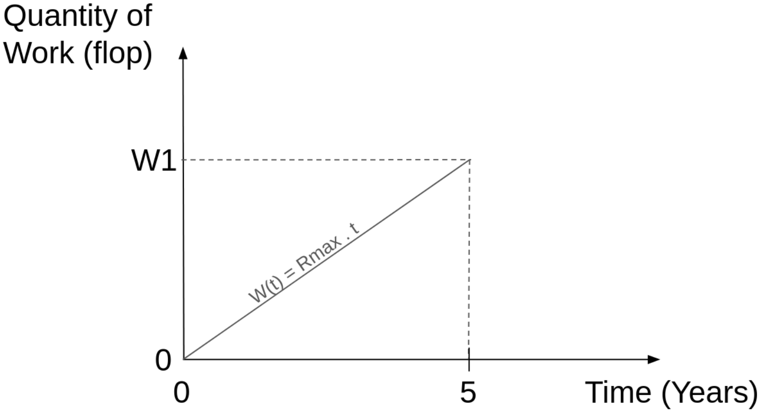
Improving Supercomputing Usage with Aging Awareness

Guillaume Pallez, Inria

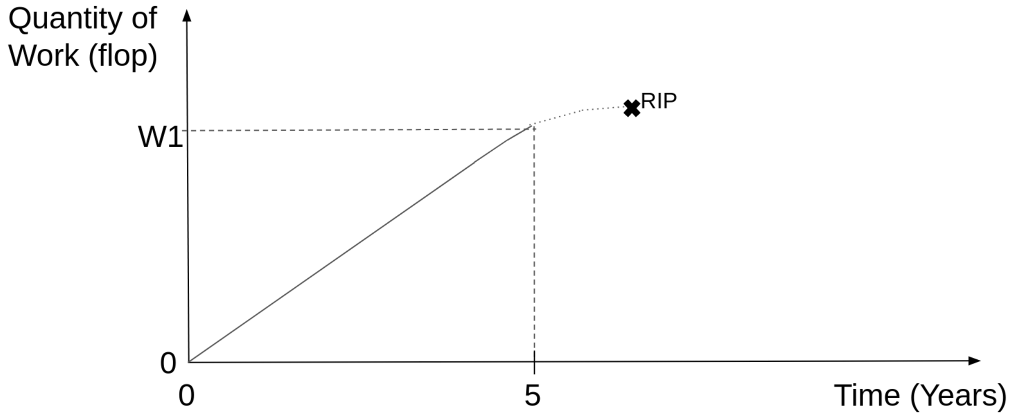
Scheduling in Fréjus, March 2026

Work with R. Boezennec, F. Dufossé, A. Tremodeux

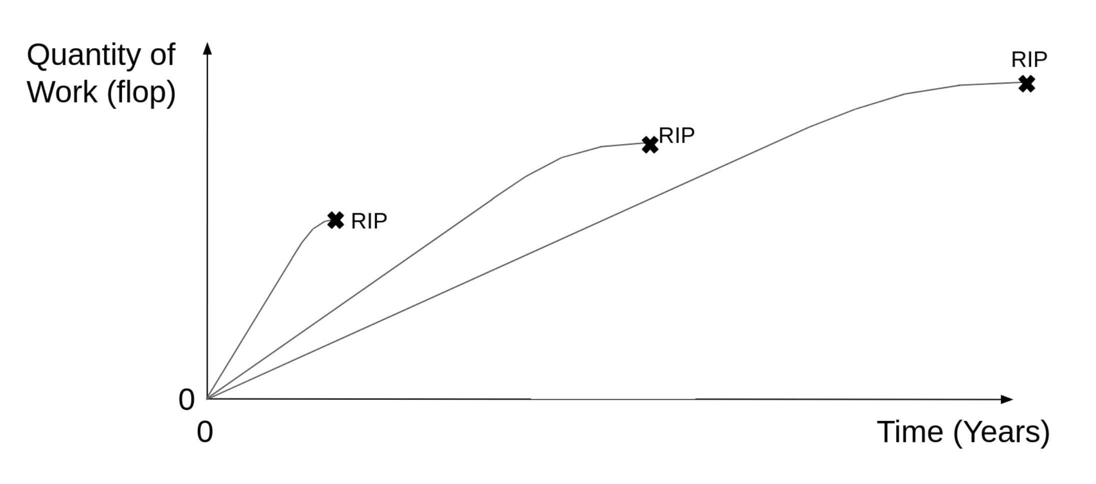
How are supercomputers currently used?



What if we extend the lifetime?



What if we use the machine differently?



Goal: Demonstrate the usefulness of considering aging in resource management

Limit/Challenge: Having accurate aging models

Hard because:

- Lack of accurate Models (i.e. parametrized Equation)
  - currently based on chemical reactions, not on actual aging analysis
- Instantiation of the models also hard
  - too many factors
- Model Validation is challenging

Goal: Demonstrate the usefulness of considering aging in resource management

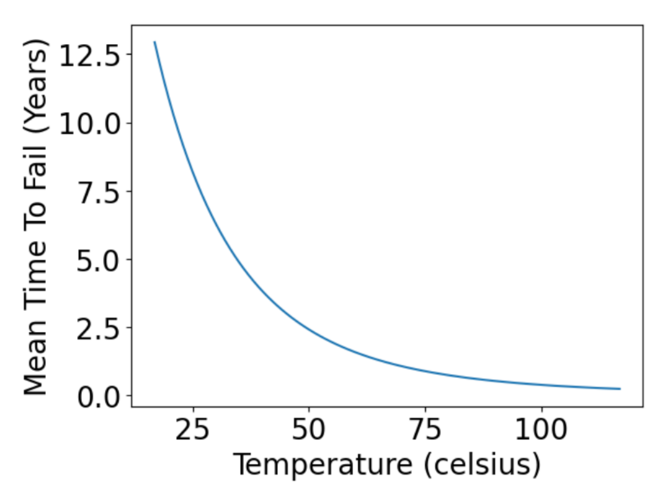
Limit/Challenge: Having accurate aging models

Hard because:

- Lack of accurate Models (i.e. parametrized Equation)
  - currently based on chemical reactions, not on actual aging analysis
- Instantiation of the models also hard
  - too many factors
- Model Validation is challenging

Subgoal: to convince on the importance of research on HPC aging

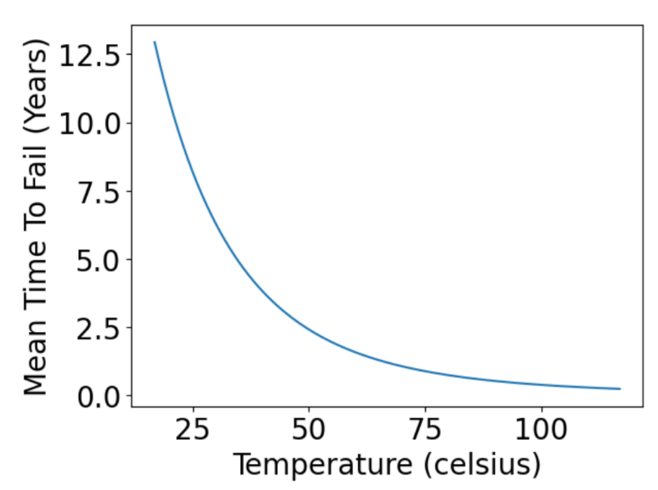
How do compute nodes age?



Temperature depends on

- Computing frequency
- Cooling parameter

How do compute nodes age?

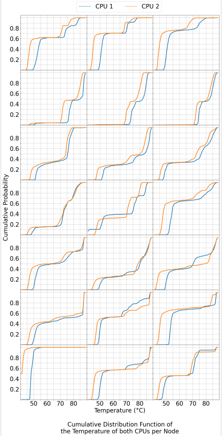


Temperature depends on

- Computing frequency
- Cooling parameter

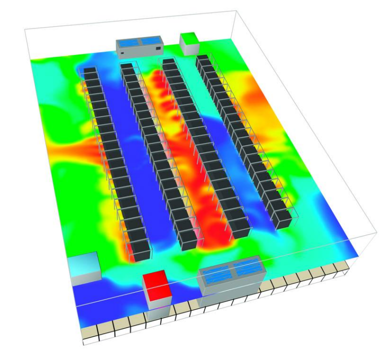
$$MTTF(T) = c \cdot \left(\left[\ln\left(\frac{A}{1+2e^{\frac{B}{k_b T}}} \right) - \ln\left(\frac{A}{1+2e^{\frac{B}{k_b T}}} - C \right) \right] \cdot \frac{T}{e^{\frac{-D}{k_b T}}} \right)^{\frac{1}{\beta}}$$

Not all nodes are created equal



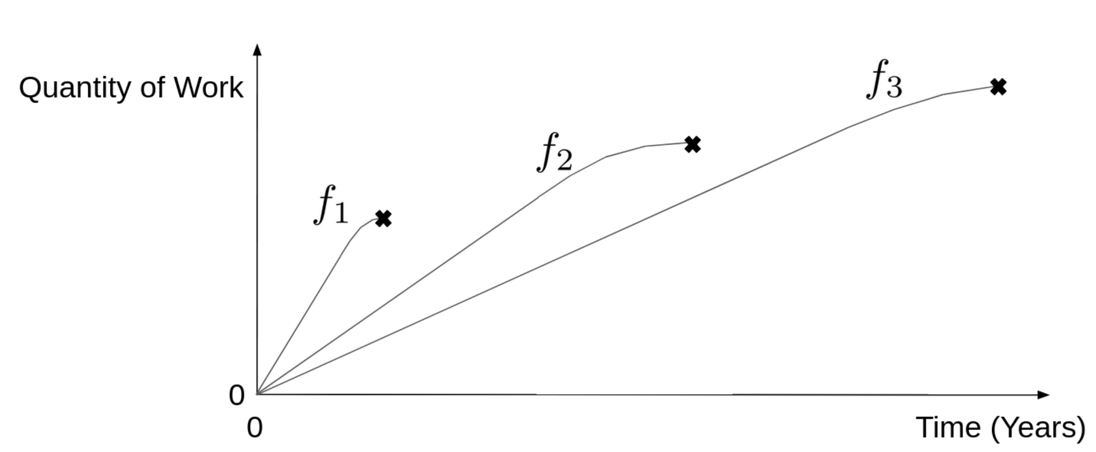
21 nodes cluster (each with 2 AMD Epyc 9224 CPUs) during 10 days

Source: white paper, hal-05312072.



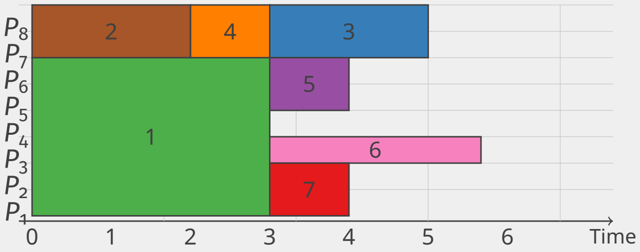
Source: CRAC unit sizing: Dos and don'ts, Chube 2012.

Approach: modifying the node frequencies (DVFS-like)



Scheduling problem

- Placement: Where to schedule jobs?
- Frequency: At which frequency to run them (identical for all job nodes)?



Challenge: how to measure a “good” schedule?

- **Idea: Maximize work over lifetime?** Not really
- Maximize work until X% of the machines are dead?
  - Optimal behavior seems to have the same problem (stop using some nodes when they are about to die).

Challenge: how to measure a “good” schedule?

- **Idea: Maximize work over lifetime?** Not really
- Maximize work until X% of the machines are dead?
  - Optimal behavior seems to have the same problem (stop using some nodes when they are about to die).

The objective function needs to allow “sane/continuous” behavior as well as measuring the trade-off.

Challenge: how to measure a “good” schedule?

- **Idea: Maximize work over lifetime?** Not really
- Maximize work until X% of the machines are dead?
  - Optimal behavior seems to have the same problem (stop using some nodes when they are about to die).

The objective function needs to allow “sane/continuous” behavior as well as measuring the trade-off.

From event 1 to event 2:
- Total work done
- Degradation
- Length

Challenge: how to measure a "good" schedule?

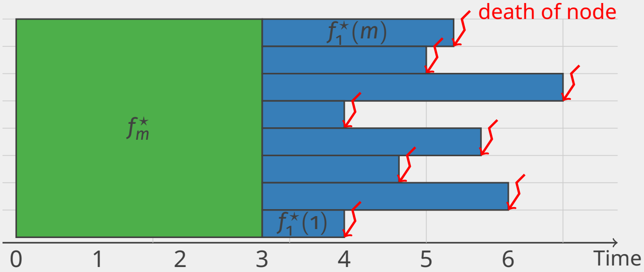
- **Idea: Maximize work over lifetime?** Not really
- Maximize work until X% of the machines are dead?
  - Optimal behavior seems to have the same problem (stop using some nodes when they are about to die).

The objective function needs to allow "sane/continuous" behavior as well as measuring the trade-off.

From event 1 to event 2:
- Total work done
- Degradation
- Length

⟶

→ "average" work per unit of degradation
→ mean system frequency

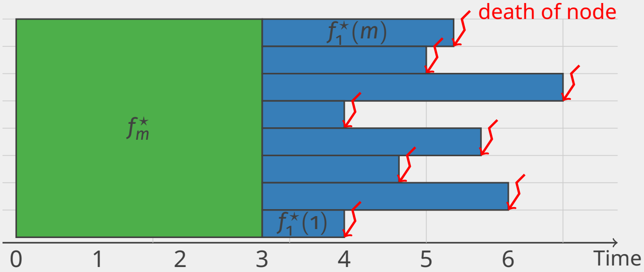


Goal:

- Do large job
- Maximize total work

Theorem

- $f_m^\star$ *does not depend on size of large job or state of degradation of nodes.*
- $f_m^\star \in [\min_i f_1^\star(i), \max_i f_1^\star(i)]$



Goal:

- Do large job
- Maximize total work

Theorem

- *$f_m^\star$ does not depend on size of large job or state of degradation of nodes.*
- $f_m^\star \in [\min_i f_1^\star(i), \max_i f_1^\star(i)]$

And.. that's it. A little bit disappointing.

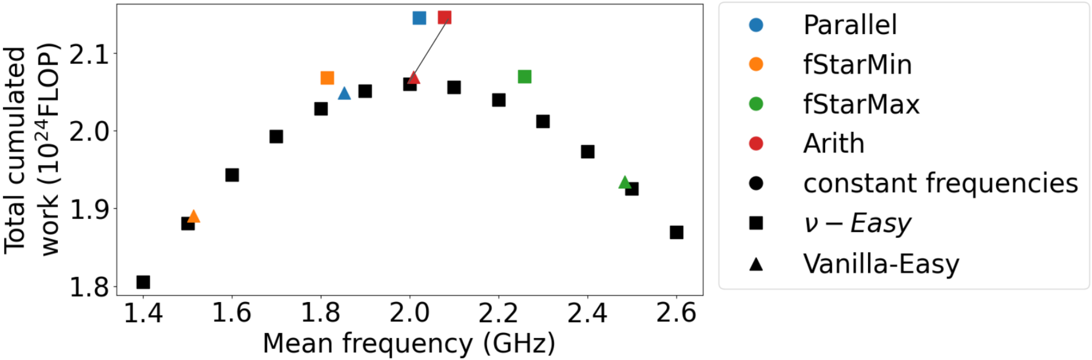
Algorithms

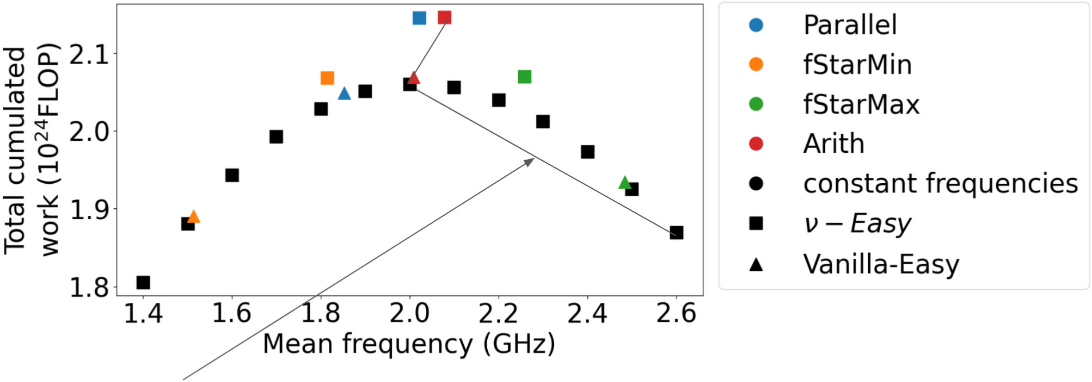
Placement: Easy-BF with node ordered:

- Random-order (*Vanilla*)
- By cooling parameter ($\nu - EASY$)

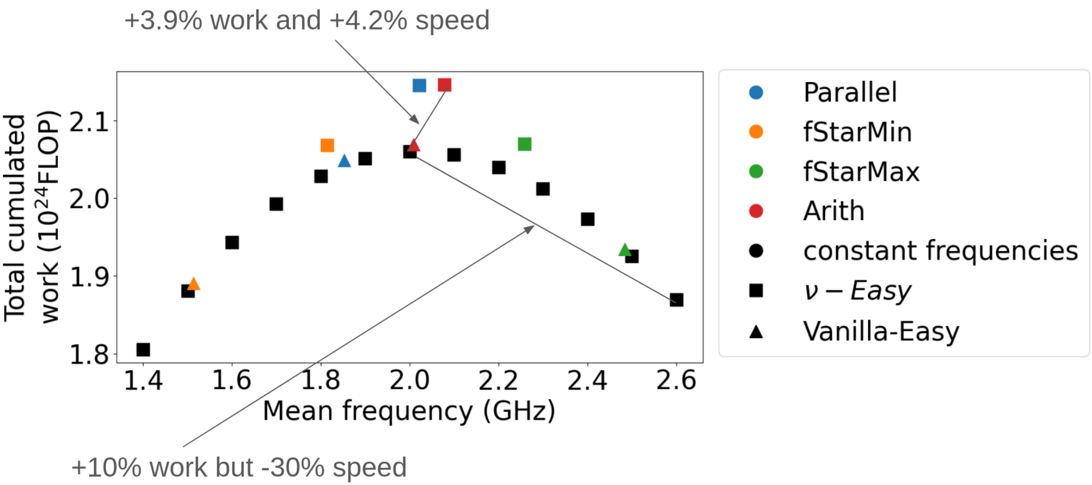
Frequency selection:

- Parallel: $f_m^\star$
- Arith: $1/m \sum_i f_1^\star(i)$
- $\cdots$


Total cumulated work (10^{24}FLOP)
2.1
2.0
1.9
1.8
1.4
1.6
1.8
2.0
2.2
2.4
2.6
Mean frequency (GHz)
Parallel
fStarMin
fStarMax
Arith
constant frequencies
$\nu - Easy$
Vanilla-Easy



+10% work but -30% speed


+3.9% work and +4.2% speed
Total cumulated work (10^{24}FLOP)
2.1
2.0
1.9
1.8
1.4
1.6
1.8
2.0
2.2
2.4
2.6
Mean frequency (GHz)
+10% work but -30% speed
Parallel
fStarMin
fStarMax
Arith
constant frequencies
$\nu - Easy$
Vanilla-Easy

Importance of Model Unreliability (I)

$$MTTF(R,f) = R \cdot e^{b - a \cdot f^3}$$

$$a = a_0 \cdot (1 + \varepsilon) \qquad \varepsilon \approx \mathcal{N}(0, \sigma)$$

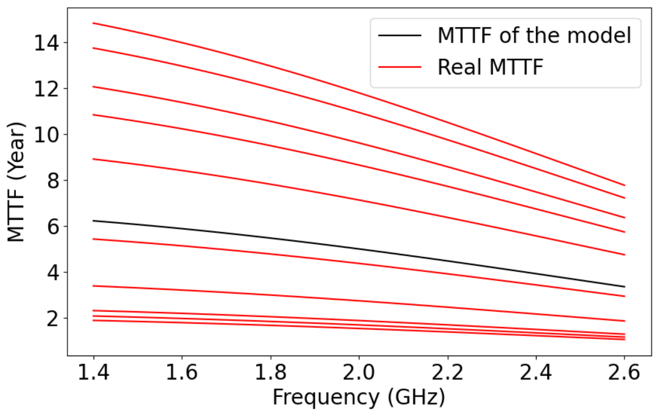
$$b = b_0 \cdot (1 + \varepsilon) \qquad \varepsilon \approx \mathcal{N}(0, \sigma)$$

Importance of Model Unreliability (I)

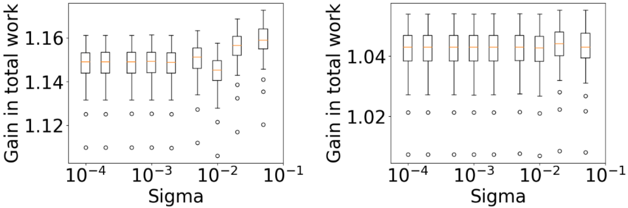
$$MTTF(R,f) = R \cdot e^{b-a\cdot f^3}$$

$$a = a_0 \cdot (1+\varepsilon) \qquad \varepsilon \approx \mathcal{N}(0,\sigma)$$
$$b = b_0 \cdot (1+\varepsilon) \qquad \varepsilon \approx \mathcal{N}(0,\sigma)$$

$$\sigma = .05$$



Importance of Model Unreliability (II)



(a) Compared to 2.6 GHz **(b) Compared to 2 GHz**

Figure 5: Quantity of work performed by `Parallel` compared to 2 and 2.6 GHz, as a function to the level of noise.

Concluding remarks

Some observations:

- If node reordering is not available, it is not necessary to do advanced heuristics.
- Simple heuristic (Arith) quite efficient (no need for Parallel)
- Increasing difference in node cooling reinforces the need for aging-aware strategies
- Heuristics quite robust to model inaccuracy

Concluding remarks

Some observations:

- If node reordering is not available, it is not necessary to do advanced heuristics.
- Simple heuristic (Arith) quite efficient (no need for Parallel)
- Increasing difference in node cooling reinforces the need for aging-aware strategies
- Heuristics quite robust to model inaccuracy

Next:

- need more practical work on aging models
- ways to measure aging
- but also: variable behavior depending on energy cost..

Extending the life of supercomputers

Why?

Increasing the life of a machine for:

- Economical motivations (really?)
- Ecological motivations (likely)
- Geopolitical motivations (clearly!)

“Aurora was first announced in 2015 and to be finished in 2018. (…) Later, in 2017, Intel announced that Aurora would be delayed to 2021 but scaled up to 1 exaFLOP. (…) The system was fully installed on June 22, 2023.” (–wikipedia)

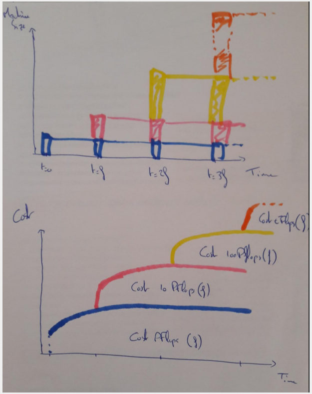
Ex: Economical motivations? (I)

As time evolves, technology also allows to do more Flops. How can we compare the cost of various renewal strategies?

Ex: Economical motivations? (I)

As time evolves, technology also allows to do more Flops. How can we compare the cost of various renewal strategies?

$$\texttt{CostFlop}(f) = \sum_{i=0}^{\infty} \left(\texttt{CapEx}(f \cdot i) + \sum_{t=0}^{f-1} \texttt{OpEx}(f \cdot i, f \cdot i + t) \right)$$



$\texttt{CostFlop}(f)$ measures the relative cost for each *color* (i.e. fixed volume of Flop), and should be used to compare various renewal strategies.

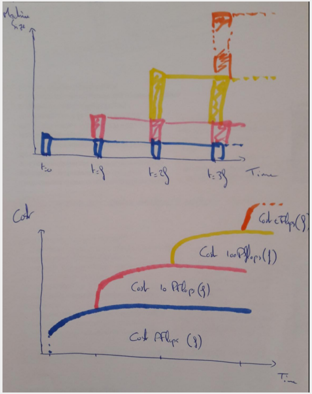
Ex: Economical motivations? (I)

As time evolves, technology also allows to do more Flops. How can we compare the cost of various renewal strategies?

$$\texttt{CostFlop}(f) = \sum_{i=0}^{\infty}\left(\texttt{CapEx}(f\cdot i) + \sum_{t=0}^{f-1}\texttt{OpEx}(f\cdot i, f\cdot i + t)\right)$$

Exponential model:

$$\begin{aligned}
\texttt{CapEx}(n+1) &= \texttt{CapEx}(n)\cdot C_{\texttt{capex}} & C_{\texttt{capex}} &< 1\\
\texttt{OpEx}(n+1, n+1) &= \texttt{OpEx}(n,n)\cdot C_{\texttt{opex}} & C_{\texttt{opex}} &< 1\\
\texttt{OpEx}(n, n+1) &= \texttt{OpEx}(n,n)\cdot \texttt{deg} & \texttt{deg} &> 1
\end{aligned}$$

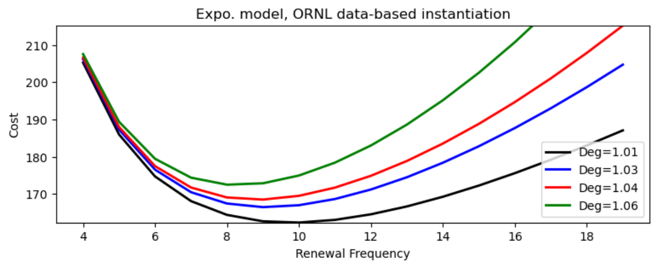


$\texttt{CostFlop}(f)$ measures the relative cost for each *color* (i.e. fixed volume of Flop), and should be used to compare various renewal strategies.

Ex: Economical motivations? (Ex: ORNL)

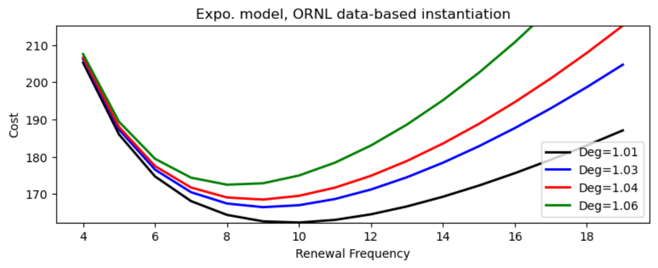
	Name	Price	Cons.	PFlops
2012	Titan	$97M	8.2MW	17.6
2018	Summit	$ 325M	13MW	200
2024	Frontier	$ 600M	25MW	1300

Eq 200 PFlops	
$1103M	93MW
$ 325M	12MW
$100M	4MW



ORNL approximation (for illustration purpose):

- $\frac{\mathrm{OpEx}(0,0)}{\mathrm{CapEx}(0)} = \frac{13}{325} = 0.04 \quad \left(\approx \frac{25}{600}\right)$
- $C^6_{capex} = 1/3$
- $C^6_{opex} = 1/3$
- $\deg^6 = 1.08$ (resp. 1.20, 1.26, 1.42)



However:
Renewing every 5y,
not just about cost.
→ **Idea is to do more.**

How to evaluate the gain/economical cost **to do more compute** ?

Inria challenge on aging

Goal:

- Produce recommendations on the service life of machines;
- Produce recommendations and tools for managing HPC machines in order to extend their lifespan. ;
- Knowing how to measure a machine’s *actual age*.

Spoiler: Panel@IPDPS’26 about extending the lifespan of HPC:

- Judith Hill (LLNL)
- Al Geist (ORNL)
- Fumishomi Yoshi (Riken)