Fixed-work vs fixed-time checkpointing

Anne Benoit
Inria, ENS Lyon & Institut Universitaire de France & IDEaS, GaTech, Atlanta, USA

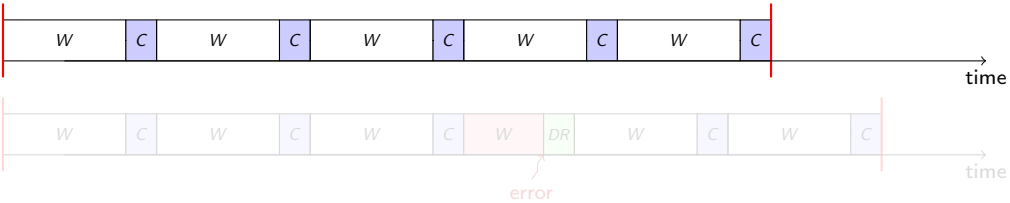
Quentin Barbut (ENSL), Thomas Herault (UTK, TN & Inria), Lucas Perotin (Vanderbilt University, Nashville, TN), Yves Robert, Frédéric Vivien (Inria, ENSL)

May 14, 2025 – JLESC Workshop

Problem: Fixed-work checkpointing

Fixed-work checkpointing: divide work into chunks, checkpoint each chunk

- Goal: Complete the execution as soon as possible (minimize expected time to execute a fixed amount of work)

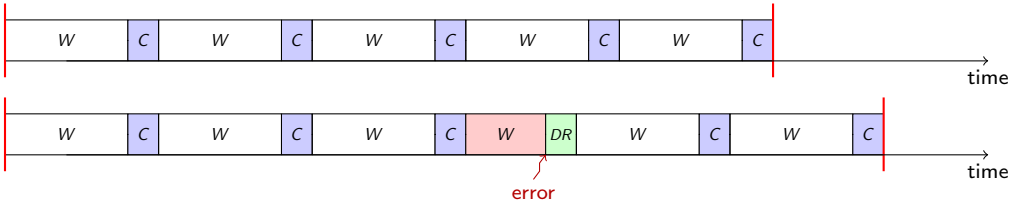


- If a fail-stop error strikes, restart the chunk
- Well-understood: periodic strategy, Young-Daly period $W_{YD} = \sqrt{2\mu C}$

Problem: Fixed-work checkpointing

Fixed-work checkpointing: divide work into chunks, checkpoint each chunk

- Goal: Complete the execution as soon as possible (minimize expected time to execute a fixed amount of work)

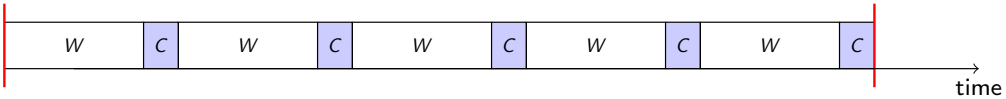


- If a fail-stop error strikes, restart the chunk
- Well-understood: periodic strategy, Young-Daly period $W_{YD} = \sqrt{2\mu C}$

Problem: Fixed-time checkpointing

Fixed-time checkpointing: work within a fixed time frame, e.g., the reservation for your job

- Goal: Complete as much work as possible (minimize expected time to execute a fixed amount of work)

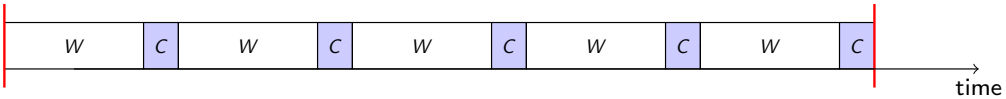


- If a fail-stop error strikes, what do we do next?

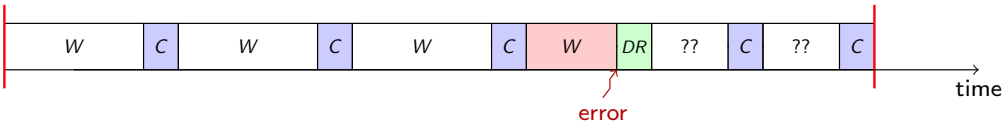
Problem: Fixed-time checkpointing

Fixed-time checkpointing: work within a fixed time frame, e.g., the reservation for your job

- Goal: Complete as much work as possible (minimize expected time to execute a fixed amount of work)



- If a fail-stop error strikes, what do we do next?

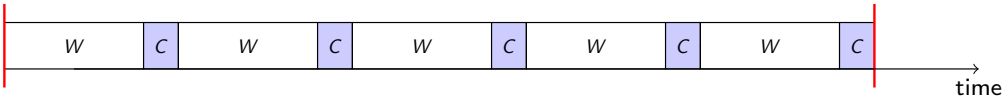


(1) Take two checkpoints as before, smaller chunks?

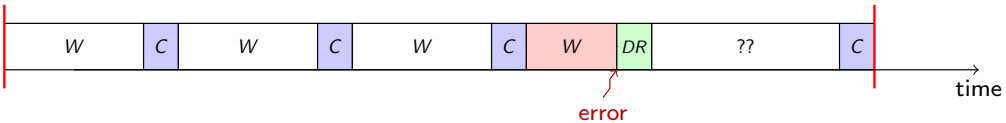
Problem: Fixed-time checkpointing

Fixed-time checkpointing: work within a fixed time frame, e.g., the reservation for your job

- Goal: Complete as much work as possible (minimize expected time to execute a fixed amount of work)



- If a fail-stop error strikes, what do we do next?

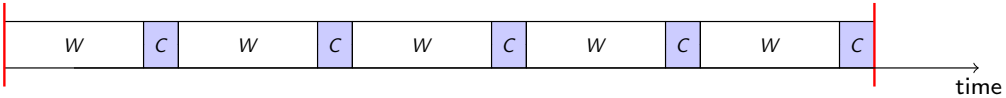


(2) Do all the work then checkpoint?

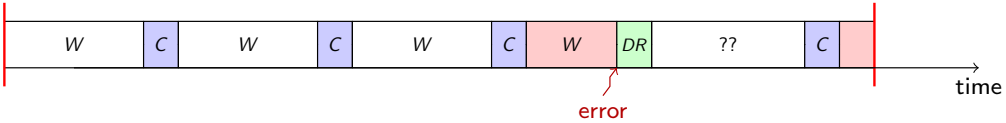
Problem: Fixed-time checkpointing

Fixed-time checkpointing: work within a fixed time frame, e.g., the reservation for your job

- Goal: Complete as much work as possible (minimize expected time to execute a fixed amount of work)



- If a fail-stop error strikes, what do we do next?



(3) Do some work then checkpoint?

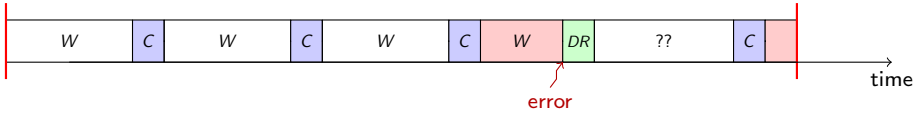
Problem: Fixed-time checkpointing

- Checkpointing strategies for a large-scale parallel application
- Nodes are subject to failures (Exponential distribution of parameter λ, $\mu = \frac{1}{\lambda}$)
- The application executes for a fixed duration, i.e., the length T of its reservation
- Checkpoint cost C, recovery cost R, and downtime D
- The **last checkpoint** within the reservation plays a particular role:
  $\Rightarrow$ all the work executed after that checkpoint will be lost
  $\Rightarrow$ take it close to, or exactly at the very end of the reservation

- Important question, but not yet addressed
- Dual of the fixed-work checkpointing classical problem

Problem: Fixed-time checkpointing

- Checkpointing strategies for a large-scale parallel application
- Nodes are subject to failures (Exponential distribution of parameter λ, $\mu = \frac{1}{\lambda}$)
- The application executes for a fixed duration, i.e., the length T of its reservation
- Checkpoint cost C, recovery cost R, and downtime D
- The **last checkpoint** within the reservation plays a particular role:
  $\Rightarrow$ all the work executed after that checkpoint will be lost
  $\Rightarrow$ take it close to, or exactly at the very end of the reservation



- Important question, but not yet addressed
- Dual of the fixed-work checkpointing classical problem

Why a fixed-length reservation?

- Job executes and checkpoints at the end of its reservation of length T
- Job is killed after T seconds
- Typical scenario:
  - long running HPC application
  - split into multiple smaller reservations

- **No failures:** Checkpoint at time $T - C$?
  - C not always constant, but may have a bounded support $[C_{min}, C_{max}]$
  - Checkpointing at time $T - C_{max}$ is pessimistic and may lead to wasting execution, but no risk taken...
  - Full study in [FTXS'23: When to checkpoint at the end of a fixed-length reservation?]
  - Problem solved if it is possible to checkpoint an any instant, for various probability laws of $C \Rightarrow$ later than at time $T - C_{max}$!

With failures: Dynamic checkpointing strategy

- Initially, the time left is $t_{left} = T$
- The strategy decides how many checkpoints will be taken, and at which instants, if there is no failure during the whole execution
- Let $t_{end}(i) \leq T$ be the completion time of checkpoint number i, with $1 \leq i \leq k$
- If there is no failure up to time $t_{end}(k)$ (when the last checkpoint completes), then the work achieved by the strategy is $W = t_{end}(k) - kC$.
- If a (first) failure strikes at time $t \leq T$, let $\ell \leq k$ be the number of the last checkpoint that completed before the failure:
  - The work done after time $t_{end}(\ell)$ and up to time t is lost
  - The work achieved up to the failure is $W = t_{end}(\ell) - \ell C$
- Now after the failure at time t, there is a downtime and a recovery; hence, the time left is $t_{left} = (T - t) - D - R$
- If $t_{left} \geq C$, call recursively the strategy and add the work done then to W
- The objective is to maximize the expected amount of work $\mathbb{E}(T) = W$

With failures: Dynamic checkpointing strategy

- Initially, the time left is $t_{left} = T$

Dynamic approach

• For any value t_{left}, the strategy decides how many checkpoints will be taken, and at which instants, **but it is applied until the end only if no failure strikes during the execution.**

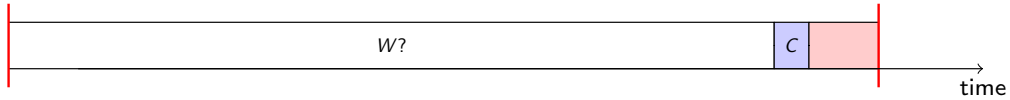
• The strategy will not be the same for different values of t_{left}: the distances between consecutive checkpoints are recomputed after each failure

Nice but challenging little scheduling problem 

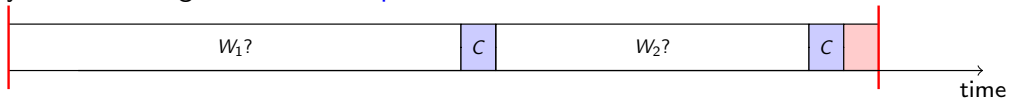
- If $t_{left} \geq C$, call recursively the strategy and add the work done then to W
- The objective is to maximize the expected amount of work $\mathbb{E}(T) = W$

Difficulty

- No closed-form formula even for a single checkpoint at the end
- If you decide to go for a single checkpoint: do not always place it at the end of the reservation, even if C is constant!



- If you decide to go for two checkpoints: almost never use two same-size chunks!



Single checkpoint at the end of the reservation

Fixed-time checkpointing

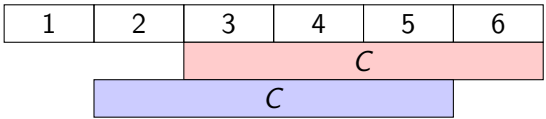
$$\mathbb{E}^{end}(T) = e^{-\lambda T}(T - C) + (1 - e^{-\lambda T}) \int_{t=0}^{T-D-R-C} \lambda t \frac{e^{-\lambda t}}{1-e^{-\lambda T}} \mathbb{E}_R^{end}(T\text{- t - }D)dt$$
$$\mathbb{E}_R^{end}(t_{left}) = e^{-\lambda T}(t_{left} - R - C) + \int_{t=0}^{t_{left}-D-R-C} \lambda t \frac{e^{-\lambda t}}{1-e^{-\lambda T}} \mathbb{E}_R^{end}(t_{left} \text{ - t - } D)dt$$

 No closed-form expression 

Fixed-work checkpointing

$$\mathbb{E}(W) = e^{-\lambda(W+C)}(W + C) + (1 - e^{-\lambda(W+C)})(\mathbb{E}(T_{lost}(W + C)) + \mathbb{E}(T_{rec}) + \mathbb{E}(W))$$
$$\Rightarrow \mathbb{E}(W) = \left(\tfrac{1}{\lambda} + D\right) e^{\lambda R}(e^{\lambda(W+C)} - 1)$$

Single checkpoint is **not always** at the end of the reservation



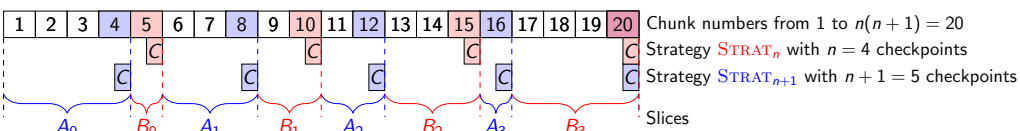
Short reservation, $T = 6$, with $D = 0$, $C = R = 4$
Strategy $\textsc{Strat}_1$: Checkpoint at the end
Strategy $\textsc{Strat}_2$: Checkpoint before the end

Expected gain $\textsc{Gain}$ of $\textsc{Strat}_1$ over $\textsc{Strat}_2$

- Contrived example: no time left for any work after a failure
- No failure: $\textsc{Gain} = \mathbb{P}_{succ}(6) \times 1$:
- First failure strikes after $t = 5$: $\textsc{Gain} = \mathbb{P}_{succ}(5)\mathbb{P}_{fail}(1) \times (-1)$
- First failure strikes before $t = 5$: $\textsc{Gain} = 0$
- $\textsc{Gain} = e^{-6\lambda} - e^{-5\lambda}(1 - e^{-\lambda}) = 2e^{-6\lambda} - e^{-5\lambda} = e^{\ln(2)-6\lambda} - e^{-5\lambda}$
- If $\lambda > \ln(2)$, better to checkpoint before the end of the reservation!

Dynamic threshold-based strategy

- Natural conjecture:
  - Easy to prove that expected work increases with T
  - Assume same-size segments and last checkpoint at the end
  - Conjecture that longer reservations initially require more segments (hence more checkpoints)
- (Sub-optimal) strategy based on this conjecture:
  - Construct sequences of thresholds T_n such that there are exactly n checkpoints for $T_n \leq T \leq T_{n+1}$, use number of checkpoints corresponding to t_{left}



$$\text{GAIN}(T, n+1) = \mathbb{E}(T, n+1) - \mathbb{E}(T, n)$$

Numerically, there is a single solution to $\text{GAIN}(T, n+1) = 0$ ☺

Strategies

- YoungDaly is the baseline reference, which takes checkpoints periodically following the Young/Daly formula W_{YD} (with mandatory checkpoint at the end)
- NumericalOptimum uses a numerical approximation of the thresholds T_n
- FirstOrder uses a first-order approximation of the thresholds:
  $$T_{n+1} \approx \sqrt{2n(n+1)\mu C}$$
  For $n = 1$, $T_2 \approx \sqrt{4\mu C}$ while $W_{YD} = \sqrt{2\mu C}$ (single checkpoint for longer reservations)

- **Optimal strategy**: Quantum-based DynamicProgramming algorithm
  - General approach using small time quanta
  - Use dynamic programming to compute the best checkpointing strategy
  - The smaller the quanta, the more accurate the results ...

    ... but the more costly the algorithm

Dynamic programming

- $\mathbb{E}(n, k, \delta)$ optimal work (in expectation)
  - during n quanta
  - when taking exactly k checkpoints initially
  - $\delta = 0$ if there is no initial recovery with the work, and $\delta = 1$ otherwise
- $k_{max} = \left\lfloor \frac{T^*}{C^*} \right\rfloor$ maximum number of checkpoints (if we checkpoint all the time!)
- Compute $\mathbb{E}_{opt}(T^*) = \max_{1 \leq k \leq k_{max}} \mathbb{E}(T^*, k, 0)$

Complexity: $O((T^*)^2 k_{max}^2)$

Quantum duration	u
Number of quanta	$T^* = \frac{T}{u}$
Checkpoint time in numbers of quanta	$C^* = \frac{C}{u}$

Solution

With one checkpoint

$$\begin{aligned}\mathbb{E}(n,1,\delta) &= \max_{\delta R^*+C^*+1\le i\le n} \mathbb{P}^*_{succ}(i)(i - C^* - \delta R^*) \\ &\quad + \sum_{f=1}^{i-D^*-R^*-C^*} p_f^* \mathbb{E}(n-f-D^*,1,1)\end{aligned}$$

With k checkpoints

$$\begin{aligned}\mathbb{E}(n,k,\delta) = \max_{\delta R^*+C^*+1\le i\le n-(k-1)C^*} \Big(&\mathbb{P}^*_{succ}(i)\big(i-C^*-\delta R^* + \mathbb{E}(n-i,k-1,0)\big) \\ &+ \sum_{f=1}^{i} p_f^* \max_{1\le m\le k} \mathbb{E}(n-f-D^*,m,1)\Big)\end{aligned}$$

Initialization

$$\mathbb{E}(n,k,0) = 0 \quad \text{for } n \le kC^* \text{ (where possibly } n \le 0)$$
$$\mathbb{E}(n,k,1) = 0 \quad \text{for } n \le R^* + kC^* \text{ (where possibly } n \le 0)$$

Simulation framework

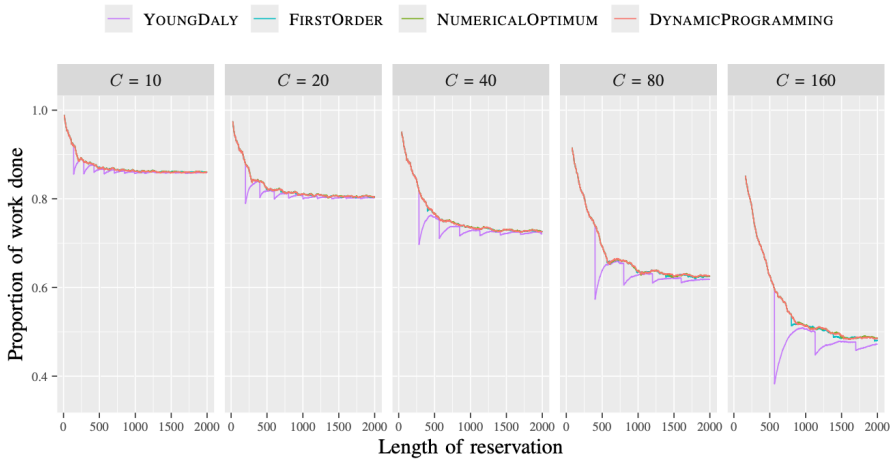
Parameters

- $C \in \{10, 20, 40, 80, 160\}$, and $R = C$
- $D \in \{0, 5\}$
- $\lambda \in \{0.01, 0.001, 0.0001\}$
- $T \in [C, 2000\}$
- Quantum size $u = 1$ for DynamicProgramming (after experimenting with other sizes)

(arbitrary time-unit – change granularity of the scenarios)

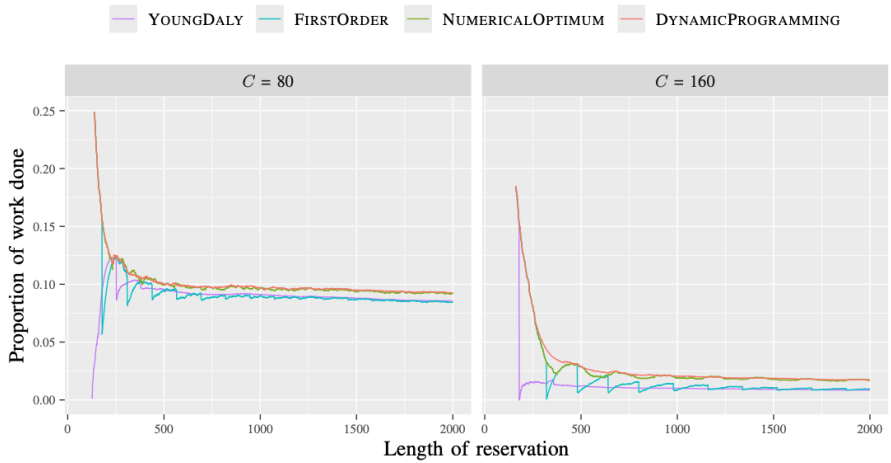
Report proportion of work achieved $= \frac{\text{amount of work achieved}}{T-C}$ (the higher the better)

Results



Proportion of work when $\lambda = 0.001$ and $D = 0$

Results



With more failures: Proportion of work when $\lambda = 0.01$, $D = 0$, and $C \in \{80, 160\}$

Summary

- YoungDaly achieves significantly low performance when only a handful of checkpoints are required, but performs well for long reservations
- NumericalOptimum $\geq$ FirstOrder $\geq$ YoungDaly
- DynamicProgramming only slightly better than NumericalOptimum, but much more costly – Dynamic threshold strategy very easy to use and close to optimal ☺
- Big differences mainly for high failure rates and short reservations

Details in [FTXS'2024: Checkpointing strategies for a fixed-length execution]

Conclusion

- *Fixed-time checkpointing* much more difficult than *Fixed-work checkpointing*
- Optimal solution seems out of reach ☹
- Three good news:
  - Optimal discretized solution as a guarantee of quality
  - Young/Daly performs well for reservations lasting at least, say, a dozen Young/Daly periods $W_{YD} = \sqrt{2\mu C}$
  - Dynamic threshold strategy very easy to use and performs remarkably well
- **Future work:**
  - Extend results to a stochastic framework, where both checkpoint durations and application progress rate are no longer fully deterministic
  - Apply to sparse linear algebra problems