

# ROS2 EXECUTOR BENCHMARKING

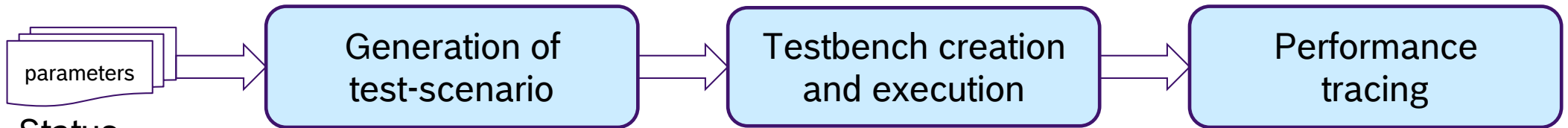
JAN STASCHULAT, RALPH LANGE

# ROS 2 Execution

## Executor Benchmarking



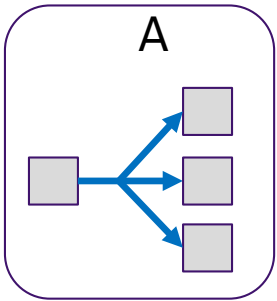
- ▶ How to quantify performance of ROS2 Executors(standard rclcpp-, static rclcpp-, LET-Executor)?
  - ▶ Existing applications (Nobleo) are too complex to investigate root causes of high CPU-load
  - ▶ rclcpp/static rclcpp Executor are not comparable with LET-Executor (intra-process vs. DDS)
  - ▶ Performance evaluation with docker-stats (CPU-load) not reliable
- ▶ Approach
  - ▶ Simple configuration of benchmarks( few parameters only, like number of topics, message size, rate etc.)
  - ▶ Automatic generation of test setup
  - ▶ Performance evaluation with tracing tools (measures total time (ms) per function)



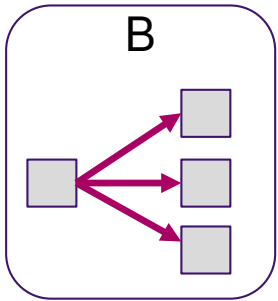
- ▶ Status
  - ▶ Setup of entire workflow finished
  - ▶ Ready to perform experiments

# Benchmarking ROS2 Executor

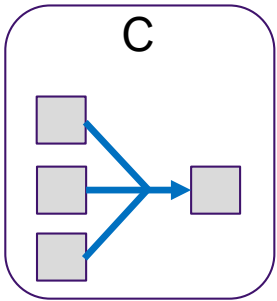
## Topology types



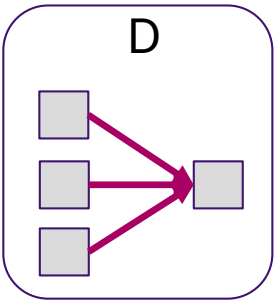
- ▶ Pub: 1
- ▶ Sub: n
- ▶ Topics: 1



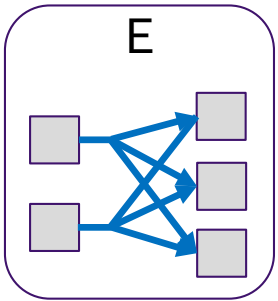
- ▶ Pub: 1
- ▶ Sub: n
- ▶ Topics: n



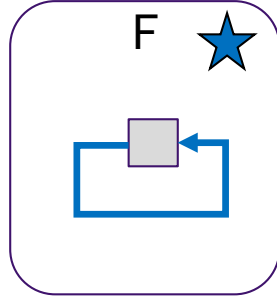
- ▶ Pub: n
- ▶ Sub: 1
- ▶ Topics: 1



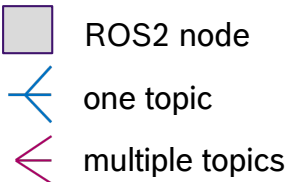
- ▶ Pub: n
- ▶ Sub: 1
- ▶ Topics: n



- ▶ Pub: n
- ▶ Sub: m
- ▶ Topics: n



- ▶ Pub: n
- ▶ Sub: m
- ▶ Topics: n



- ▶ Pub: number of publishing nodes
- ▶ Sub: number of subscribing node
- ▶ Topics: number of topics



One node with n publishers, m subscribers and n different topics (same as E but one node)

# Benchmarking ROS2 Executor

## Parameters

- ▶ Configuration parameters:

- ▶ Number of topics (int)
- ▶ Topology type (Type A - E)
- ▶ Message rate (Hz)
- ▶ Message size(String message: #characters)
- ▶ Communication type (Intra-Process, DDS)
- ▶ Total number of messages (send/received)

- ▶ Communication type: Intra-Process / Intra-Process (within DDS) / DDS

- ▶ Scope: only for publisher-side, only for subscriber-side or every node

**=> With number of topics and topology type all relevant use-cases can be generated,**

**=> With communication type intra-process/DDS is configured**

# Benchmarking ROS2 Executor

## API notes

- ▶ Communication type: Intra-Process / DDS
  - ▶ configuration on process-level (because topic name need to be known for each pub/sub)
- ▶ Topology type
  - ▶ Setup launch-file (python) which creates a list of parameters per process; Example Topology A:
    - ▶ (node1: pub: topic\_1, sub: - )
    - ▶ (node2: pub: topic\_1, sub: - )
    - ▶ (node3: pub: topic\_1, sub: - )
    - ▶ (node4: pub: - , sub: topic\_1)
- ▶ Multiple nodes with publishers: (Problem: every node needs a timer\_callback, in which the messages are published at the specified rate)
  - ▶ Create as many timers as number of nodes
  - ▶ One function timer\_callback, which publishes all topics for all nodes, compares parameter ptr\_timer with global mapping of ptr\_timer => Node => Publisher\_i
  - ▶ Selects the publishers and publishes the message