Algorithmes d'ordonnancement et schémas de résilience pour les pannes et les erreurs silencieuses

---

Aurélien Cavelan <aurelien.cavelan@unibas.ch>

ENS de Lyon et Inria, France

January 31, 2017

Top500 List

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
2	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P , NUDT National Super Computer Center in Guangzhou China	3,120,000	33,862.7	54,902.4	17,808
3	**Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 , Cray Inc. Swiss National Supercomputing Centre (CSCS) Switzerland	361,760	19,590.0	25,326.3	2,272
4	**Gyoukou** - ZettaScaler-2.2 HPC system, Xeon D-1571 16C 1.3GHz, Infiniband EDR, PEZY-SC2 700Mhz , ExaScaler Japan Agency for Marine-Earth Science and Technology Japan	19,860,000	19,135.8	28,192.0	1,350
5	**Titan** - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x , Cray Inc. DOE/SC/Oak Ridge National Laboratory United States	560,640	17,590.0	27,112.5	8,209
6	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom , IBM DOE/NNSA/LLNL United States	1,572,864	17,173.2	20,132.7	7,890
7	**Trinity** - Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect ,	979,968	14,137.3	43,902.6	3,844

Outline

I. Failures

II. Silent Errors

I. Failures

Mean Time Between Failures (MTBF)

Consider one processor (e.g. in your laptop):

- **MTBF = 100 years**
- (Almost) no failures in practice ...

Theorem.
MTBF decreases linearly with the number of processors.

- 36500 processors
  - **MTBF = 1 day**
  - A failure every day on average!

Large simulations can execute for weeks at a time.

A petascale computer

- $400m^2$
- 17.59 PetaFlops
- 693.6 TiB of RAM

Titan has 37376 processors and GPUs and ≈ 1 day MTBF.

Fail-Stop Errors

Failures proportional to number of processors

- 2013: **Preprodudcion** Blue Waters requires repairs ≈ 4 hours
- 2014: Titan ($37,376$ processors) loses a node every ≈ 1.5 days
- 2015: Blue Waters ($26,868$ processors) loses ≈ 2 nodes per day

Characteristics

- Component failure (node, network, power, ...)
- Application fails and data is lost

An Inconvenient Truth

Top ranked supercomputers in the US (June 2017)

Rank	Name	Laboratory	Technology	Processors	PFlops/s	MTBF
5	Titan	ORNL	Cray XK7	37,376	17.59	≈ 1 day
6	Sequoia	LLNL	BG/Q	98,304	17.17	≈ 1 day
8	Cori	LBNL	Cray XC40	11,308	14.01	≈ 1 day
11	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 failures:

- Make applications more fault tolerant!
- **Design better resilience techniques!**

An Inconvenient Truth

Top ranked supercomputers in the US (June 2017)

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

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

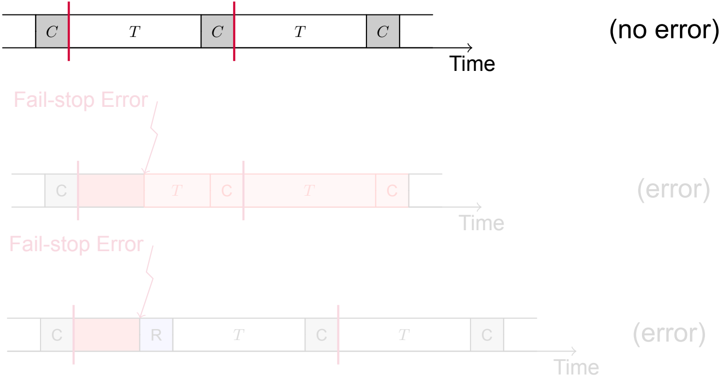
- Larger processors count: millions of processors
- **MTBF is expected to drop dramatically**
- Down to **the hour** or even worse

Coping with failures:

- Make applications more fault tolerant!
- **Design better resilience techniques!**

Coping with Fail-Stop Errors

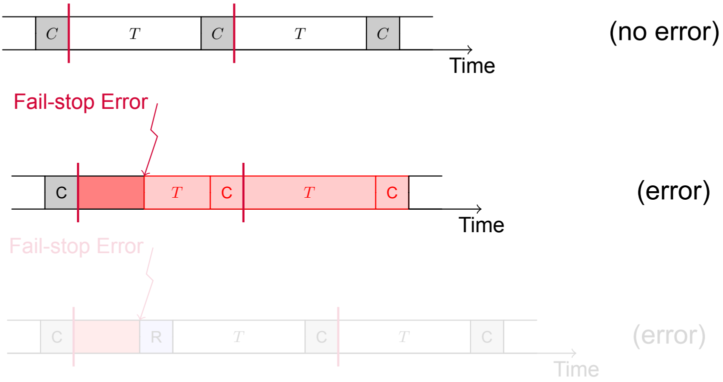
Periodic checkpoint, rollback, and recovery:



- 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

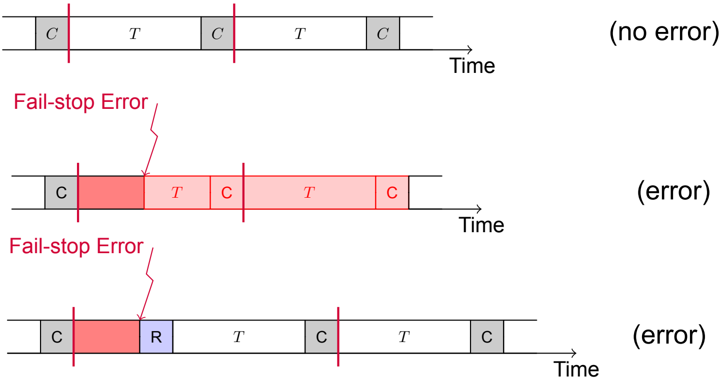
Periodic checkpoint, rollback, and recovery:



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

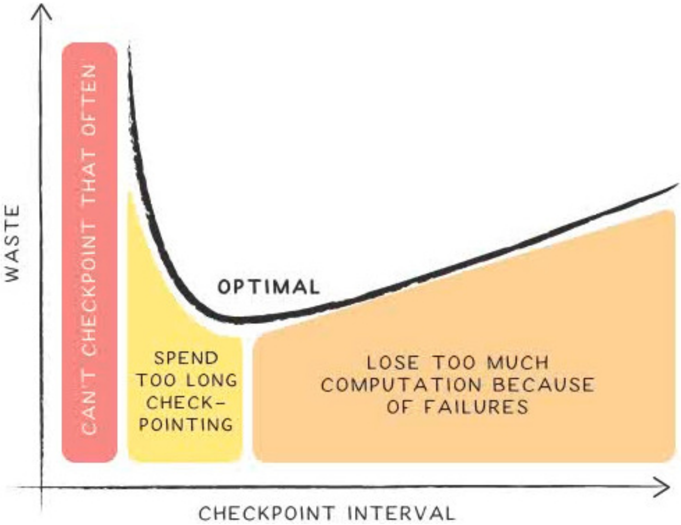
Coping with Fail-Stop Errors

Periodic checkpoint, rollback, and recovery:



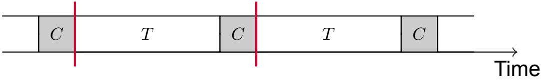
- Coordinated checkpointing (the platform is a giant macro-processor)
- Assume instantaneous interruption and detection.
- Rollback to last checkpoint and re-execute.

Optimal Checkpoint Interval



Minimize Expected Execution Time

- T: Pattern length
- C: Checkpoint time
- R: Recovery time



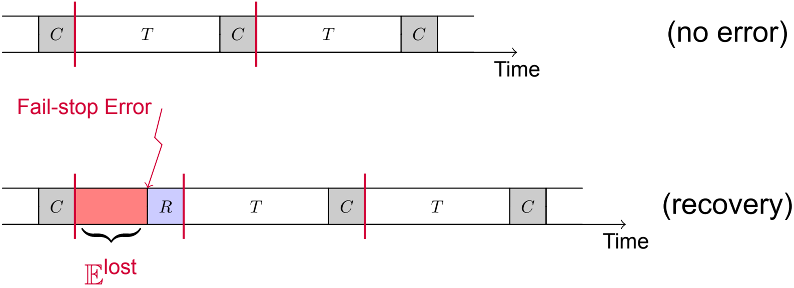
(no error)

$$\mathbb{E}(T) = \mathbb{P}_{no-error} \cdot (T + C)$$

$$+$$

Minimize Expected Execution Time

- T: Pattern length
- C: Checkpoint time
- R: Recovery time



$$\mathbb{E}(T) = \mathbb{P}_{no-error} \cdot (T + C) + \mathbb{P}_{error} \cdot \left(\mathbb{E}^{\text{lost}} + R + \mathbb{E}(T)\right)$$

Optimization

- Choose fault-model
- Minimize $\mathbb{E}(T)$ for T

...

Theorem. [Young 1974, Daly 2006]
$T^{opt} = \sqrt{2\mu C}$

- C: checkpoint cost
- μ (MTBF): Mean Time Between Failures

Extension: Multiple Levels of Checkpoints

Now, suppose that multiple types of checkpoints are available, e.g.

- Parallel File System (PFS)
- Local memory/SSD
- Partner copy/XOR

We can use **synchronized** checkpointing:

Synchronized checkpointing

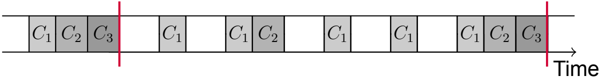
- k levels of checkpoints
- An error at level i kills all checkpoints $C_j < C_i$

Extension: Multiple Levels of Checkpoints

Now, suppose that multiple types of checkpoints are available, e.g.

- Parallel File System (PFS)
- Local memory/SSD
- Partner copy/XOR

We can use **synchronized** checkpointing:



Synchronized checkpointing

- k levels of checkpoints
- An error at level i kills all checkpoints $C_j < C_i$

Generalization to k Levels

Theorem (single-level) [Young,Daly]:

Optimal pattern length: $T^{\text{opt}} = \sqrt{\frac{2C}{\lambda}}$

Theorem (multi-level):

Optimal pattern length: $T^{\text{opt}} = \sqrt{\frac{\sum_{\ell=1}^{k} N_\ell^{\text{opt}} C_\ell}{\frac{1}{2}\sum_{\ell=1}^{k} \frac{\lambda_\ell}{N_\ell^{\text{opt}}}}}$

Optimal #chkpts at level ℓ: $N_\ell^{\text{opt}} = \sqrt{\frac{\lambda_\ell}{C_\ell} \cdot \frac{C_k}{\lambda_k}}\ , \ \forall \ell = 1, \dots, k$

II. Silent Errors

a.k.a. Silent Data Corruptions

Silent Data Corruptions

Characteristics

- Bit flip (Disk, RAM, Cache, Bus, ...)
- Problem: detection latency, wrong results

Number of errors proportional to area and circuit design

- 2002: **Unprotected address bus** ASCI Q at Los Alamos National Laboratory could not run more than one hour
- 2003: **No ECC** Virginia Tech $1,100$ Apple Power Mac G5 supercomputer could not boot
- 2010: **ECC protected** Jaguar saw 350 bit-flips/min
- 2010: **ECC protected** Jaguar saw 1 double-bit error/day
- 2014: Titan: **reported** > 1 Double Bit Error per week

Methods for Detecting Silent Errors

General-purpose approaches

- Replication [**Fiala et al. 2012**] or triple modular redundancy and voting [**Lyons and Vanderkulk 1962**]

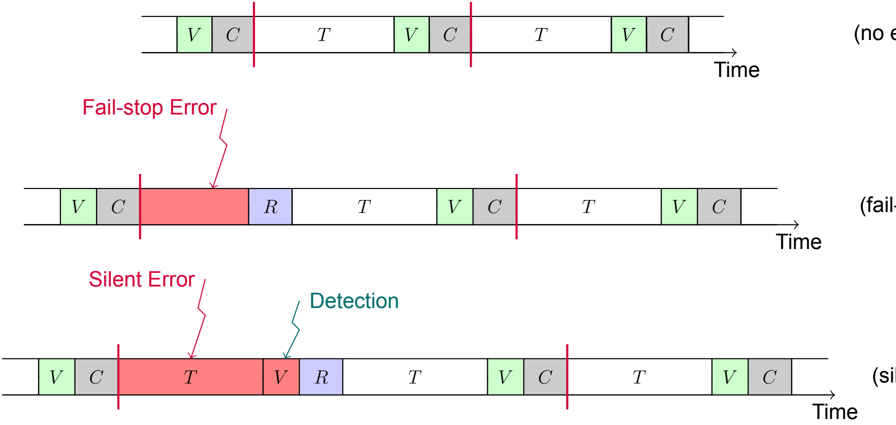
Application-specific approaches

- Algorithm-based fault tolerance (ABFT): checksums in dense matrices Limited to one error detection and/or correction in practice [**Huang and Abraham 1984**]
- Partial differential equations (PDE): use lower-order scheme as verification mechanism [**Benson, Schmit and Schreiber 2014**]
- Preconditioned conjugate gradients (PCG): orthogonalization check every k iterations, re-orthogonalization if problem detected [**Sao and Vuduc 2013, Chen 2013**]

Data-analytics approaches

- Dynamic monitoring of HPC datasets based on physical laws (e.g., temperature limit, speed limit) and space or temporal proximity [**Bautista-Gomez and Cappello 2014**]
- Time-series prediction, spatial multivariate interpolation [**Di et al. 2014**]

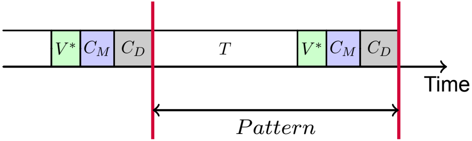
Coping with Fail-Stop and Silent Errors



What is the optimal checkpointing period?

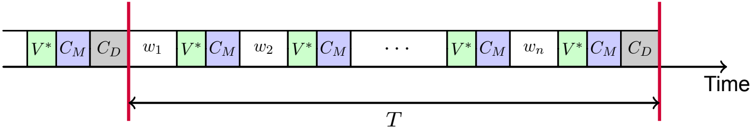
Resilience Patterns

Starting with base pattern



Simple pattern (Young-Daly)

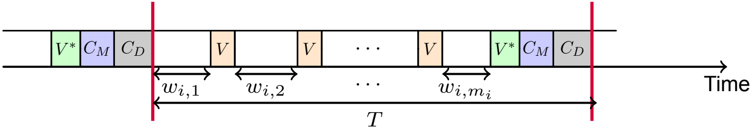
Adding verified memory checkpoints



Pattern with n segments

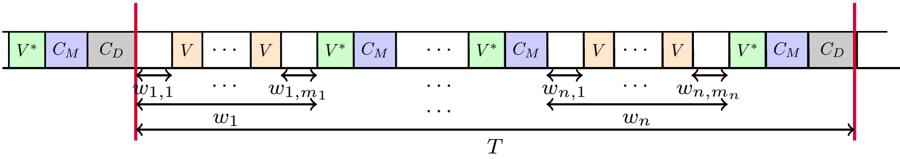
Resilience Patterns

Adding intermediate verifications between memory checkpoints



Segment w_i has m_i chunks

Putting everything together



Full pattern

...

Theorems

Our contributions:

Pattern	T^*	n^*	m^*	$H^*(\textsc{Pattern})$
P_D	$\sqrt{\frac{V^*+C_M+C_D}{\lambda^s+\frac{\lambda^f}{2}}}$	–	–	$2\sqrt{\left(\lambda^s+\frac{\lambda^f}{2}\right)(V^*+C_M+C_D)}$
P_{DV^*}	$\sqrt{\frac{m^*V^*+C_M+C_D}{\frac{1}{2}\left(1+\frac{1}{m^*}\right)\lambda^s+\frac{\lambda^f}{2}}}$	–	$\sqrt{\frac{\lambda^s}{\lambda^s+\lambda^f}\cdot\frac{C_M+C_D}{V^*}}$	$\sqrt{2(\lambda^s+\lambda^f)C_M+C_D}+\sqrt{2\lambda^s V^*}$
P_{DV}	$\sqrt{\frac{(m^*-1)V+V^*+C_M+C_D}{\frac{1}{2}\left(1+\frac{2-r}{(m^*-2)r+2}\right)\lambda^s+\frac{\lambda^f}{2}}}$	–	$2-\frac{2}{r}+\sqrt{\frac{\lambda^s}{\lambda^s+\lambda^f}}$ $\times\sqrt{\frac{2-r}{r}\left(\frac{V^*+C_M+C_D}{V}-\frac{2-r}{r}\right)}$	$\sqrt{2(\lambda^s+\lambda^f)\left(V^*-\frac{2-r}{r}V+C_M+C_D\right)}$ $+\sqrt{2\lambda^s\frac{2-r}{r}V}$
P_{DM}	$\sqrt{\frac{n^*(V^*+C_M)+C_D}{\frac{\lambda^s}{n^*}+\frac{\lambda^f}{2}}}$	$\sqrt{\frac{2\lambda^s}{\lambda^f}\cdot\frac{C_D}{V^*+C_M}}$	–	$2\sqrt{\lambda^s(V^*+C_M)}+\sqrt{2\lambda^f C_D}$
P_{DMV^*}	$\sqrt{\frac{n^*m^*V^*+n^*C_M+C_D}{\frac{1}{2}\left(1+\frac{1}{m^*}\right)\frac{\lambda^s}{n^*}+\frac{\lambda^f}{2}}}$	$\sqrt{\frac{\lambda^s}{\lambda^f}\cdot\frac{C_D}{C_M}}$	$\sqrt{\frac{C_M}{V^*}}$	$\sqrt{2\lambda^f C_D}+\sqrt{2\lambda^s C_M}+\sqrt{2\lambda^s V^*}$
P_{DMV}	$\sqrt{\frac{n^*(m^*-1)V+n^*(V^*+C_M)+C_D}{\frac{1}{2}\left(1+\frac{2-r}{(m^*-2)r+2}\right)\frac{\lambda^s}{n^*}+\frac{\lambda^f}{2}}}$	$\sqrt{\frac{\lambda^s}{\lambda^f}\cdot\frac{C_D}{V^*-\frac{2-r}{r}V+C_M}}$	$2-\frac{2}{r}$ $+\sqrt{\frac{2-r}{r}\left(\frac{V^*+C_M}{V}-\frac{2-r}{r}\right)}$	$\sqrt{2\lambda^f C_D}+\sqrt{2\lambda^s\left(V^*-\frac{2-r}{r}V+C_M\right)}$ $+\sqrt{2\lambda^s\frac{2-r}{r}V}$

Summary and Future Work

Resilience patterns:

- Checkpointing for fail-stop errors
- Verifications for silent errors
- Multi-level checkpointing for both
- Models and optimal solutions

Currently, I work on designing new detection techniques

Algorithm Based Fault Tolerance **(ABFT)**

Any application specific technique used to cope with faults.

Consider the blocked matrix multiplication $C = A \times B$.

Application Workflow

for $i = 1$ to $\lceil \frac{m}{b} \rceil$ **do**
  for $j = 1$ to $\lceil \frac{m}{b} \rceil$ **do**
    for $k = 1$ to $\lceil \frac{m}{b} \rceil$ **do**
      $C_{i,j} \leftarrow C_{i,j} + A_{i,k} \times B_{k,j}$

- m matrix size
- b block size

ABFT can be used to add per-block verification.

Algorithm Based Fault Tolerance **(ABFT)**

Any application specific technique used to cope with faults.

Consider the blocked matrix multiplication $C = A \times B$.

Application Workflow

for $i = 1$ to $\lceil \frac{m}{b} \rceil$ **do**
  for $j = 1$ to $\lceil \frac{m}{b} \rceil$ **do**
    for $k = 1$ to $\lceil \frac{m}{b} \rceil$ **do**
      $C_{i,j} \leftarrow C_{i,j} + A_{i,k} \times B_{k,j}$

- m matrix size
- b block size

ABFT can be used to add per-block verification.

Algorithm Based Fault Tolerence

Let $e^T = [1, 1, \cdots, 1]$, we define

$$A^c := \begin{pmatrix} A \\ e^T A \end{pmatrix},\ B^r := \begin{pmatrix} B & Be \end{pmatrix},\ C^f := \begin{pmatrix} C & Ce \\ e^T C & e^T Ce \end{pmatrix}.$$

Where A^c is the **column checksum matrix**, B^r is the **row checksum matrix** and C^f is the **full checksum matrix**.

$$\begin{aligned} A^c \times B^r &= \begin{pmatrix} A \\ e^T A \end{pmatrix} \times \begin{pmatrix} B & Be \end{pmatrix} \\ &= \begin{pmatrix} AB & ABe \\ e^T AB & e^T ABe \end{pmatrix} = \begin{pmatrix} C & Ce \\ e^T C & e^T Ce \end{pmatrix} = C^f \end{aligned}$$

Algorithm Based Fault Tolerence

Let $e^T = [1, 1, \cdots, 1]$, we define

$$A^c := \begin{pmatrix} A \\ e^T A \end{pmatrix},\ B^r := \begin{pmatrix} B & Be \end{pmatrix},\ C^f := \begin{pmatrix} C & Ce \\ e^T C & e^T Ce \end{pmatrix}.$$

Where A^c is the **column checksum matrix**, B^r is the **row checksum matrix** and C^f is the **full checksum matrix**.

$$\begin{aligned} A^c \times B^r &= \begin{pmatrix} A \\ e^T A \end{pmatrix} \times \begin{pmatrix} B & Be \end{pmatrix} \\ &= \begin{pmatrix} AB & ABe \\ e^T AB & e^T ABe \end{pmatrix} = \begin{pmatrix} C & Ce \\ e^T C & e^T Ce \end{pmatrix} = C^f \end{aligned}$$

ABFT: Detection

Let us build a small example:

$$A^c = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 2 & 1 & 2 \\ 5 & 4 & 2 \end{pmatrix},\; B^r = \begin{pmatrix} 1 & 1 & 1 & 3 \\ 2 & 0 & 3 & 5 \\ 0 & 2 & 2 & 4 \end{pmatrix},$$

$$C^f = A^c \times B^r = \begin{pmatrix} 5 & 1 & 7 & 13 \\ 4 & 3 & 5 & 11 \\ 4 & 6 & 9 & 19 \\ 13 & 9 & 21 & 43 \end{pmatrix}$$

Everything seems fine. However, a silent error has occurred !

Indeed, recomputing the checksums we find that:

$$\begin{pmatrix} 5 & + & 1 & + & 7 & = & 13 \\ 4 & + & 3 & + & 5 & = & 12 \\ 4 & + & 6 & + & 9 & = & 19 \\ 13 & + & 10 & + & 21 & = & 44 \end{pmatrix}$$

Checksums do not match !

ABFT: Detection

Let us build a small example:

$$A^c = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 2 & 1 & 2 \\ 5 & 4 & 2 \end{pmatrix}, \; B^r = \begin{pmatrix} 1 & 1 & 1 & 3 \\ 2 & 0 & 3 & 5 \\ 0 & 2 & 2 & 4 \end{pmatrix},$$

$$C^f = A^c \times B^r = \begin{pmatrix} 5 & 1 & 7 & 13 \\ 4 & 3 & 5 & 11 \\ 4 & 6 & 9 & 19 \\ 13 & 9 & 21 & 43 \end{pmatrix}$$

Everything seems fine. However, a silent error has occurred !

Indeed, recomputing the checksums we find that:

$$\begin{pmatrix} 5 & + & 1 & + & 7 & = & 13 \\ 4 & + & 3 & + & 5 & = & 12 \\ 4 & + & 6 & + & 9 & = & 19 \\ 13 & + & 10 & + & 21 & = & 44 \end{pmatrix}$$

Checksums do not match !

ABFT: Detection

Let us build a small example:

$$A^c = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 2 & 1 & 2 \\ 5 & 4 & 2 \end{pmatrix}, \; B^r = \begin{pmatrix} 1 & 1 & 1 & 3 \\ 2 & 0 & 3 & 5 \\ 0 & 2 & 2 & 4 \end{pmatrix},$$

$$C^f = A^c \times B^r = \begin{pmatrix} 5 & 1 & 7 & 13 \\ 4 & 3 & 5 & 11 \\ 4 & 6 & 9 & 19 \\ 13 & 9 & 21 & 43 \end{pmatrix}$$

Everything seems fine. However, a silent error has occurred !

Indeed, recomputing the checksums we find that:

$$\begin{pmatrix} 5 & + & 1 & + & 7 & = & 13 \\ 4 & + & 3 & + & 5 & = & 12 \\ 4 & + & 6 & + & 9 & = & 19 \\ 13 & + & 10 & + & 21 & = & 44 \end{pmatrix}$$

Checksums do not match !

ABFT: Detection

Let us build a small example:

$$A^c = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 2 & 1 & 2 \\ 5 & 4 & 2 \end{pmatrix}, \; B^r = \begin{pmatrix} 1 & 1 & 1 & 3 \\ 2 & 0 & 3 & 5 \\ 0 & 2 & 2 & 4 \end{pmatrix},$$

$$C^f = A^c \times B^r = \begin{pmatrix} 5 & 1 & 7 & 13 \\ 4 & 3 & 5 & 11 \\ 4 & 6 & 9 & 19 \\ 13 & 9 & 21 & 43 \end{pmatrix}$$

Everything seems fine. However, a silent error has occurred !

Indeed, recomputing the checksums we find that:

$$\begin{pmatrix} 5 & + & 1 & + & 7 & = & 13 \\ 4 & + & 3 & + & 5 & = & 12 \\ 4 & + & 6 & + & 9 & = & 19 \\ 13 & + & 10 & + & 21 & = & 44 \end{pmatrix}$$

Checksums do not match !

ABFT: Correction

$$C^f = A^c \times B^r = \begin{pmatrix} 5 & 1 & 7 & 13 \\ 4 & 3 & 5 & 11 \\ 4 & 6 & 9 & 19 \\ 13 & 9 & 21 & 43 \end{pmatrix}, \begin{pmatrix} 5 & + & 1 & + & 7 & = & 13 \\ 4 & + & 3 & + & 5 & = & 12 \\ 4 & + & 6 & + & 9 & = & 19 \\ 13 & + & 10 & + & 21 & = & 44 \end{pmatrix}$$

Both checksums are affected, giving out the location of the error.

We solve:

$$4 + x + 5 = 11 \qquad 1 + x + 6 = 9$$

$$x = 11 - 5 - 4 = 2 \qquad x = 9 - 6 - 1 = 2$$

Recomputing the checksums we find that:

$$\begin{pmatrix} 5 & + & 1 & + & 7 & = & 13 \\ 4 & + & 2 & + & 5 & = & 11 \\ 4 & + & 6 & + & 9 & = & 19 \\ 13 & + & 9 & + & 21 & = & 43 \end{pmatrix}$$ Checksums match!

ABFT: Correction

$$C^f = A^c \times B^r = \begin{pmatrix} 5 & 1 & 7 & 13 \\ 4 & 3 & 5 & 11 \\ 4 & 6 & 9 & 19 \\ 13 & 9 & 21 & 43 \end{pmatrix}, \begin{pmatrix} 5 & + & 1 & + & 7 & = & 13 \\ 4 & + & 3 & + & 5 & = & 12 \\ 4 & + & 6 & + & 9 & = & 19 \\ 13 & + & 10 & + & 21 & = & 44 \end{pmatrix}$$

Both checksums are affected, giving out the location of the error.

We solve:

$$\begin{aligned} 4 + x + 5 &= 11 \\ x &= 11 - 5 - 4 = 2 \end{aligned} \qquad \begin{aligned} 1 + x + 6 &= 9 \\ x &= 9 - 6 - 1 = 2 \end{aligned}$$

Recomputing the checksums we find that:

$$\begin{pmatrix} 5 & + & 1 & + & 7 & = & 13 \\ 4 & + & 2 & + & 5 & = & 11 \\ 4 & + & 6 & + & 9 & = & 19 \\ 13 & + & 9 & + & 21 & = & 43 \end{pmatrix}$$

Checksums match!

ABFT: Correction

$$C^f = A^c \times B^r = \begin{pmatrix} 5 & 1 & 7 & 13 \\ 4 & 3 & 5 & 11 \\ 4 & 6 & 9 & 19 \\ 13 & 9 & 21 & 43 \end{pmatrix}, \begin{pmatrix} 5 & + & 1 & + & 7 & = & 13 \\ 4 & + & 3 & + & 5 & = & 12 \\ 4 & + & 6 & + & 9 & = & 19 \\ 13 & + & 10 & + & 21 & = & 44 \end{pmatrix}$$

Both checksums are affected, giving out the location of the error.

We solve:

$$4 + x + 5 = 11 \qquad 1 + x + 6 = 9$$

$$x = 11 - 5 - 4 = 2 \qquad x = 9 - 6 - 1 = 2$$

Recomputing the checksums we find that:

$$\begin{pmatrix} 5 & + & 1 & + & 7 & = & 13 \\ 4 & + & 2 & + & 5 & = & 11 \\ 4 & + & 6 & + & 9 & = & 19 \\ 13 & + & 9 & + & 21 & = & 43 \end{pmatrix}$$ Checksums match!