{\text{I/O}} + \mathcal{T}_{\text{Down}}$, unless $\omega = 0$

CPU and I/O activities are overlapped (and both consumed) when checkpointing

ALGOE: Strategy with $\mathcal{T}^{\text{opt}}_{\text{Energy}}$

$$\mathcal{P}_{\text{Cal}} = \alpha \mathcal{P}_{\text{Static}},\ \mathcal{P}_{\text{I/O}} = \beta \mathcal{P}_{\text{Static}},\ \mathcal{P}_{\text{Down}} = \gamma \mathcal{P}_{\text{Static}}$$

$$\begin{aligned}
\frac{(T-a)^2\left(b-\frac{T}{2\mu}\right)^2}{\mathcal{P}_{\text{Static}}\mathcal{T}_{\text{base}}}\mathcal{E}'_{\text{final}} &= \frac{-ab+\frac{T^2}{2\mu}}{\mu}\left((\alpha\omega C+\beta R+\gamma D+\mu)+\frac{\alpha T}{2}+\frac{\alpha(1-\omega)C^2}{2T}+\frac{\beta C^2}{2T}\right)\\
&\quad+\frac{(T-a)(b-\frac{T}{2\mu})}{2\mu}\left(\alpha+\frac{\alpha(1-\omega)C^2-\beta C^2}{T}\right)-\beta C\left(b-\frac{T}{2\mu}\right)^2\\
&= T^3\left(\frac{1}{4\mu}-\frac{1}{4\mu}\right)+T^2\left(\frac{\alpha\omega C+\beta R+\gamma D}{2\mu^2}+\frac{b+\frac{a}{2\mu}}{2\mu}-\frac{\beta C}{4\mu^2}+\frac{1}{2\mu}\right)\\
&\quad+T\left(-\frac{ab}{2\mu}-\frac{ab}{2\mu}+\frac{\beta Cb}{\mu}-2\frac{(\alpha(1-\omega)-\beta)C^2}{4\mu^2}\right)-\beta Cb^2\\
&\quad-\frac{ab(\alpha\omega C+\beta R+\gamma D+\mu)}{\mu}-\left(\frac{b}{2\mu}-\frac{a}{4\mu^2}\right)(\alpha(1-\omega)-\beta)C^2\\
&\quad+\frac{1}{T}\left((\alpha(1-\omega)-\beta)\frac{C}{2\mu}-(\alpha(1-\omega)-\beta)\frac{C}{2\mu}\right)\\
&= T^2\left(\frac{\alpha\omega C+\beta R+\gamma D}{2\mu^2}+\frac{b}{2\mu}+\frac{a-\beta C}{4\mu^2}+\frac{1}{2\mu}\right)\\
&\quad+T\left(\frac{(\beta C-a)b}{\mu}-2\frac{(\alpha(1-\omega)-\beta)C^2}{4\mu^2}\right)\\
&\quad-\frac{ab(\alpha\omega C+\beta R+\gamma D+\mu)}{\mu}-\beta Cb^2\\
&\quad+\left(\frac{b}{2\mu}+\frac{a}{4\mu^2}\right)(\alpha(1-\omega)-\beta)C^2\ .
\end{aligned}$$

ALGOE: Strategy with $\mathcal{T}^{\text{opt}}_{\text{Energy}}$

$$\mathcal{P}_{\text{Cal}} = \alpha \mathcal{P}_{\text{Static}},\ \mathcal{P}_{\text{I/O}} = \beta \mathcal{P}_{\text{Static}},\ \mathcal{P}_{\text{Down}} = \gamma \mathcal{P}_{\text{Static}}$$

$$\frac{(T-a)^2 \left(b \ [illegible] \right)^2}{\mathcal{P}_{\text{Static}} \mathcal{T}_{\text{base}}} = \frac{-ab + \frac{T^2}{2\mu}}{\mu} \left((\alpha\omega C + \beta R + \gamma D + \mu) + \frac{\alpha T}{[illegible]} \ [illegible] \ \frac{[illegible]\omega)C^2}{2T} + \frac{\beta C^2}{2T} \right)$$

$$+ \ [illegible] \left(b - \frac{T}{2\mu} \right) \left(\alpha + \frac{\alpha(1 - [illegible]}{[illegible]} \ [illegible] C^2 \right) - \beta C \left(b - \frac{T}{2\mu} \right)^2$$

$$= T^3 \left(\frac{1}{4\mu} - \frac{1}{4[illegible]} \ [illegible] \ \frac{\alpha\omega C + \beta R + \gamma D}{2\mu^2} + \frac{b + \frac{a}{2\mu}}{2\mu} - \frac{\beta C}{4\mu^2} + \frac{1}{2\mu} \right)$$

$$+ T \left([illegible] - \frac{ab}{2\mu} + \frac{\beta C [illegible]}{\mu} \ [illegible] \ \frac{(\alpha(1-\omega) - \beta) C^2}{4\mu^2} \right) - \beta C b^2$$

$$[illegible] \frac{[illegible]\alpha\omega C + \beta R + \gamma D + \mu)}{\mu} - \left(\frac{b}{2\mu} \ [illegible] \right) (\alpha(1-\omega) - \beta) C^2$$

$$+ \frac{1}{T} \left((\alpha(1-\omega) - \beta) \frac{C}{2\mu} - (\alpha(1 \ [illegible] - \beta) \frac{C}{2\mu} \right)$$

$$= T^2 \left(\frac{\alpha\omega C + \beta R + \gamma D}{2\mu^2} + \frac{b}{2\mu} + \frac{a - \beta C}{4\mu^2} + \frac{1}{2\mu} \right)$$

$$+ T \left(\frac{(\beta C - a) b}{\mu} - 2 \frac{(\alpha(1-\omega) - \beta) C^2}{4\mu^2} \right)$$

We let Maple compute $\mathcal{T}^{\text{opt}}_{\text{Energy}}$

Outline

1. Optimal checkpointing period: time vs. energy
   - Framework
   - Optimal period for execution time
   - Optimal period for energy
   - Experiments
2. A different re-execution speed can help
   - Model and optimization problem
   - Optimal pattern size and speeds
   - Simulations
   - Extensions: both fail-stop and silent errors
3. Summary and need for trade-offs

Parameters: power

$$\rho = \frac{\mathcal{P}_{\text{Static}} + \mathcal{P}_{\text{I/O}}}{\mathcal{P}_{\text{Static}} + \mathcal{P}_{\text{Cal}}} = \frac{1+\beta}{1+\alpha}$$

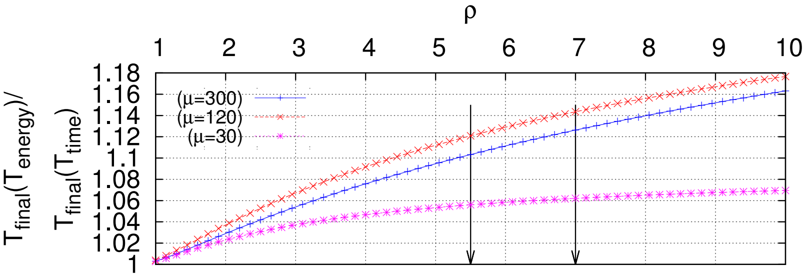
- 20 Mega-watts for Exascale platform with 10^6 nodes
- Nominal power = 20 milli-watts per node
- $1/2 \longrightarrow 1/4$ of that power in static consumption
- "I/O an order of magnitude more than computing" (J. Shalf, S. Dosanjh, and J. Morrison, "Exascale computing technology challenges," in the 9th Int. Conf. High Performance Computing for Computational Science, 2011)

- Scenario 1: $\mathcal{P}_{\text{Static}} = 10$, $\mathcal{P}_{\text{Cal}} = 10$, $\mathcal{P}_{\text{I/O}} = 100 \Rightarrow \rho = 5.5$
- Scenario 2: $\mathcal{P}_{\text{Static}} = 5$, $\mathcal{P}_{\text{Cal}} = 10$, $\mathcal{P}_{\text{I/O}} = 100 \Rightarrow \rho = 7$

Parameters: resilience

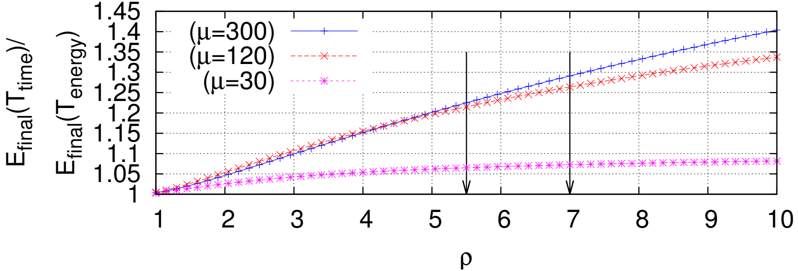
- MTBF
  - $N = 45,208$ processors: one fault per day
  - Individual (processor) MTBF $\mu_{\text{ind}} \approx 125$ years.
  - Total number of processors N: from $N = 219,150$ to $N = 2,191,500 \Rightarrow \mu = 300$ min down to $\mu = 30$ min
- $C = R = 10$ min, $D = 1$ min, and $\omega = 1/2$.

Impact of ratio ρ



How much slower, if we optimize for energy instead of optimizing for time

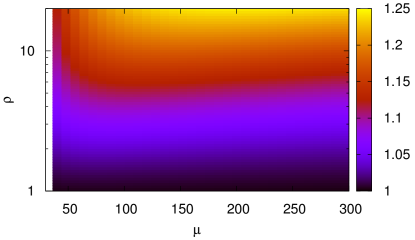
Impact of ratio ρ



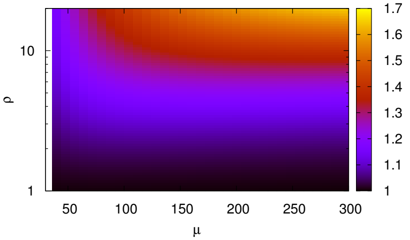
How much more energy consumption, if we optimize for time instead of optimizing for energy

ALGOT over ALGOE

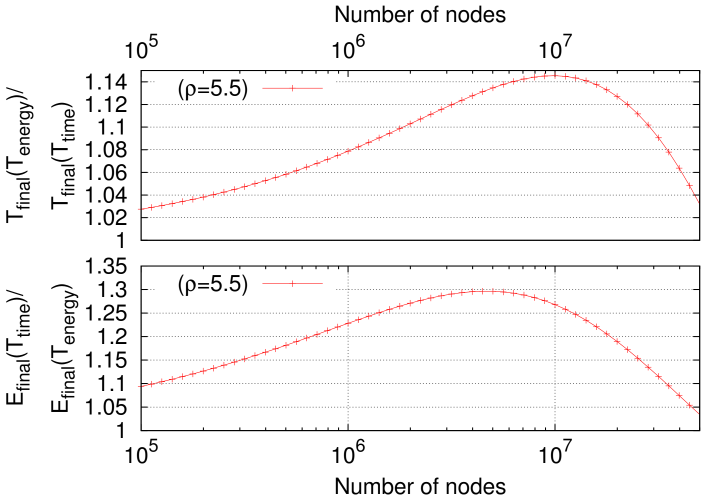
How much slower, if we optimize for energy instead of optimizing for time



How much more energy consumption, if we optimize for time instead of optimizing for energy

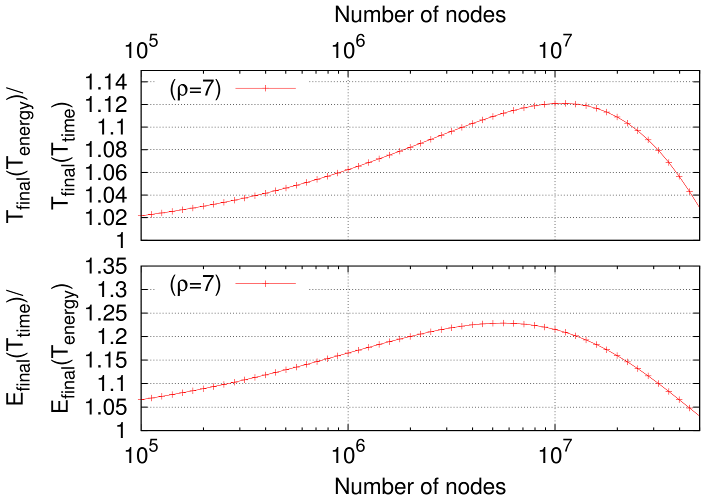


Scalability ($\rho = 5.5$)



$\mu = 120$ min for 10^6 nodes, $C = R = 1$ min, $D = 0.1$ min, $\omega = 1/2$

Scalability ($\rho = 7$)



$\mu = 120$ min for 10^6 nodes, $C = R = 1$ min, $D = 0.1$ min, $\omega = 1/2$

Conclusion

- Coordinated checkpointing, non-blocking
- Different optimal periods for time and energy
- Save more than 20% of energy with 10% increase in time
- Expect more gains for large-scale platforms

- Variety of resilience and power consumption parameters ☹
- Quite flexible analytical model ☺
- Easy to instantiate for other scenarios ☺

Conclusion

- Coordinated checkpointing, non-blocking
- Different optimal periods for time and energy
- Save more than 20% of energy with 10% increase in time
- Expect more gains for large-scale platforms

- Variety of resilience and power consumption parameters ☹
- Quite flexible analytical model ☺
- Easy to instantiate for other scenarios ☺

Outline

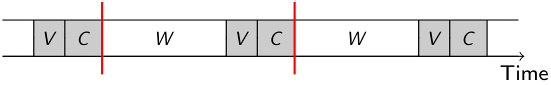
1. Optimal checkpointing period: time vs. energy
   - Framework
   - Optimal period for execution time
   - Optimal period for energy
   - Experiments

2. A different re-execution speed can help
   - Model and optimization problem
   - Optimal pattern size and speeds
   - Simulations
   - Extensions: both fail-stop and silent errors

3. Summary and need for trade-offs

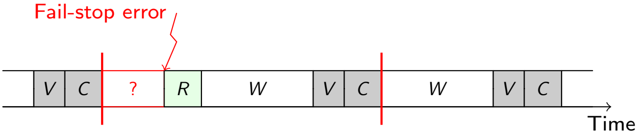
Silent errors

- Another major challenge for Exascale: frequent striking of silent errors
- How to deal with these errors? Add a verification to the classical Checkpoint/Restart protocol
- Verification mechanism: general-purpose (replication, triplication) or application-specific
- *Verified checkpoints*: a verification is performed just before each checkpoint

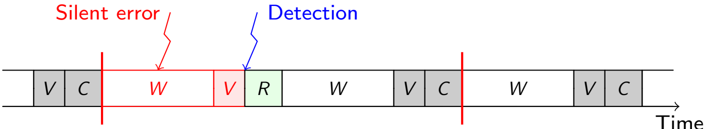


Silent vs Fail-stop errors

- C: time to checkpoint; λ: error rate (platform MTBF $\mu = 1/\lambda$);
  V: time to verify; R: time to recover
- Optimal checkpointing period W for fail-stop errors (Young/Daly): $W = \sqrt{2C/\lambda}$ $(V = 0)$



- Silent errors: $W = \sqrt{(V+C)/\lambda}$ ($C \rightarrow V + C$; missing factor 2)



Back to energy consumption

- Power requirement of current petascale platforms = small town
- Need to reduce energy consumption of future platforms
- Popular technique: dynamic voltage and frequency scaling (DVFS)
- Lower speed → energy savings: when computing at speed σ, power proportional to σ^3 and execution time proportional to $1/\sigma$
  → (dynamic) energy proportional to σ^2
- Also account for static energy: trade-offs to be found
- Realistic approach: minimize energy while guaranteeing a performance bound
- ⇒ At which speed should we execute the workload?

Back to energy consumption

- Power requirement of current petascale platforms = small town
- Need to reduce energy consumption of future platforms
- Popular technique: dynamic voltage and frequency scaling (DVFS)
- Lower speed $\rightarrow$ energy savings: when computing at speed σ, power proportional to σ^3 and execution time proportional to $1/\sigma$
  $\rightarrow$ (dynamic) energy proportional to σ^2
- Also account for static energy: trade-offs to be found
- Realistic approach: minimize energy while guaranteeing a performance bound
- $\Rightarrow$ At which speed should we execute the workload?

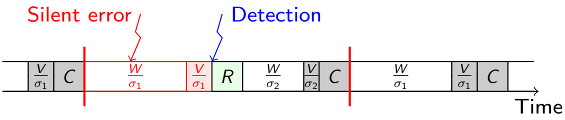
Outline

Framework

- Divisible-load applications, blocking model
- Subject to silent data corruption
- Checkpoint/restart strategy: periodic patterns that repeat over time
- Verified checkpoints
- Is it better to use two different speeds rather than only one? What are the optimal checkpointing period and optimal execution speeds?

Model

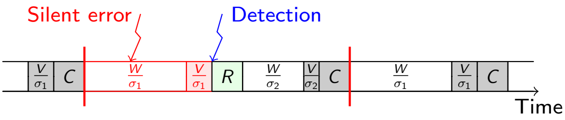
- Set of speeds $S = \{s_1, \ldots, s_K\}$: $\sigma_1 \in S$ speed for first execution, $\sigma_2 \in S$ speed for re-executions
- Silent errors: exponential distribution of rate λ
- Verification: V units of work; Checkpointing: time C; Recovery: time R
- P_{idle} and P_{io} constant; and $P_{\text{cpu}}(\sigma) = \kappa\sigma^3$
- Energy for W units of work at speed σ: $\frac{W}{\sigma}(P_{\text{idle}} + \kappa\sigma^3)$
  Energy of a verification at speed σ: $\frac{V}{\sigma}(P_{\text{idle}} + \kappa\sigma^3)$
  Energy of a checkpoint: $C(P_{\text{idle}} + P_{\text{io}})$
  Energy of a recovery: $R(P_{\text{idle}} + P_{\text{io}})$



With a silent error

Model

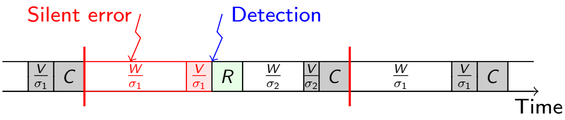
- Set of speeds $S = \{s_1, \dots, s_K\}$: $\sigma_1 \in S$ speed for first execution, $\sigma_2 \in S$ speed for re-executions
- Silent errors: exponential distribution of rate λ
- Verification: V units of work; Checkpointing: time C; Recovery: time R
- P_{idle} and P_{io} constant; and $P_{\text{cpu}}(\sigma) = \kappa\sigma^3$
- Energy for W units of work at speed σ: $\frac{W}{\sigma}(P_{\text{idle}} + \kappa\sigma^3)$
  Energy of a verification at speed σ: $\frac{V}{\sigma}(P_{\text{idle}} + \kappa\sigma^3)$
  Energy of a checkpoint: $C(P_{\text{idle}} + P_{\text{io}})$
  Energy of a recovery: $R(P_{\text{idle}} + P_{\text{io}})$



With a silent error

Model

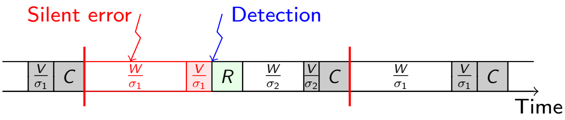
- Set of speeds $S = \{s_1, \ldots, s_K\}$: $\sigma_1 \in S$ speed for first execution, $\sigma_2 \in S$ speed for re-executions
- Silent errors: exponential distribution of rate λ
- Verification: V units of work; Checkpointing: time C; Recovery: time R
- P_{idle} and P_{io} constant; and $P_{\text{cpu}}(\sigma) = \kappa\sigma^3$
- Energy for W units of work at speed σ: $\frac{W}{\sigma}(P_{\text{idle}} + \kappa\sigma^3)$
  Energy of a verification at speed σ: $\frac{V}{\sigma}(P_{\text{idle}} + \kappa\sigma^3)$
  Energy of a checkpoint: $C(P_{\text{idle}} + P_{\text{io}})$
  Energy of a recovery: $R(P_{\text{idle}} + P_{\text{io}})$



With a silent error

Model

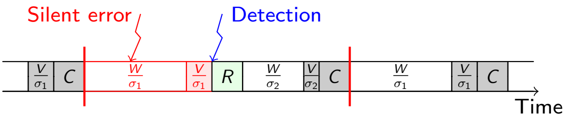
- Set of speeds $S = \{s_1, \ldots, s_K\}$: $\sigma_1 \in S$ speed for first execution, $\sigma_2 \in S$ speed for re-executions
- Silent errors: exponential distribution of rate λ
- Verification: V units of work; Checkpointing: time C; Recovery: time R
- P_{idle} and P_{io} constant; and $P_{\text{cpu}}(\sigma) = \kappa\sigma^3$
- Energy for W units of work at speed σ: $\frac{W}{\sigma}(P_{\text{idle}} + \kappa\sigma^3)$
  Energy of a verification at speed σ: $\frac{V}{\sigma}(P_{\text{idle}} + \kappa\sigma^3)$
  Energy of a checkpoint: $C(P_{\text{idle}} + P_{\text{io}})$
  Energy of a recovery: $R(P_{\text{idle}} + P_{\text{io}})$



With a silent error

Model

- Set of speeds $S = \{s_1, \ldots, s_K\}$: $\sigma_1 \in S$ speed for first execution, $\sigma_2 \in S$ speed for re-executions
- Silent errors: exponential distribution of rate λ
- Verification: V units of work; Checkpointing: time C; Recovery: time R
- P_{idle} and P_{io} constant; and $P_{\text{cpu}}(\sigma) = \kappa\sigma^3$
- Energy for W units of work at speed σ: $\frac{W}{\sigma}(P_{\text{idle}} + \kappa\sigma^3)$
  Energy of a verification at speed σ: $\frac{V}{\sigma}(P_{\text{idle}} + \kappa\sigma^3)$
  Energy of a checkpoint: $C(P_{\text{idle}} + P_{\text{io}})$
  Energy of a recovery: $R(P_{\text{idle}} + P_{\text{io}})$



With a silent error

Problem

Optimization problem BiCrit:

$$\text{Minimize } \frac{\mathcal{E}(W,\sigma_1,\sigma_2)}{W} \text{ s.t. } \frac{\mathcal{T}(W,\sigma_1,\sigma_2)}{W} \leq \rho,$$

- $\mathcal{E}(W,\sigma_1,\sigma_2)$ is the expected energy consumed to execute W units of work at speed σ_1, with eventual re-executions at speed σ_2
- $\mathcal{T}(W,\sigma_1,\sigma_2)$ is the expected execution time to execute W units of work at speed σ_1, with eventual re-executions at speed σ_2
- ρ is a performance bound, or admissible degradation factor

Outline

Computing expected execution time

Proposition (1)

For the BiCrit *problem with a single speed,*

$$\mathcal{T}(W,\sigma,\sigma) = C + e^{\frac{\lambda W}{\sigma}}\left(\frac{W+V}{\sigma}\right) + \left(e^{\frac{\lambda W}{\sigma}} - 1\right) R$$

Proposition (2)

For the BiCrit *problem,*

$$\mathcal{T}(W,\sigma_1,\sigma_2) = C + \frac{W+V}{\sigma_1} + \left(1 - e^{-\frac{\lambda W}{\sigma_1}}\right) e^{\frac{\lambda W}{\sigma_2}} \left(R + \frac{W+V}{\sigma_2}\right)$$

Proof of Proposition 1

Proof.

The recursive equation to compute $\mathcal{T}(W,\sigma,\sigma)$ writes:

$$\mathcal{T}(W,\sigma,\sigma) = \frac{W+V}{\sigma} + p(W/\sigma)\left(R + \mathcal{T}(W,\sigma,\sigma)\right) + (1-p(W/\sigma))C,$$

where $p(W/\sigma) = 1 - e^{-\frac{\lambda W}{\sigma}}$. The reasoning is as follows:

- We always execute W units of work followed by the verification, in time $\frac{W+V}{\sigma}$;
- With probability $p(W/\sigma)$, a silent error occurred and is detected, in which case we recover and start anew;
- Otherwise, with probability $1-p(W/\sigma)$, we simply checkpoint after a successful execution.

Solving this equation leads to the expected execution time. □

Proof of Proposition 2

Proof.

The recursive equation to compute $\mathcal{T}(W, \sigma_1, \sigma_2)$ writes:

$$\mathcal{T}(W, \sigma_1, \sigma_2) = \frac{W+V}{\sigma_1} + p(W/\sigma_1)\left(R + \mathcal{T}(W, \sigma_2, \sigma_2)\right) + (1 - p(W/\sigma_1))C,$$

where $p(W/\sigma_1) = 1 - e^{-\frac{\lambda W}{\sigma_1}}$. The reasoning is as follows:

- We always execute W units of work followed by the verification, in time $\frac{W+V}{\sigma_1}$;
- With probability $p(W/\sigma_1)$, a silent error occurred and is detected, in which case we recover and start anew at speed σ_2;
- Otherwise, with probability $1 - p(W/\sigma_1)$, we simply checkpoint after a successful execution.

Solving this equation leads to the expected execution time. □

Computing expected energy consumption

Proposition

For the BiCrit *problem,*

$$\mathcal{E}(W,\sigma_1,\sigma_2) = \left(C + \left(1 - e^{-\frac{\lambda W}{\sigma_1}}\right) e^{\frac{\lambda W}{\sigma_2}} R\right)(P_{\text{io}} + P_{\text{idle}}) + \frac{W+V}{\sigma_1}(\kappa\sigma_1^3 + P_{\text{idle}}) + \frac{W+V}{\sigma_2}(1 - e^{-\frac{\lambda W}{\sigma_1}}) e^{\frac{\lambda W}{\sigma_2}} (\kappa\sigma_2^3 + P_{\text{idle}})$$

Power spent during checkpoint or recovery: $P_{\text{io}} + P_{\text{idle}}$; power spent during computation and verification at speed σ: $P_{\text{cpu}}(\sigma) + P_{\text{idle}} = \kappa\sigma^3 + P_{\text{idle}}$. From Proposition 2, we get the expression of $\mathcal{E}(W,\sigma_1,\sigma_2)$.

Finding optimal pattern length (1)

To get closed-form expression for optimal value of W, use of first-order approximations, using Taylor expansion $e^{\lambda W} = 1 + \lambda W + O(\lambda^2 W^2)$:

$$\frac{\mathcal{T}(W,\sigma_1,\sigma_2)}{W} = \frac{1}{\sigma_1} + \frac{\lambda W}{\sigma_1\sigma_2} + \frac{\lambda R}{\sigma_1} + \frac{\lambda V}{\sigma_1\sigma_2} + \frac{C + V/\sigma_1}{W} + O(\lambda^2 W) \quad (1)$$

$$\frac{\mathcal{E}(W,\sigma_1,\sigma_2)}{W} = \frac{\kappa\sigma_1^3 + P_{\text{idle}}}{\sigma_1} + \frac{\lambda W}{\sigma_1\sigma_2}(\kappa\sigma_2^3 + P_{\text{idle}}) + \frac{\lambda R}{\sigma_1}(P_{\text{io}} + P_{\text{idle}}) + \frac{\lambda V}{\sigma_1\sigma_2}(\kappa\sigma_1^3 + P_{\text{idle}}) + \frac{C(P_{\text{io}} + P_{\text{idle}}) + V(\kappa\sigma_1^3 + P_{\text{idle}})/\sigma_1}{W} + O(\lambda^2 W) \quad (2)$$

Finding optimal pattern length (2)

Theorem

Given σ_1, σ_2 and ρ, consider the equation $aW^2 + bW + c = 0$, where $a = \frac{\lambda}{\sigma_1\sigma_2}$, $b = \frac{1}{\sigma_1} + \lambda\left(\frac{R}{\sigma_1} + \frac{V}{\sigma_1\sigma_2}\right) - \rho$ and $c = C + \frac{V}{\sigma_1}$.

- *If there is no positive solution to the equation, i.e., $b > -2\sqrt{ac}$, then* BiCrit *has no solution.*
- *Otherwise, let W_1 and W_2 be the two solutions of the equation with $W_1 \leq W_2$ (at least W_2 is positive and possibly $W_1 = W_2$). Then, the optimal pattern size is*

$$W_{\text{opt}} = \min(\max(W_1, W_e), W_2), \tag{3}$$

where

$$W_e = \sqrt{\frac{C(P_{\text{io}} + P_{\text{idle}}) + \frac{V}{\sigma_1}(\kappa\sigma_1^3 + P_{\text{idle}})}{\frac{\lambda}{\sigma_1\sigma_2}(\kappa\sigma_2^3 + P_{\text{idle}})}}. \tag{4}$$

Finding optimal pattern length (3)

Proof.

Neglecting lower-order terms, Equation (2) is minimized when $W = W_e$ given by Equation (4).

Two cases:

- ρ is too small $\Rightarrow$ no solution
- $W_2 > 0$:
  - $W_e < W_1$
  - $W_1 \leq W_e \leq W_2$
  - $W_e > W_2$

Using that the energy overhead is a convex function, we get the result (W_{opt} is in the interval $[W_1, W_2]$) □

Finding optimal speed pair

- Speed pair (s_i, s_j), with $1 \leq i, j \leq K$: $\rho_{i,j}$ is the minimum performance bound for which the BiCrit problem with $\sigma_1 = s_i$ and $\sigma_2 = s_j$ admits a solution
- For each speed pair, compute W_1, W_2 the roots of $aW^2 + bW + c$; discard pairs with $\rho < \rho_{i,j}$
- For each remaining speed pair (σ_1, σ_2), compute W_{opt} and associated energy overhead
- Select speed pair (σ_1^*, σ_2^*) that minimizes energy overhead
- Time $O(K^2)$, where K is the number of available speeds, usually a small constant

Outline

Simulation setup

- Platform parameters, based on real platforms

Platform	λ	$C = R$	V
Hera	3.38e-6	$300s$	15.4
Atlas	7.78e-6	$439s$	9.1
Coastal	2.01e-6	$1051s$	4.5
Coastal SSD	2.01e-6	$2500s$	180.0

- Power parameters, determined by the processor used

Processor	Normalized speeds	$P(\sigma)$ (mW)
Intel Xscale	$0.15, 0.4, 0.6, 0.8, 1$	$1550\sigma^3 + 60$
Transmeta Crusoe	$0.45, 0.6, 0.8, 0.9, 1$	$5756\sigma^3 + 4.4$

- Default values: P_{io} equivalent to power used when running at lowest speed; $\rho = 3$

Simulation results, using Hera/XScale configuration

A different re-execution speed does help!

And all speed pairs can be optimal solutions (depending on ρ)!

σ_1	**Best** σ_2	W_{opt}	$\frac{\mathcal{E}(W_{opt},\sigma_1,\sigma_2)}{W_{opt}}$
0.15	0.4	1711	466
0.4	**0.4**	2764	416
0.6	0.4	3639	674
0.8	0.4	4627	1082
1	0.4	5742	1625

$\rho = 8$

σ_1	**Best** σ_2	W_{opt}	$\frac{\mathcal{E}(W_{opt},\sigma_1,\sigma_2)}{W_{opt}}$
0.15	-	-	-
0.4	**0.4**	2764	416
0.6	0.4	3639	674
0.8	0.4	4627	1082
1	0.4	5742	1625

$\rho = 3$

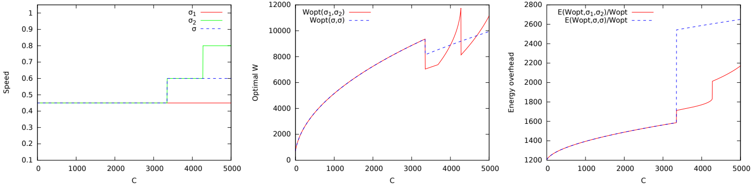
σ_1	**Best** σ_2	W_{opt}	$\frac{\mathcal{E}(W_{opt},\sigma_1,\sigma_2)}{W_{opt}}$
0.15	-	-	-
0.4	-	-	-
0.6	**0.8**	4251	690
0.8	0.4	4627	1082
1	0.4	5742	1625

$\rho = 1.775$

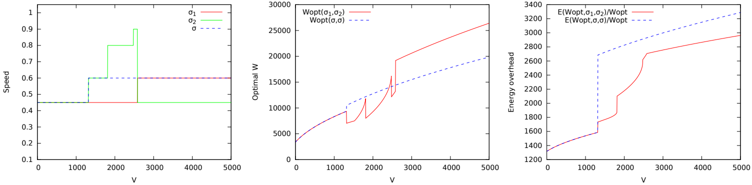
σ_1	**Best** σ_2	W_{opt}	$\frac{\mathcal{E}(W_{opt},\sigma_1,\sigma_2)}{W_{opt}}$
0.15	-	-	-
0.4	-	-	-
0.6	-	-	-
0.8	**0.4**	4627	1082
1	0.4	5742	1625

$\rho = 1.4$

Simulations - Impact of the parameters (1)



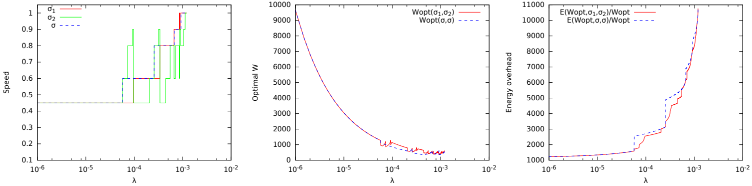
Opt. solution (speed pair, pattern size, and energy overhead) as a function of the checkpointing time c in Atlas/Crusoe configuration.



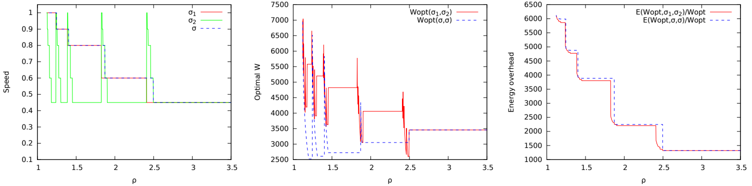
Opt. solution (speed pair, pattern size, and energy overhead) as a function of the verification time v in Atlas/Crusoe configuration.

Dotted line: one single speed; up to 35% improvement with two speeds

Simulations - Impact of the parameters (2)



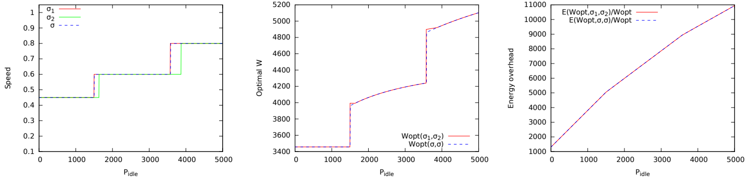
Opt. solution (speed pair, pattern size, and energy overhead) as a function of the error rate λ in Atlas/Crusoe configuration.



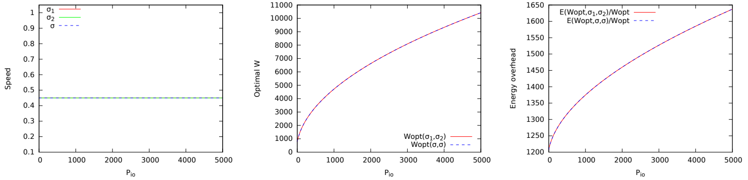
Opt. solution (speed pair, pattern size, and energy overhead) as a function of the performance bound ρ in Atlas/Crusoe configuration.

Two speeds: checkpoint less frequently and provide energy savings

Simulations - Impact of the parameters (3)



Optimal solution (speed pair, pattern size, and energy overhead) as a function of the idle power P_{idle} in Atlas/Crusoe configuration.



Optimal solution (speed pair, pattern size, and energy overhead) as a function of the I/O power P_{io} in Atlas/Crusoe configuration.

Increase of W and E with P_{idle} and P_{io}; P_{io} has no impact on speeds

Outline

1. Optimal checkpointing period: time vs. energy
   - Framework
   - Optimal period for execution time
   - Optimal period for energy
   - Experiments
2. A different re-execution speed can help
   - Model and optimization problem
   - Optimal pattern size and speeds
   - Simulations
   - Extensions: both fail-stop and silent errors
3. Summary and need for trade-offs

Extensions: With fail-stop errors

- f: proportion of fail-stop errors
- s: proportion of silent errors

Proposition (3)

With fail-stop and silent errors,

$$\frac{\mathcal{T}(W,\sigma_1,\sigma_2)}{W} = \cdots + \left(\frac{(f+s)}{\sigma_1\sigma_2} - \frac{f}{2\sigma_1^2}\right)\lambda W + O(\lambda^2 W). \quad (5)$$

$$\frac{\mathcal{E}(W,\sigma_1,\sigma_2)}{W} = \cdots + \left(\frac{(f+s)(\kappa\sigma_2^3 + P_{\text{idle}})}{\sigma_1\sigma_2} - \frac{f(\kappa\sigma_1^3 + P_{\text{idle}})}{2\sigma_1^2}\right)\lambda W + O(\lambda^2 W) \quad (6)$$

Limit of the first-order approximation

For BiCrit, the first-order approximation leads to a solution iff

$$\left(2\left(1+\frac{s}{f}\right)\right)^{-1/2} < \frac{\sigma_2}{\sigma_1} < 2\left(1+\frac{s}{f}\right)$$

Use second-order approximation? Open problem in the general case!

Interesting case

Theorem

When considering *only fail-stop errors* *with rate λ, the optimal pattern size W to minimize the time overhead $\frac{\mathcal{T}(W,\sigma,2\sigma)}{W}$ is*

$$W_{\text{opt}} = \sqrt[3]{\frac{12C}{\lambda^2}}\sigma$$

- Young/Daly's formula: $W_{\text{opt}} = \sqrt{2C/\lambda}\sigma = O(\lambda^{-1/2})$
- Here: $W_{\text{opt}} = O(\lambda^{-2/3})$

Conclusion

- A different re-execution speed indeed helps saving energy while satisfying a performance constraint
- Silent errors: extension of Young/Daly formula → general closed-form solution to get optimal speed pair and optimal checkpointing period (first-order)
- Extensive simulations: up to 35% energy savings, any speed pair can be optimal

- BiCrit still open for general case with both silent and fail-stop errors
- Interesting case with fail-stop errors and double re-execution speed: $O(\lambda^{-2/3})$ vs $O(\lambda^{-1/2})$
- New methods needed to capture the general case

Outline

1. Optimal checkpointing period: time vs. energy
   - Framework
   - Optimal period for execution time
   - Optimal period for energy
   - Experiments

2. A different re-execution speed can help
   - Model and optimization problem
   - Optimal pattern size and speeds
   - Simulations
   - Extensions: both fail-stop and silent errors

3. Summary and need for trade-offs

Summary and need for trade-offs

- Two major challenges for Exascale systems:
  - Resilience: need to handle failures
  - Energy: need to reduce energy consumption
- The main objective is often performance, such as execution time, but other criteria must be accounted for
- Two scenarios where looking at energy consumption may impact the decisions that are taken with respect to resilience
  - Adopt a different checkpointing period to optimize energy consumption
  - Use a different re-execution speed after a failure
- Still a lot of challenges to address, and techniques to be developed for many kinds of high-performance applications, making trade-offs between performance, reliability, and energy consumption

Summary and need for trade-offs

- Two major challenges for Exascale systems:
  - Resilience: need to handle failures
  - Energy: need to reduce energy consumption
- The main objective is often performance, such as execution time, but other criteria must be accounted for
- Two scenarios where looking at energy consumption may impact the decisions that are taken with respect to resilience
  - Adopt a different checkpointing period to optimize energy consumption
  - Use a different re-execution speed after a failure
- Still a lot of challenges to address, and techniques to be developed for many kinds of high-performance applications, making trade-offs between performance, reliability, and energy consumption

Summary and need for trade-offs

- Two major challenges for Exascale systems:
  - Resilience: need to handle failures
  - Energy: need to reduce energy consumption
- The main objective is often performance, such as execution time, but other criteria must be accounted for
- Two scenarios where looking at energy consumption may impact the decisions that are taken with respect to resilience
  - Adopt a different checkpointing period to optimize energy consumption
  - Use a different re-execution speed after a failure
- Still a lot of challenges to address, and techniques to be developed for many kinds of high-performance applications, making trade-offs between performance, reliability, and energy consumption

Thanks...

- ... to my co-authors
  - Guillaume Aupy
  - Thomas Hérault
  - Jack Dongarra
  - Yves Robert
  - Aurélien Cavelan
  - Valentin Le Fèvre
  - Hongyang Sun

- ... and to HPPAC organizers for their kind invitation!