ForEach Activity parallelism(how ADF supports it)

Last updated by | Ranjith Katukojwala | Mar 7, 2023 at 11:35 AM PST

Contents

Issue

Customers report that they are noticing reduced DOP (degree of parallelism) with foreach activity.

The goal of this TSG is to understand how the Foreach works covering parallelism and queue through examples.It is very common to have customers complaining about the Foreach activity performance/parallelism. Usually, customer do not observe the max parallelism they set with batch count property.

The degree of parallelism in ForEach is actually max degree of parallelism. We cannot guarantee a specific number of executions happening at the same time, but this parameter will guarantee that we never go above the value that was set. You should see this as a limit, to be leveraged when controlling concurrent access to your sources and sinks.

Known Facts about Foreach

1. Foreach has a property called batch count(n) where default value is 20 and the max is 50
2. The batch count, n, is used to construct n queues. Later we will discuss some details on how these queues are constructed.
3. Every queue runs sequentially, but you can have several queues running in parallel.
4. The queues are pre-created. This means there is no rebalancing of the queues during the runtime.

Consequences from the facts above

1. At any time, you have at most one item being process per queue. This means at most n items being processed at any given time ([one item per queue]*n).
2. The foreach total processing time is equal to the processing time of the longest queue. This means that the foreach activity depends on how the queues are constructed.
3. Imagine, you have set a foreach activity with batch count of 2 to process 4 copy activities of duration 1min, 1min, 2min and 2min.
    - Q1=[1,1] and Q2=[2,2] means processing Q1 in 2min, Q2 in 4min and Foreach in 4min.
    - Q1=[1,2] and Q2=[1,2] means processing Q1 in 3min, Q2 in 3min and Foreach in 3min.

Foreach performance through ADF samples

To analyse the performance of the foreach we will be implementing a pipeline where a foreach activity will process 25 wait activities. The pipeline has an array parameter with the 25 durations for the wait activities. We will do two tests with the same array, but with different orders. Please, find the pipeline jsons at the bottom of this tsg.

Fast test

array: [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,5,5,5,5,5]

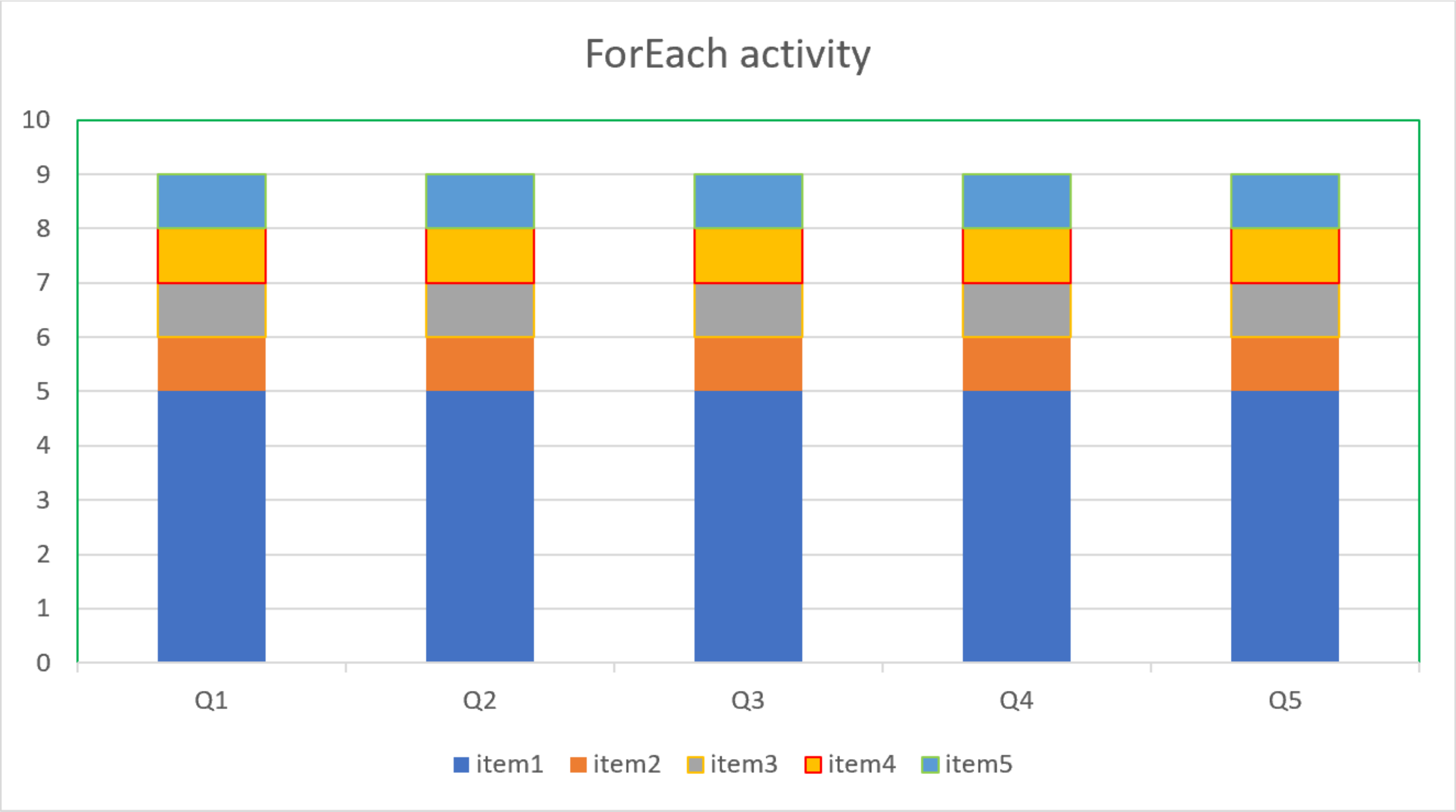
Batch count: 5

Queue size: 5

Pipeline total time execution is 9 minutes.

We have parallelism always equal to 5.

The height of each queue is the queue total processing time



Slow test

array: [1,1,1,1,5,1,1,1,1,5,1,1,1,1,5,1,1,1,1,5,1,1,1,1,5]

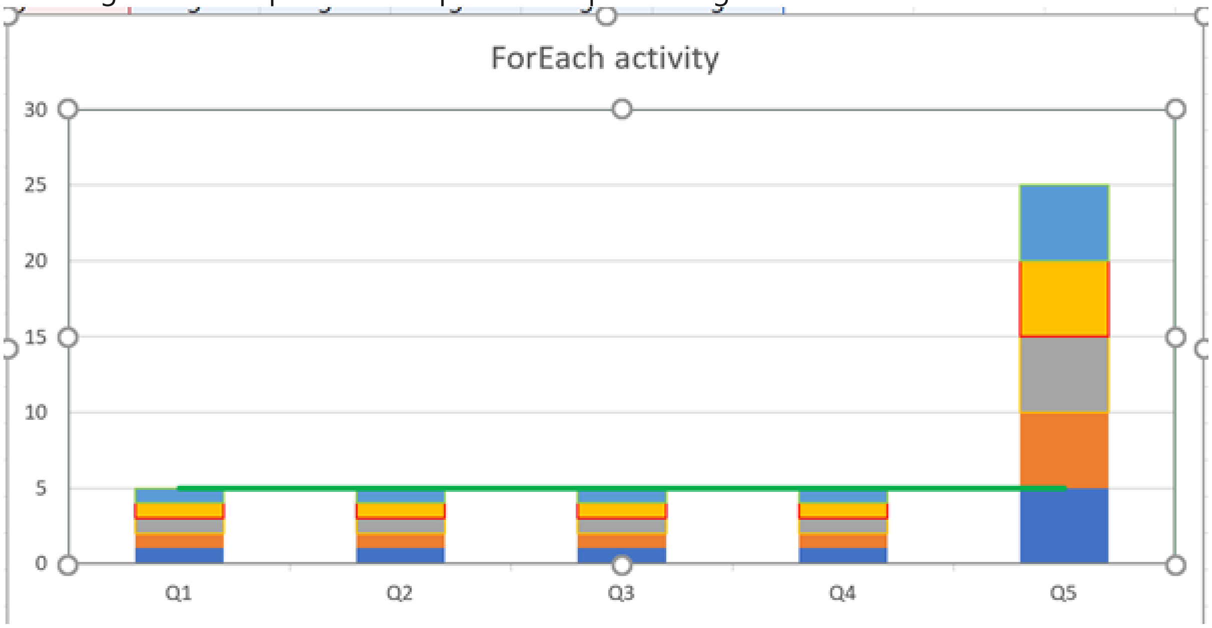
Batch count: 5

Queue size: 5

Pipeline total time execution is 25 minutes.

During the first five minutes we have max parallelism (5), and then it runs sequentially.

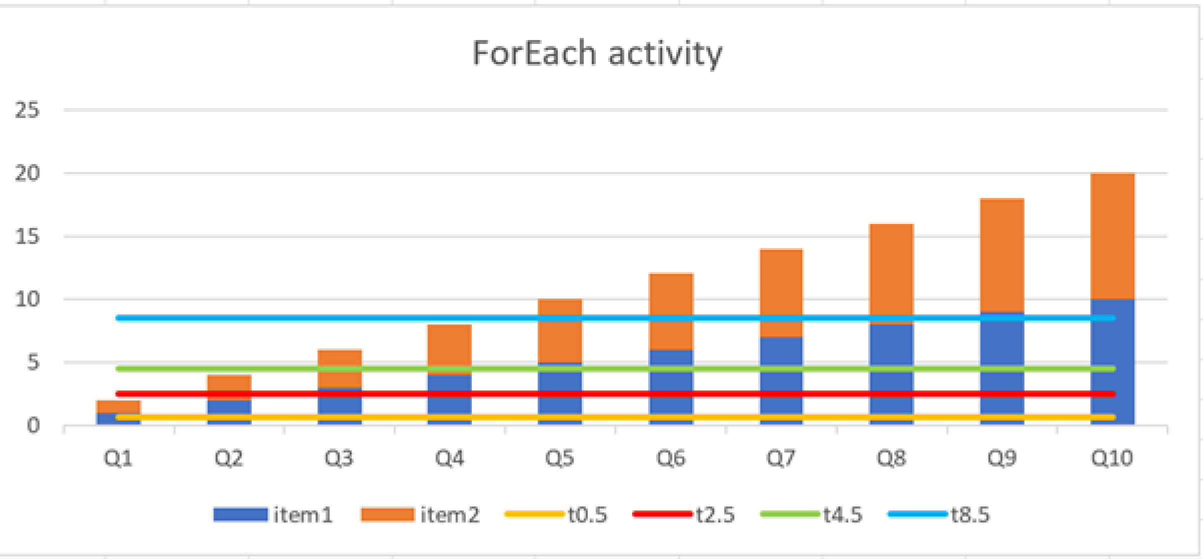
The height of each queue is the queue total processing time



Foreach classic customer scenario

Customer sees the parallelism decreasing over time. Customer has 20 items and set batch count equal to 10. The height of each queue is the queue total processing time.

- At t =0.5, customer observes 10 parallel runs
- At t=2.5, customer observes 9 parallel runs
- At t=4.5, customer observes 8 parallel runs
- At t=8.5, customer observes 6 parallel runs.



Further information

Queues distribution(very important)

Please, do not share this info with customers. You start enumerating the elements from 1 to batch count. Once the batch count is reached you restarted the enumeration till all elements have been enumerated. This enumeration is then used to build the queues.

Example

- Let's apply the process above described to the array [1,2,3,4,5,1,1,1,1,5,1,1,1,1,5,1,1,1,1,5,1,1,1,1,5] considering batch count of 5
- We would have the following enumeration array [1,2,3,4,5, 1,2,3,4,5, 1,2,3,4,5, 1,2,3,4,5, 1,2,3,4,5]
- Every queue will have all items matching enumeration element value.
- Q1=[1,1,1,1,5], Q2=[2,1,1,1,5], Q3=[3,1,1,1,5], Q4=[4,1,1,1,5] and Q5=[5,1,1,1,5]

How to improve foreach performance?

- Taking in consideration the way the queues are constructed customer can improve the foreach peformance by setting multiple foreaches where each foreach will have items with similar processing time. This will ensure that long runs are processed in parallel rather sequentially.

SetVariable Inside Foreach activity

Customer should not use SetVariable inside a foreach that runs in parallel. Please , check here (https://supportability.visualstudio.com/AzureDataFactory/_wiki/wikis/AzureDataFactory/394715/SetVariable-inside-ForEach) for further details.

ADF pipeline jsons

```
{
    "name": "demo_fast_foreach",
    "properties": {
        "description": "order of items may have impact in the performance",
        "activities": [
            {
                "name": "ForEach1",
                "type": "ForEach",
                "dependsOn": [],
                "userProperties": [],
                "typeProperties": {
                    "items": {
                        "value": "@pipeline().parameters.myitems",
                        "type": "Expression"
                    },
                    "batchCount": 5,
                    "activities": [
                        {
                            "name": "Wait1",
                            "type": "Wait",
                            "dependsOn": [],
                            "userProperties": [],
                            "typeProperties": {
                                "waitTimeInSeconds": {
                                    "value": "@mul(item(),60)",
                                    "type": "Expression"
                                }
                            }
                        }
                    ]
                }
            }
        ],
        "parameters": {
            "myitems": {
                "type": "array",
                "defaultValue": [
                    1,
                    1,
                    1,
                    1,
                    1,
                    1,
                    1,
                    1,
                    1,
                    1,
                    1,
                    1,
                    1,
                    1,
                    1,
                    1,
                    1,
                    1,
                    1,
                    1,
                    5,
                    5,
                    5,
                    5,
                    5
                ]
            }
        },
        "annotations": []
        }
    }
```

```
{
    "name": "demo_slow_foreach",
    "properties": {
        "description": "order of items may have impact in the performance",
        "activities": [
            {
                "name": "ForEach1",
                "type": "ForEach",
                "dependsOn": [],
                "userProperties": [],
                "typeProperties": {
                    "items": {
                        "value": "@pipeline().parameters.myitems",
                        "type": "Expression"
                    },
                    "batchCount": 5,
                    "activities": [
                        {
                            "name": "Wait1",
                            "type": "Wait",
                            "dependsOn": [],
                            "userProperties": [],
                            "typeProperties": {
                                "waitTimeInSeconds": {
                                    "value": "@mul(item(),60)",
                                    "type": "Expression"
                                }
                            }
                        }
                    ]
                }
            }
        ],
        "parameters": {
            "myitems": {
                "type": "array",
                "defaultValue": [
                    1,
                    1,
                    1,
                    1,
                    5,
                    1,
                    1,
                    1,
                    1,
                    5,
                    1,
                    1,
                    1,
                    1,
                    5,
                    1,
                    1,
                    1,
                    1,
                    5,
                    1,
                    1,
                    1,
                    1,
                    5
                ]
            }
        },
        "annotations": []
    }
}
```

- **Icm References:**
  - https://icm.ad.msft.net/imp/v3/incidents/details/108126341/home
  - https://icm.ad.msft.net/imp/v3/incidents/details/128086182/home
  - The length of duration can depend on partner service queue status also (see example ICM https://icm.ad.msft.net/imp/v3/incidents/details/108126341/home where customer had latency against Machine Learning service)
- **Author:** negome
- **Reviewer:** grorcai
- **Keywords:**

How good have you found this content?

🙂 🙁