# TFM-FernandoMartín

Fernando Martín Canfrán

January 15, 2025

# 1 Analysis of Results

Objectives to answer:

- How does it behave the $DP_{ATM}$ . Is it worthy wrt to a sequential / non-multiple filter version?

## 1.1 Small GDB

Note: 1f is the equivalent sequential version... in these experiments.

### 1.1.1 For each stream size

1. How the different configurations behave -

**Radial plots - for each stream - comparing among the different number of cores configurations**
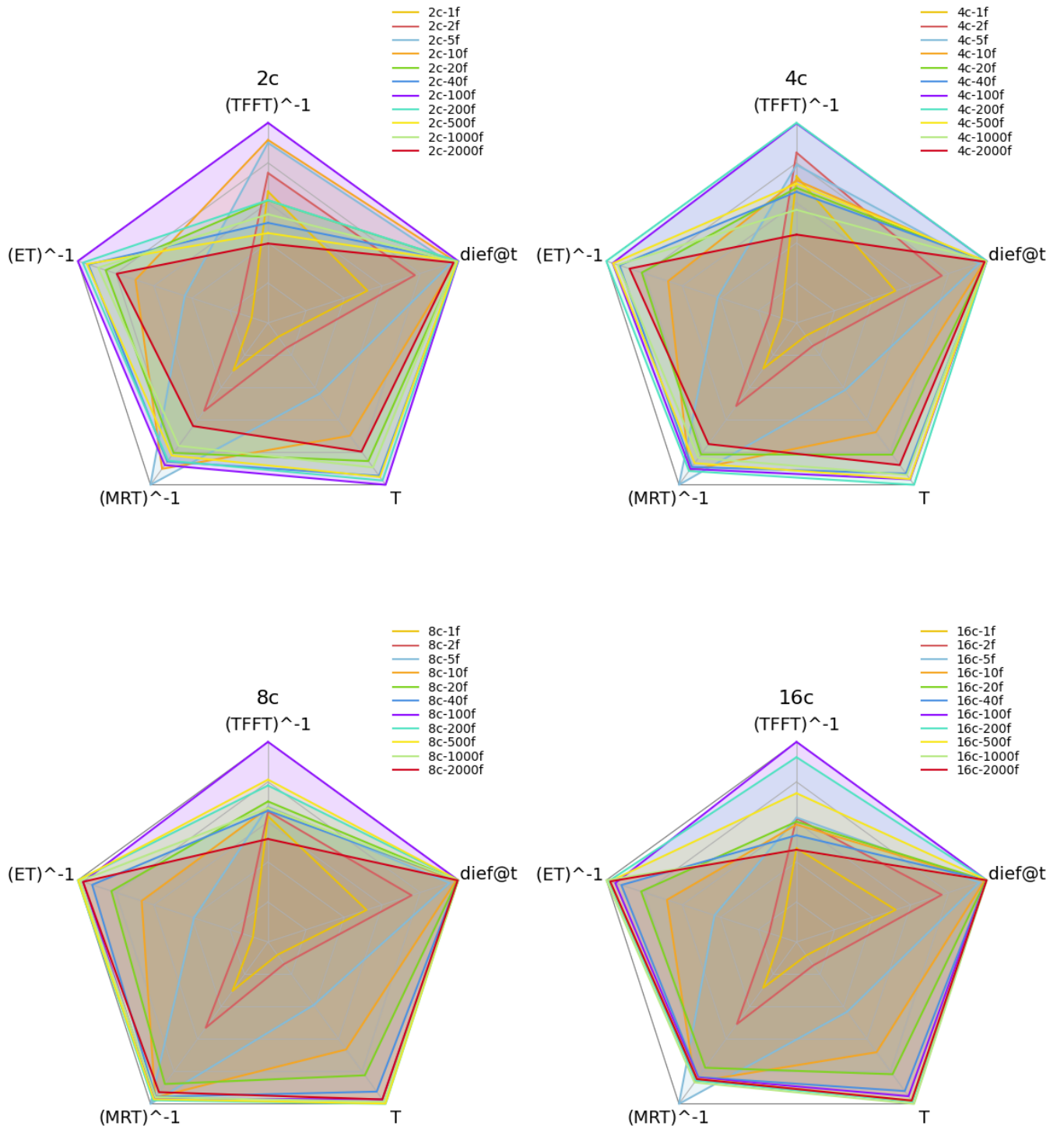
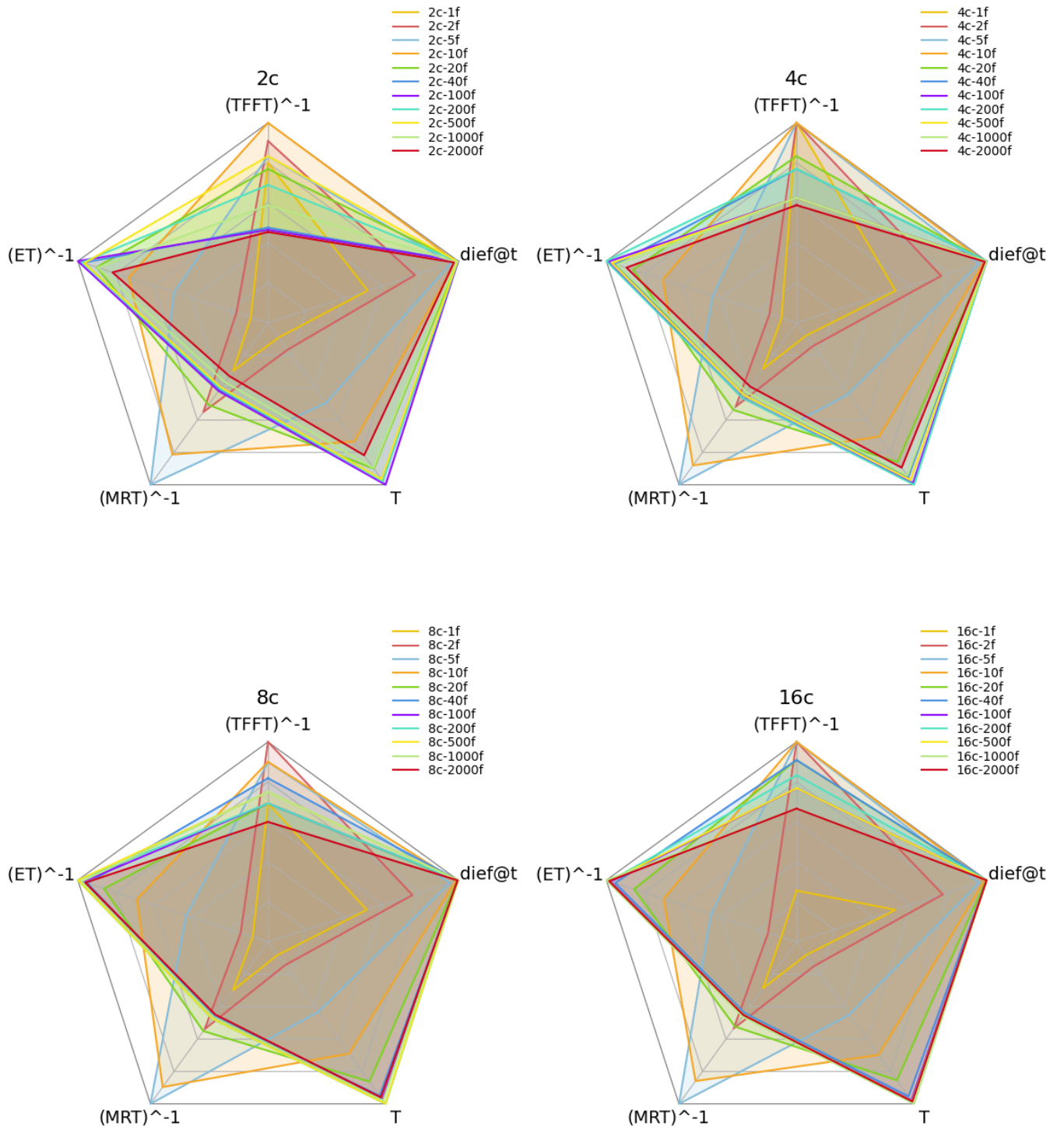Figure 1: Radial Plots for small stream: 30-0.02

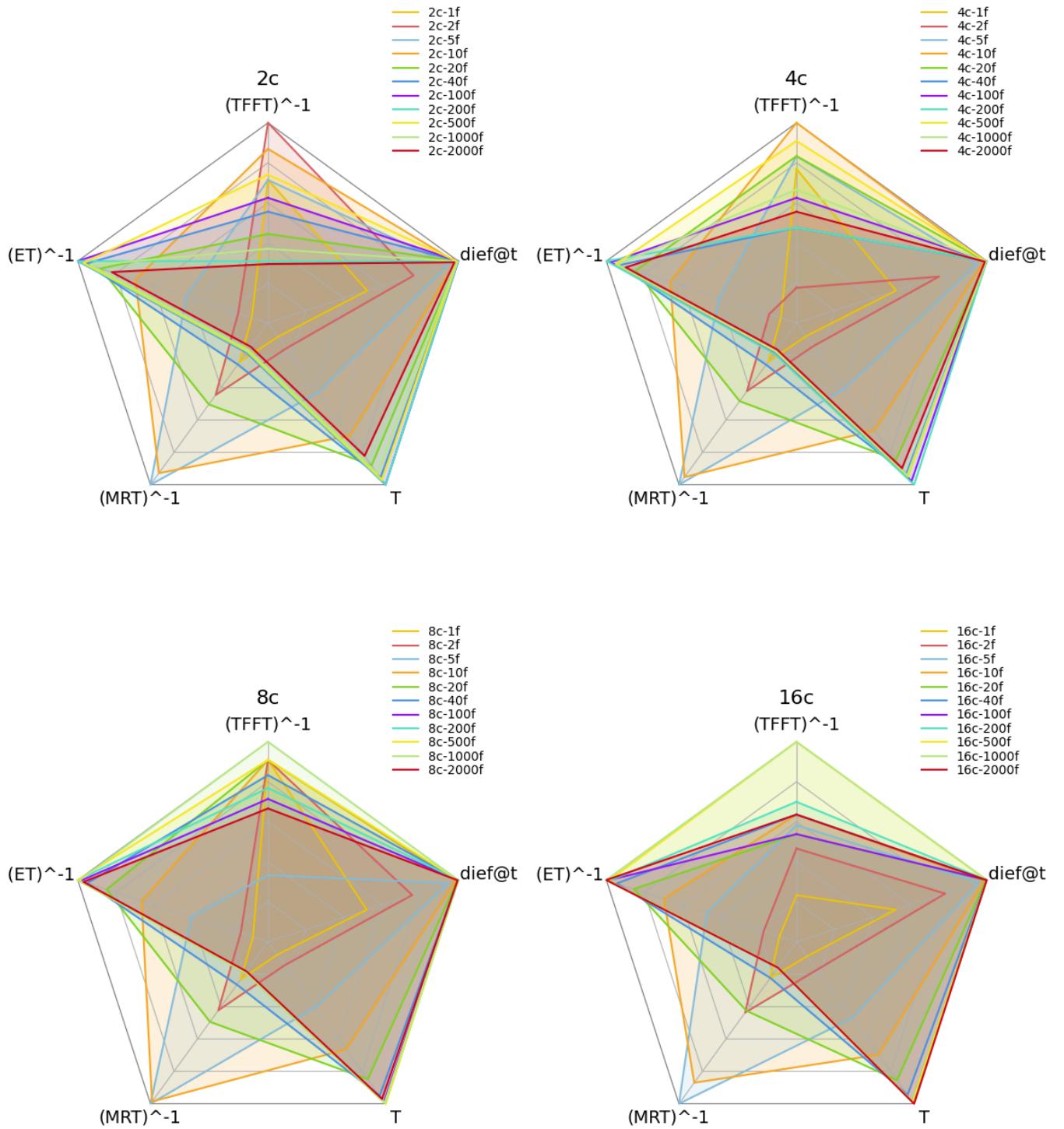Figure 2: Radial Plots for medium stream: 60-0.02

Figure 3: Radial Plots for big stream: 120-0.02

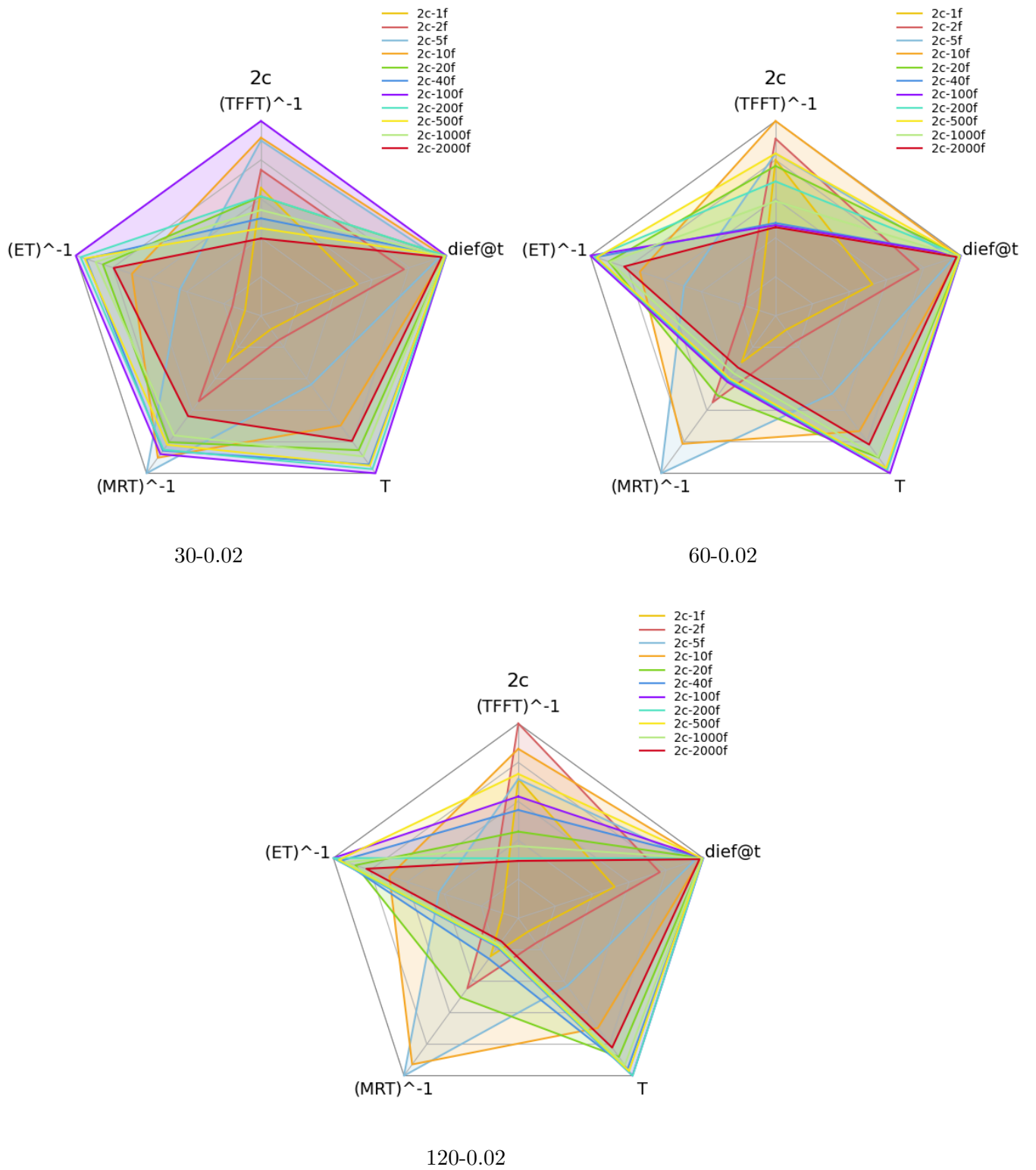**Radial plots - for a core number - different stream sizes**

30-0.02



60-0.02



120-0.02

Figure 4: Radial Plots for 2c

**Radial plots - for a core number - different stream sizes**

30-0.02



60-0.02



120-0.02
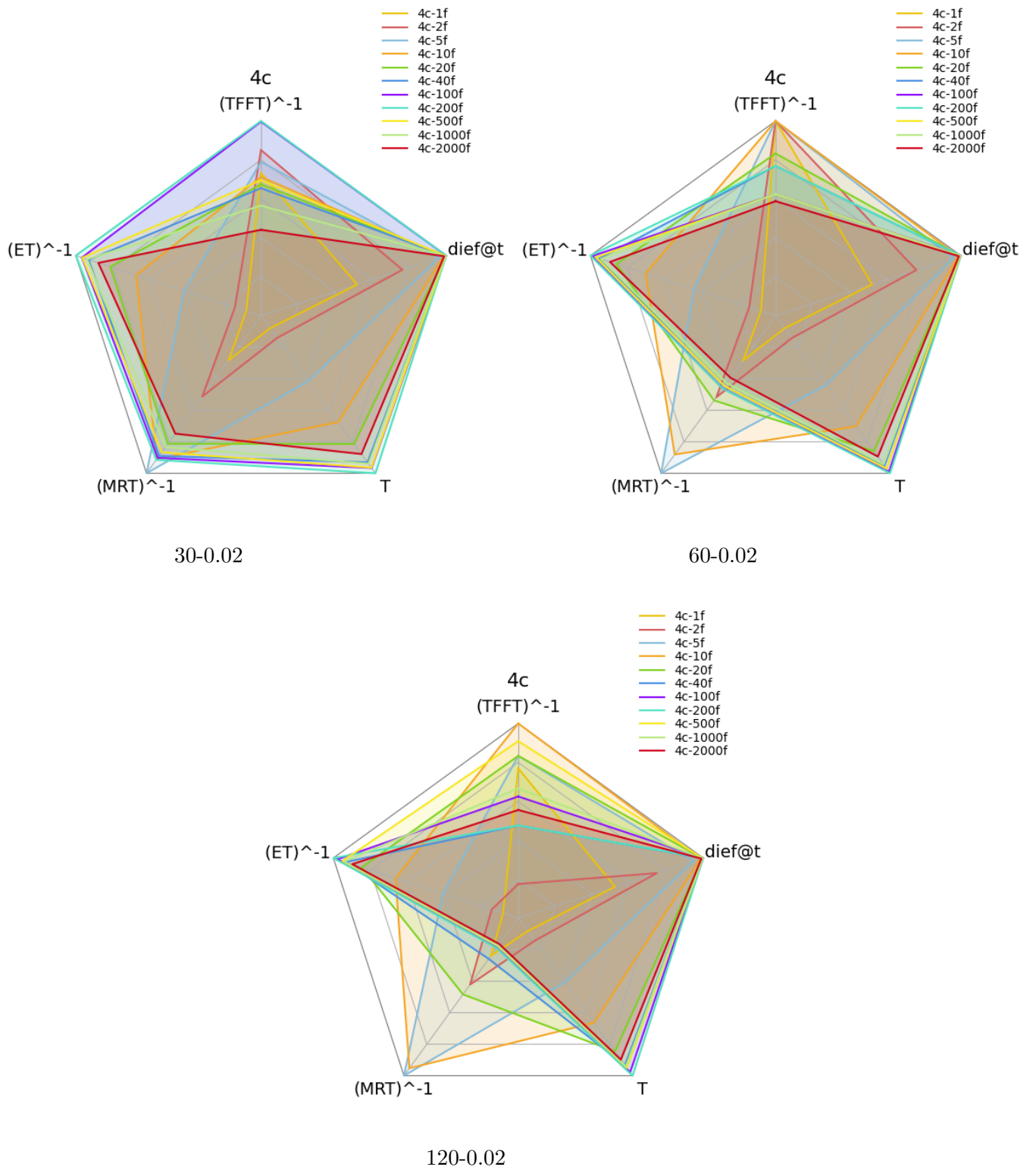
Figure 5: Radial Plots for 4c

**Radial plots - for a core number - different stream sizes**
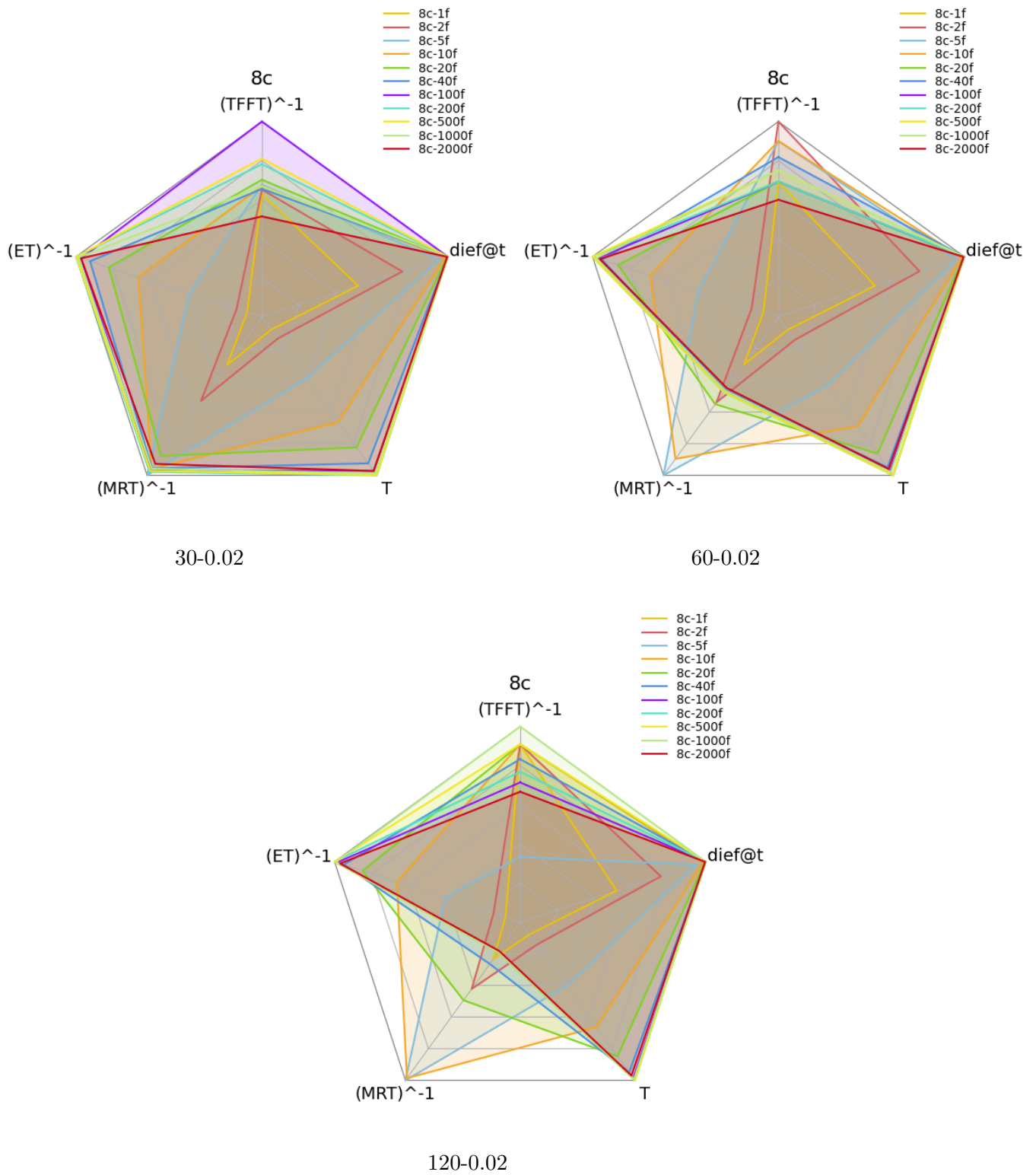
30-0.02



60-0.02



120-0.02

Figure 6: Radial Plots for 8c

**Radial plots - for a core number - different stream sizes**
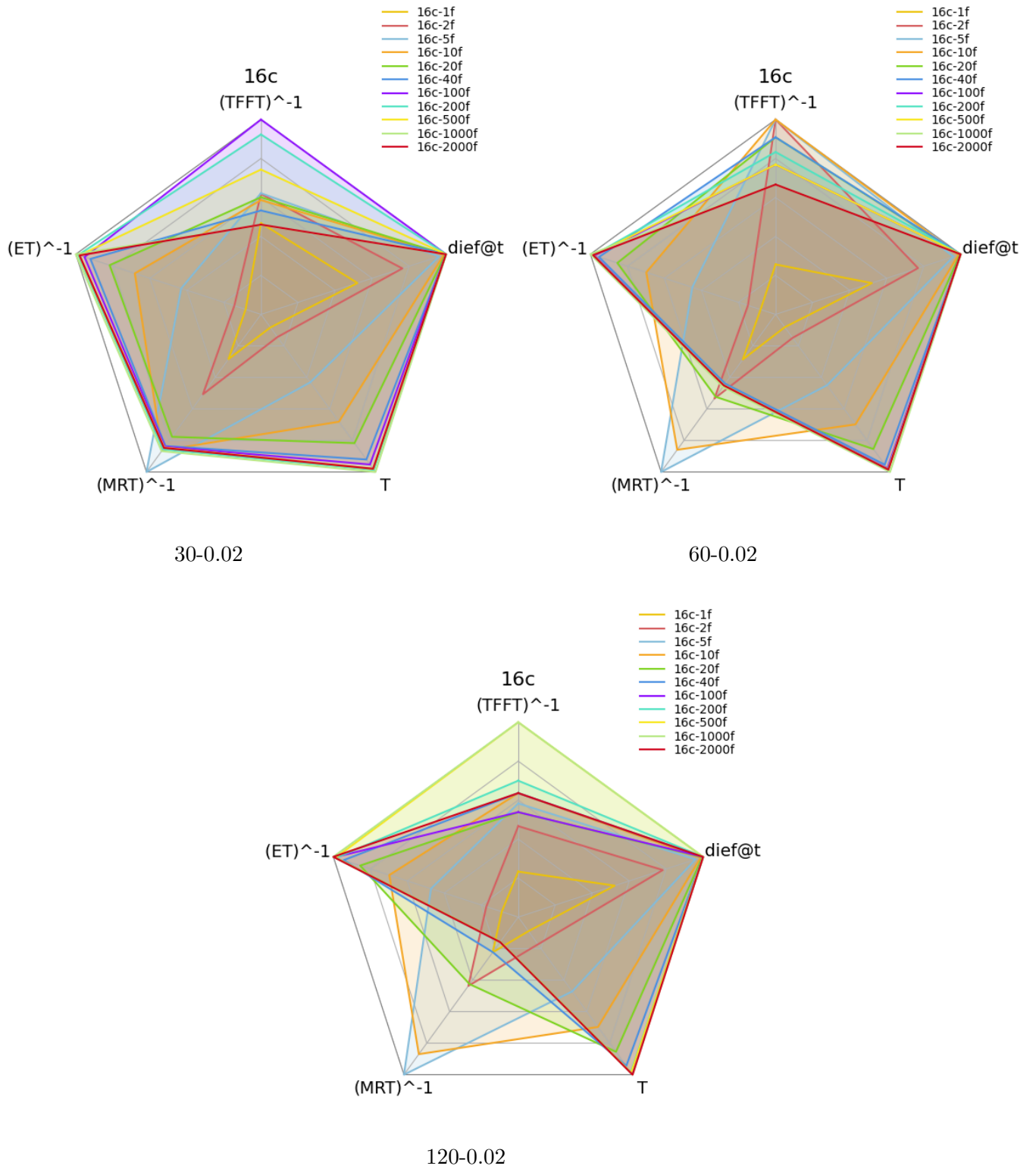
30-0.02

60-0.02

120-0.02

Figure 7: Radial Plots for 16c

**MRT plots - for a core number 4c - different stream sizes**
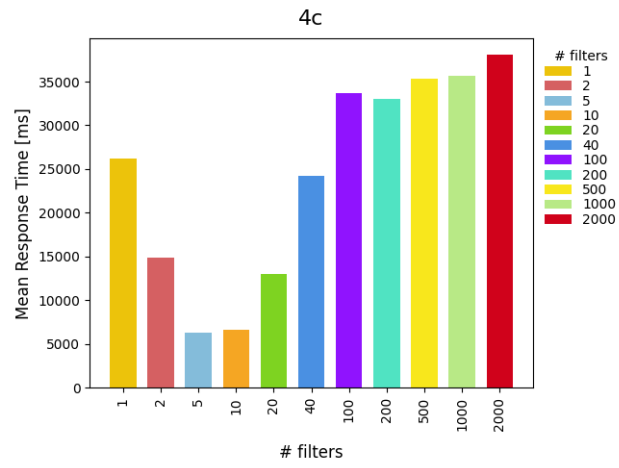
30-0.02



60-0.02



120-0.02

Figure 8: MRT Plots for 4c

**Exec time plots - for a core number - different stream sizes**
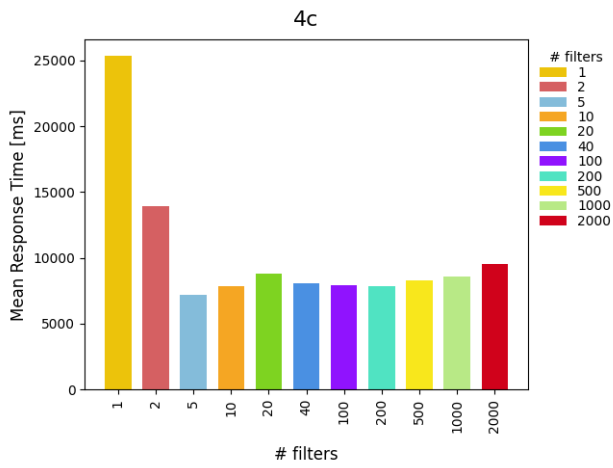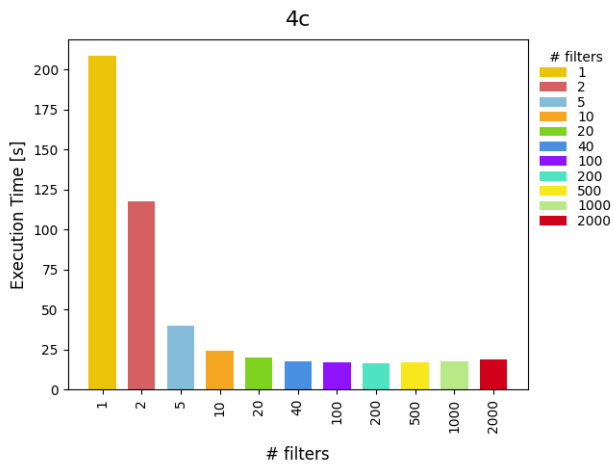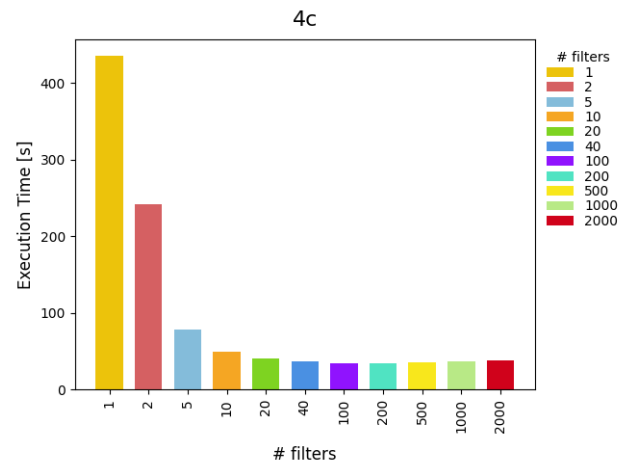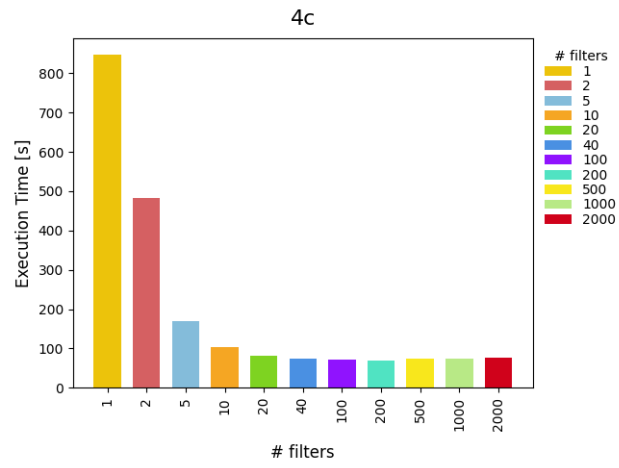
30-0.02



60-0.02



120-0.02

Figure 9: Execution Time Plots for 4c
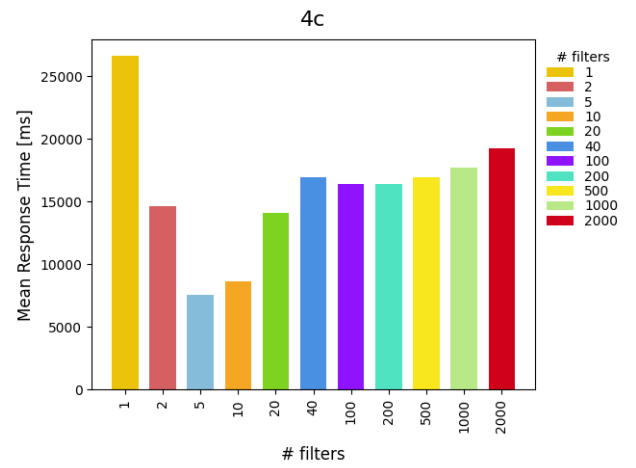
**MRT plots - for a core number 16 - different stream sizes**
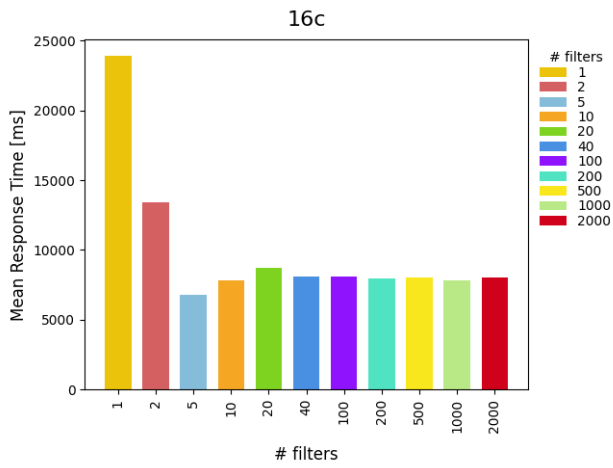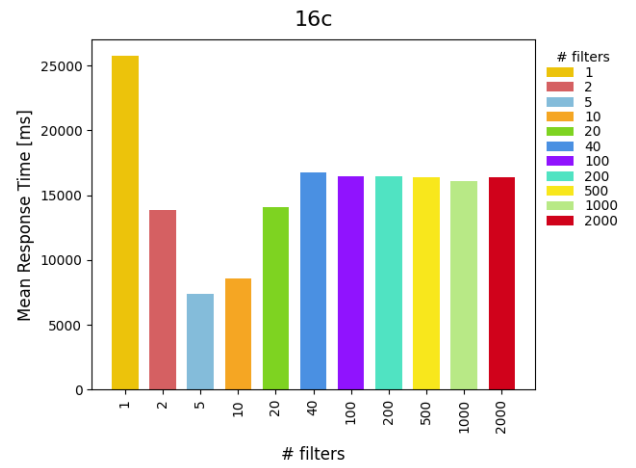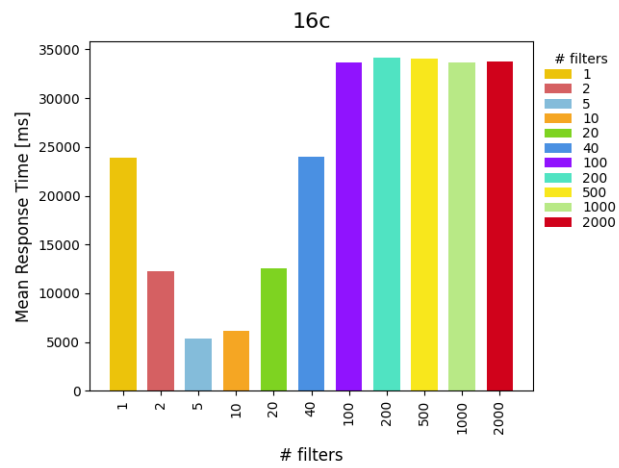
30-0.02



60-0.02



120-0.02

Figure 10: MRT Plots for 16c

**Exec time plots - for a core number - different stream sizes**
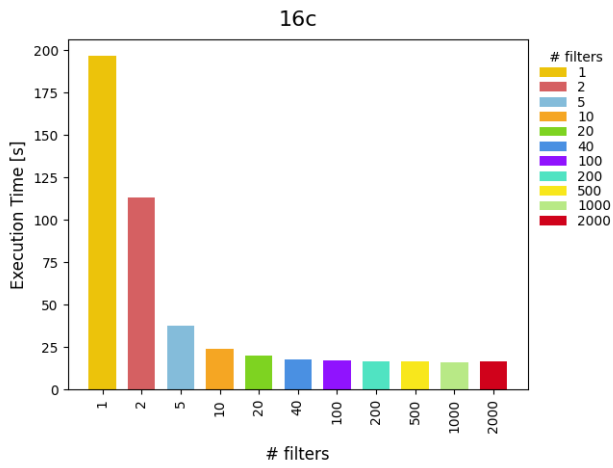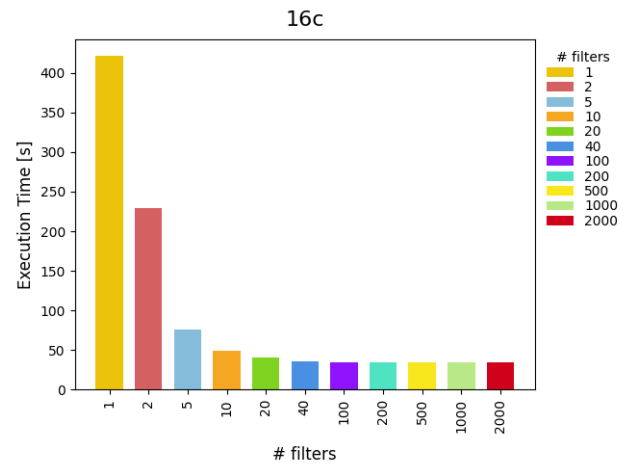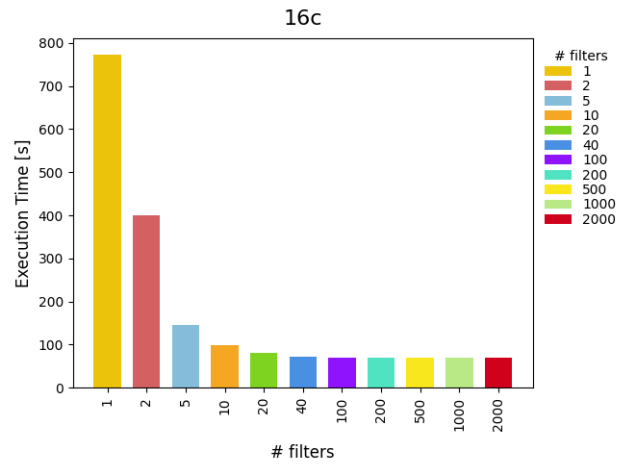
30-0.02



60-0.02



120-0.02

Figure 11: Execution Time Plots for 16c

**Trace and response time trace for 16c and big stream**
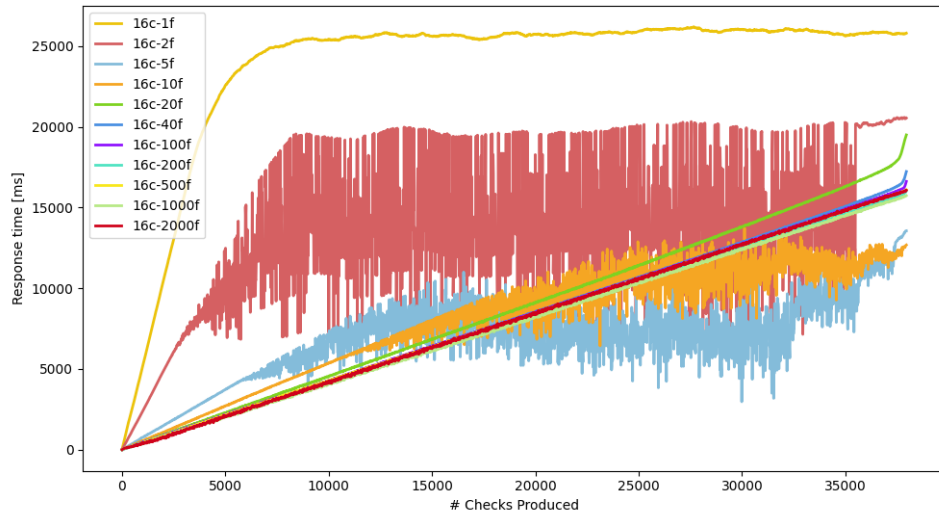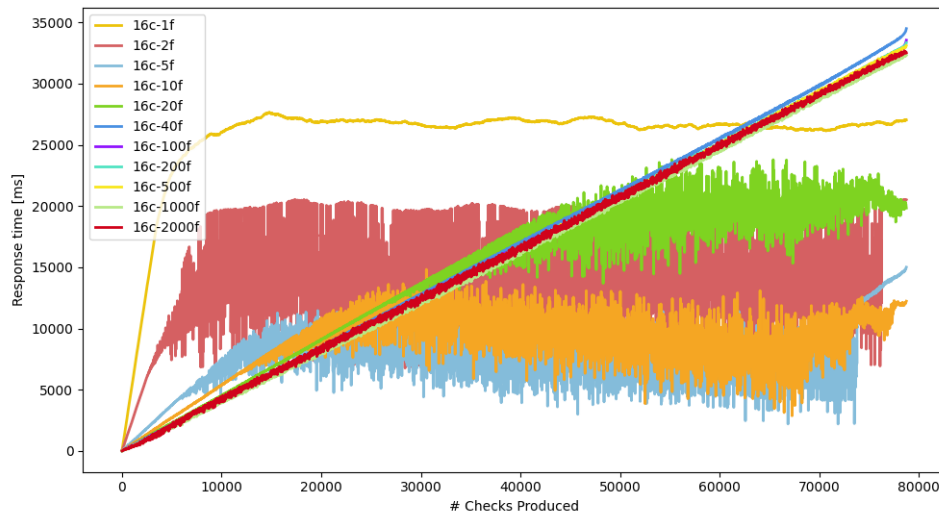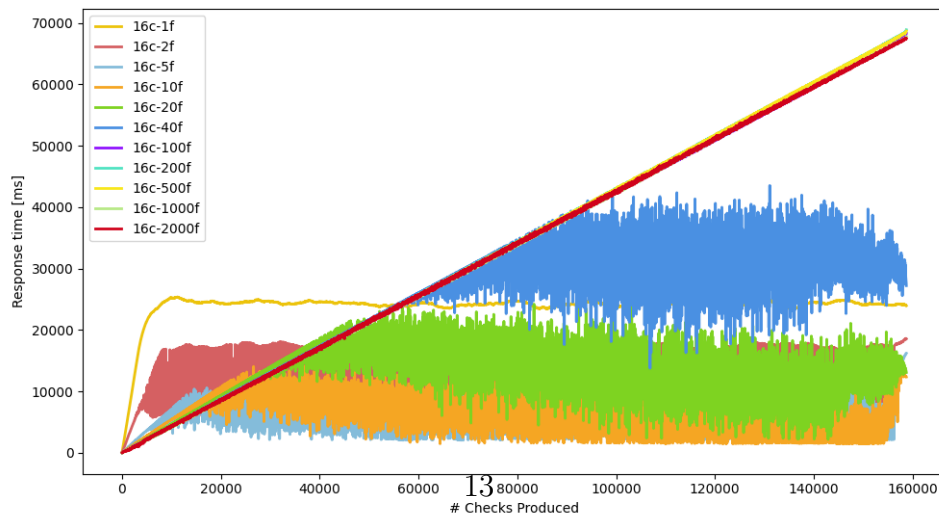
(a) 30-0.02



(b) 60-0.02



(c) 120-0.02

Figure 12: Execution Time Plots for 16c
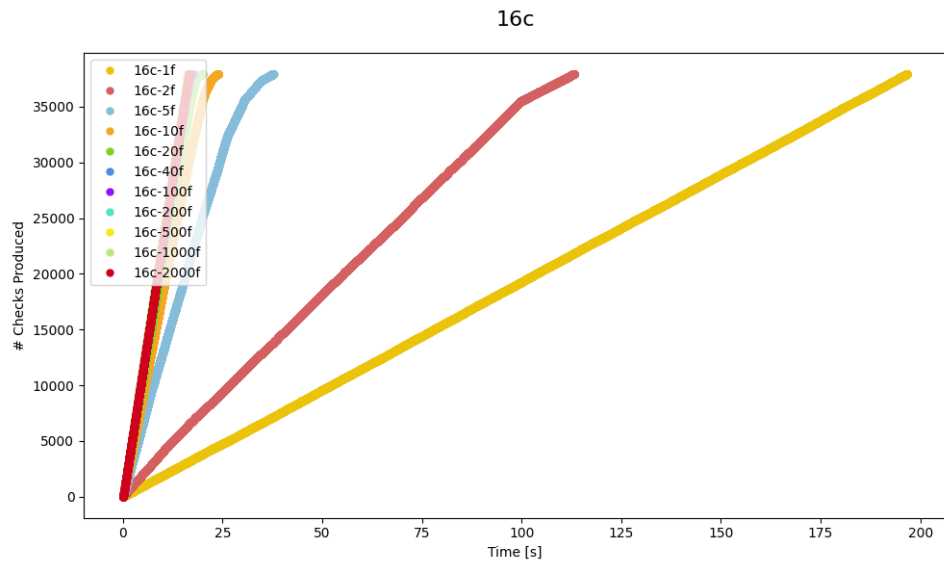
(a) 30-0.02
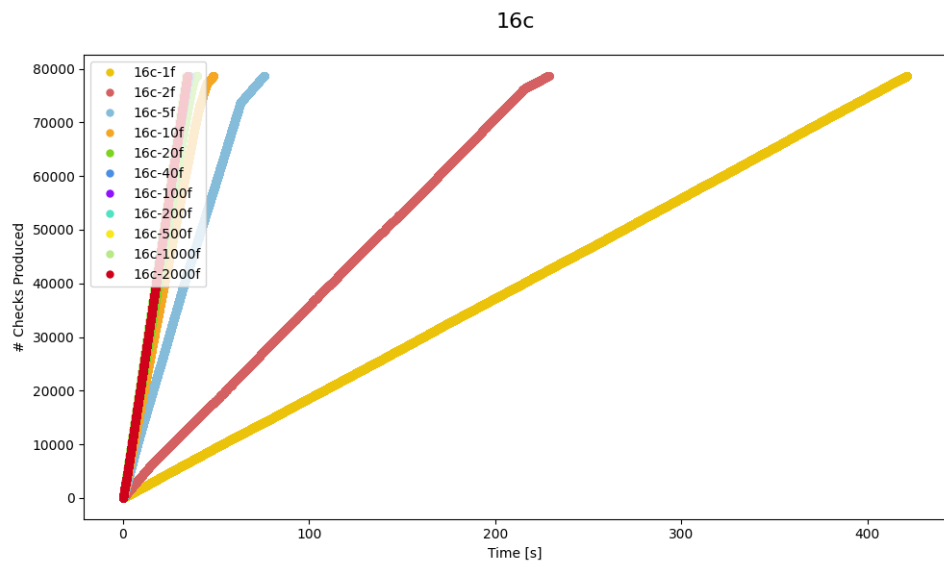


(b) 60-0.02
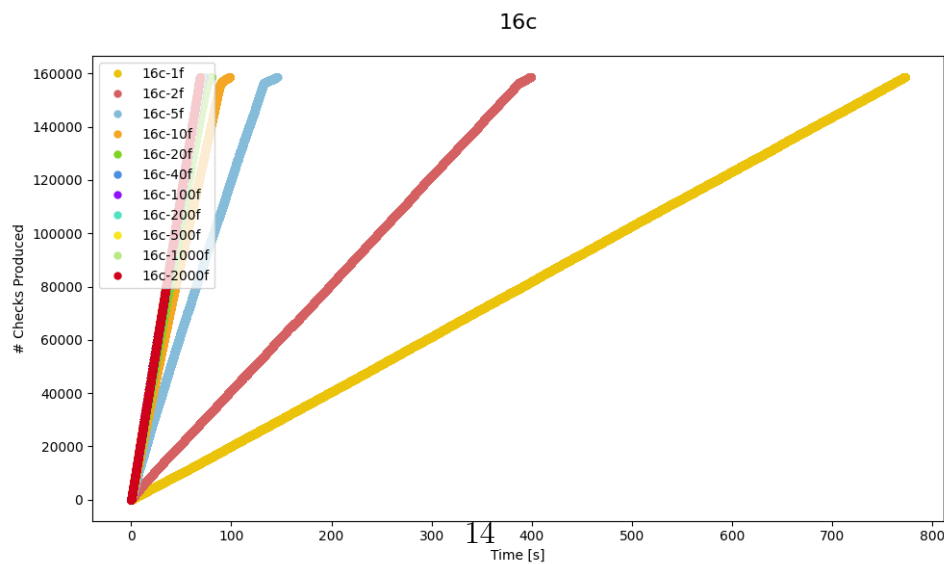
(c) 120-0.02

Figure 13: Execution Time Plots for 16c

**Trace and response time trace for 16c and big stream**

2. Show how incrementing the resources (number of cores) help to improve the behavior
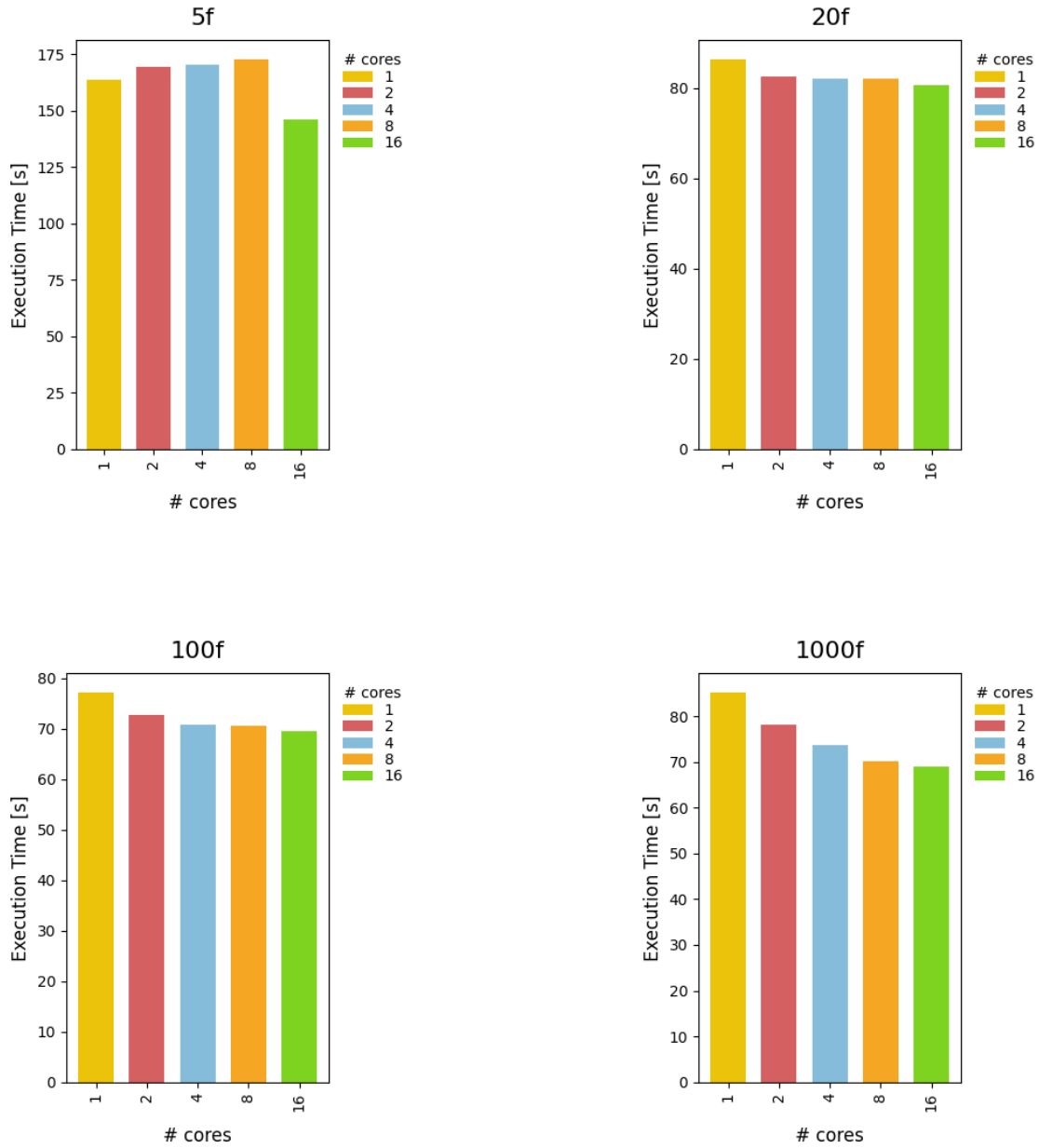


Figure 14: Radial Plots for big stream: 120-0.02
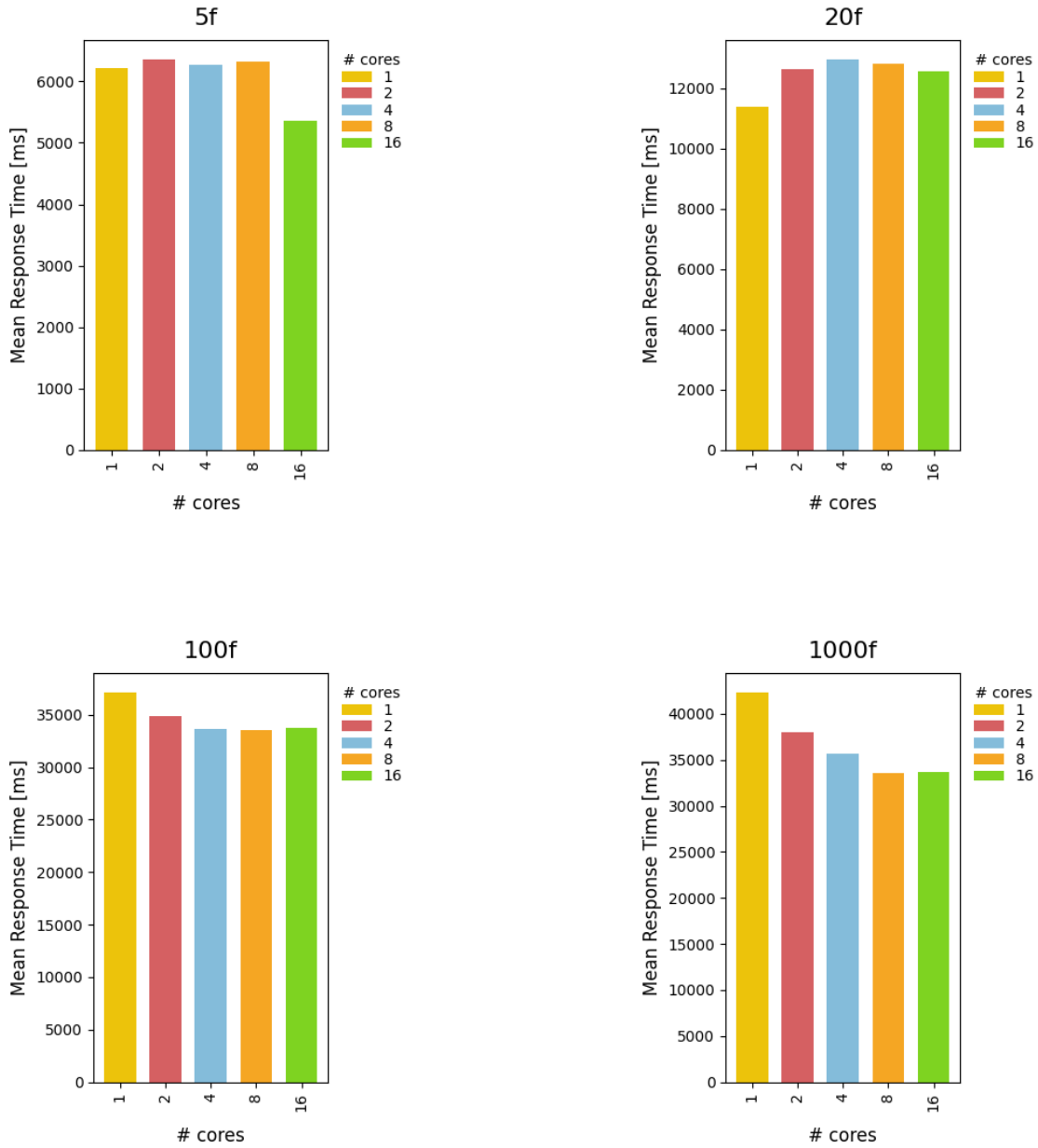
Figure 15: Radial Plots for big stream: 120-0.02

Ideas on the analysis so far:

- More cores help to improve the behavior, especially for the variants with large number of filters (expected).

- For a same core configuration (e.g. 16 cores), the total execution time (time to process all the stream input) is larger for the approaches with less cores, tending to decrease. However the continuous behavior is different: the best is observed for a number of filters in the range of 5-10 filters. From that point and on the continuous behavior tends to degrade when increasing the number

filters, especially for bigger stream sizes. Even larger than the lowest number of filters version (close to a sequential version) in these cases. This can be due to an overhead on the number of goroutines utilized and the overhead in the communication of the pipeline that this is causing.

These same conclusions are easily to observe in:
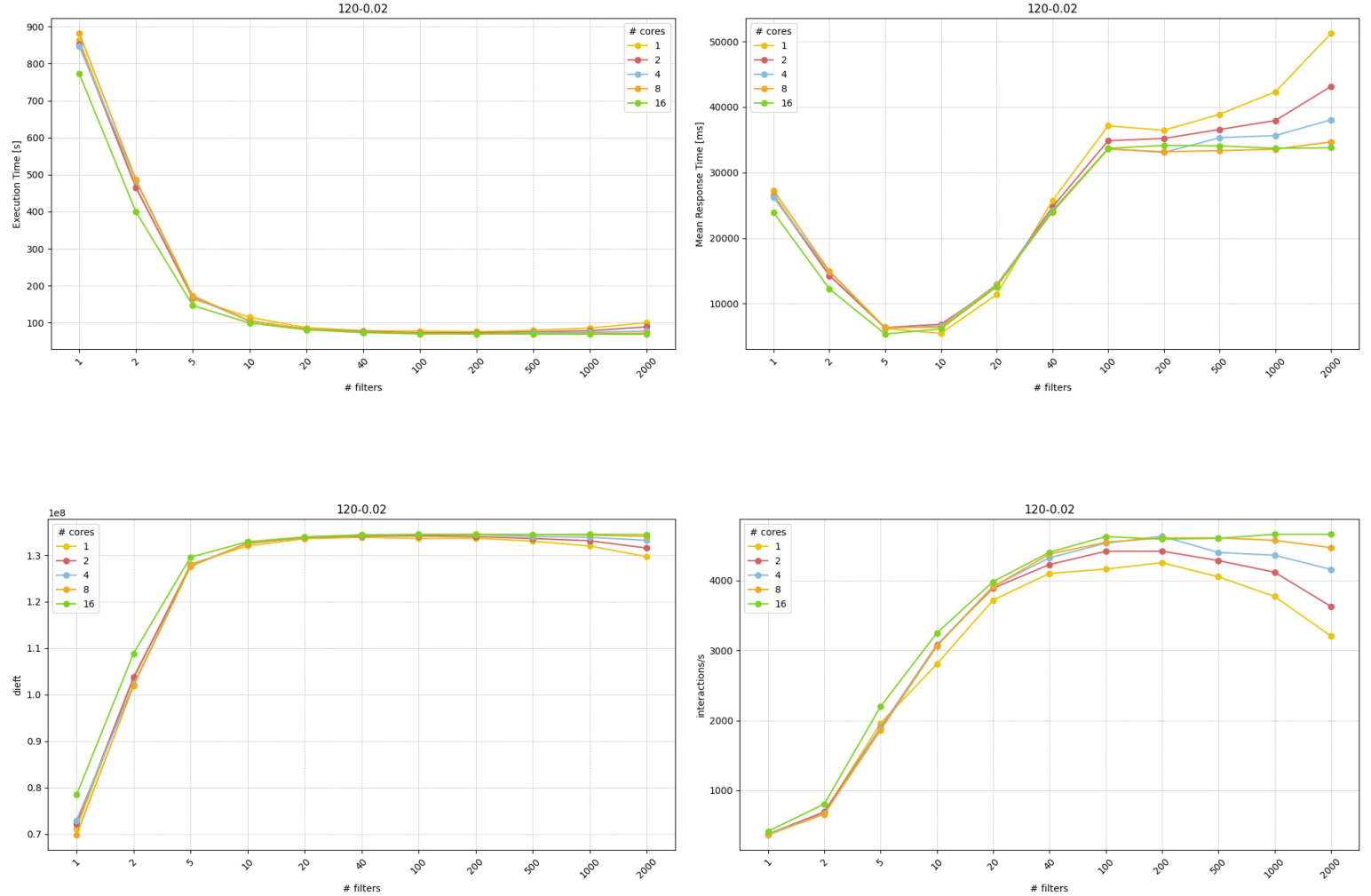


Figure 16: Radial Plots for big stream: 120-0.02

La pregunta es, por qué teniendo menor tiempo de ejecución, el mrt es superior? - Dar una explicación a esto...

### 1.1.2 Differences among the different stream sizes

# 2 Experiments Summary

## 2.1 E1 - NRT

**ONLY CHECKS in these experiments**

Bank Sizes:

- Small: $|Card| = 2000$, $|ATM| = 50$

- Medium: $|Card| = 500000$, $|ATM| = 1000$

Stream Sizes:

| Bank Size | Num Days | Anomalous Ratio | Stream Size | Regular tx | Anomalous tx |
|-----------|----------|-----------------|-------------|------------|--------------|
| Small | 30 | 0.02 (2%) | 39959 | 39508 | 451 1% |
| Small | 60 | 0.02 (2%) | 80744 | 79005 | 1739 |
| Small | 120 | 0.02 (2%) | 160750 | 157756 | 2994 |
| Medium | 7 | 0.03 (3%) | 2428286 | | |
| Medium | 15 | 0.03 (3%) | 4856573 | 4805920 | 50653 |
| Medium | | | | | |
| Big | | | | | |
| Big | | | | | |
| Big | | | | | |

For different core variations, we are going to try different combinations of the system in terms of the number of the maximum number of cards per filter, that consequently will produce an inverse variation in the number of filters of the system.

### 2.1.1 Small Bank Size

| # cards per filter | # filters |
|--------------------|-----------|
| 2000 | 1 |
| 1000 | 2 |
| 400 | 5 |
| 200 | 10 |
| 100 | 20 |
| 50 | 40 |
| 20 | 100 |
| 10 | 200 |
| 4 | 500 |
| 2 | 1000 |
| 1 | 2000 |

- # of times / runs each job = 10.

- Maximum RAM limited to 16GB.

- Run for 1c, 2c, 4c, 8c and 16c.

### 2.1.2 Medium Bank Size

For these experiments, to generate the stream of tx, we needed to simplify this process in order to be able to generate a stream in a feasible amount of time. In particular we used the simplifed version of the `txGenerator.py`: `txGenerator-simplified.py` → with a random ATM-subset instead of a closest to client ATM-subset. Also variation on the transaction distribution times.

- Initial filter configuration setups:

| # cards per filter | # filters |
|:---:|:---:|
| 500000 | 1 |
| 100000 | 5 |
| 50000 | 10 |
| 5000 | 100 |
| 2000 | 250 |
| 1000 | 500 |
| 500 | 1000 |
| 250 | 2000 |
| 100 | 5000 |
| 50 | 10000 |
| 10 | 50000 |

Run with:

- 16GB RAM

- x1 run each job

- x: Run and plots done.

- " ": Not run.

- outMem: out of memory error.

### Stream - 7 Days

| #cores | 1f | 5f | 10f | 100f | 250f | 500f | 1000f | 2000f | 5000f | 10000f | 50000f |
|:---:|:---:|:---:|:---:|:---:|:---:|:---:|:---:|:---:|:---:|:---:|:---:|
| 1 |  | x | x | x | x | x | x | x | x | x |  |
| 2 |  | x | x | x | x | x | x | x | x | x |  |
| 4 |  | x | x | x | x | x | x | x | x | x |  |
| 8 |  | x | x | x | x | x | x | x | x | x |  |
| 16 | x | x | x | x | x | x | x | x | x | x | outMem |

Plots:

- FixedCores: OK

- FixedFilters: OK

- Combined: Running (128GB of RAM) - OTHERWISE REDUCE!

**Stream - 15 Days**

| #cores | 1f | 5f | 10f | 100f | 250f | 500f | 1000f | 2000f | 5000f | 10000f | 50000f |
|--------|----|----|-----|------|------|------|-------|-------|-------|--------|--------|
| 1      |    | x  | x   | x    | x    | x    | x     | x     | x     | x      |        |
| 2      |    | x  | x   | x    | x    | x    | x     | x     | x     | x      |        |
| 4      |    | x  | x   | x    | x    | x    | x     | x     | x     | x      |        |
| 8      | x  | x  | x   | x    | x    | x    | x     | x     | x     | x      |        |
| 16     | x  | x  | x   | x    | x    | x    | x     | x     | x     | x      | outMem |

Plots: $\rightarrow$ Not done so far, only the reduced version explained next. Since for the 7D plots we could already observe that a large number of filters did not produce any advantage, we prefer to reduce the interval of filters in which to show the plots.

- FixedCores: TODO

- FixedFilters: TODO

- Combined: TODO

Based on the results seen (it seems that a really great number of filters is not beneficial), we want to see what happens with a combination of a lower number of filters (like in the experiments for the small bank database):

| # cards per filter | # filters |
|--------------------|-----------|
| 2000               | 1         |
| 1000               | 2         |
| 400                | 5         |
| 200                | 10        |
| 100                | 20        |
| 50                 | 40        |
| 20                 | 100       |
| 10                 | 200       |
| 4                  | 500       |
| 2                  | 1000      |
| 1                  | 2000      |

| #cores | 20f | 40f | 200f |
|--------|-----|-----|------|
| 1      |     |     |      |
| 2      |     |     |      |
| 4      |     |     |      |
| 8      |     |     |      |
| 16     |     |     |      |

| #cores | 2f | 20f | 40f | 200f |
|--------|-----|-----|-----|------|
| 1      |     |     |     |      |
| 2      |     |     |     |      |
| 4      |     |     |     |      |
| 8      |     |     |     |      |
| 16     |     |     |     |      |

**Stream - 7 Days**
Results: DONE
Plots:

- FixedCores: Done

- FixedFilters: Done

- Combined: TODO, increase RAM memory to do it, higher than 64GB...

**Stream - 15 Days**
Results: Done
Plots:

- FixedCores: Obtaining

- FixedFilters: TODO → not for the moment

- Combined: TODO, increase RAM memory to do it, higher than 64GB... → not for the moment

## 2.2 NEW: Reduced comparison

Plots for the medium gdb, with a reduced number of filters (1f...200f) and including the baseline, to see better what happens.
Obtaining baselines run

- 7D: 1c, 2c, 4c, 8c, 16c

- 15D: 1c, 2c, 4c, 8c, 16c

Obtaining plots - only fixed cores plots

- 7D:

- 15D: