

# *MARV Robotics - Tool for Large Data Storage, Analysis, Viewing and Testing*

Dejan Pangercic, Faraday Future Inc.

Daniel Di Marco, Deepfield Robotics

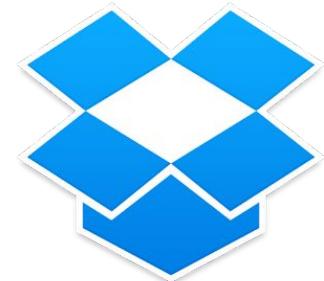
Florian Friesdorf, Ternaris

Marko Durkovic, Ternaris

# Motivation

## Where is my bag?

```
dejanpangercic@marv:~$ locate *.bag
/home/dejanpangercic/Downloads/gps_track_gps_fix_orientation.bag
/home/dejanpangercic/Downloads/gps_track_leica_trimble_imu.bag
/home/dejanpangercic/Downloads/navsatfix-example1.bag
/home/dejanpangercic/Downloads/navsatfix-example2.bag
/home/dejanpangercic/Downloads/recording_Ihinger-hof_Plot 3__2016-03-02-09-59-
57-926422_cambB.bag
/home/dejanpangercic/Downloads/test_2015-02-05-12-59-06_0.bag
/home/dejanpangercic/Downloads/test_2015-02-05-12-59-08_1.bag
/home/dejanpangercic/Downloads/test_2015-02-05-12-59-10_2.bag
/home/dejanpangercic/Downloads/zr_as20150818_day07_jai_parzelle02_2015-08-30-16-
06-03_0.bag
/home/dejanpangercic/Downloads/zr_as20150818_day12_jai_parzelle02_2015-09-04-09-
39-43_0.bag
dejanpangercic@marv:~$
```



# Motivation

At some point it is  
unmanageable



BagBunker



## Summary

Fileset Count	File Count	Total Size	Duration (s)
13372	16051	30.43 TB	1689791.44954

Datasets (13372 found)

bulk operations

# Motivation

## What if?



+

← → C <https://photos.google.com/albums> ⭐

☰ Albums  + ⌂

Shared People Places Things Videos Collages

...

# MARV - ROSbag app

## Quick demo

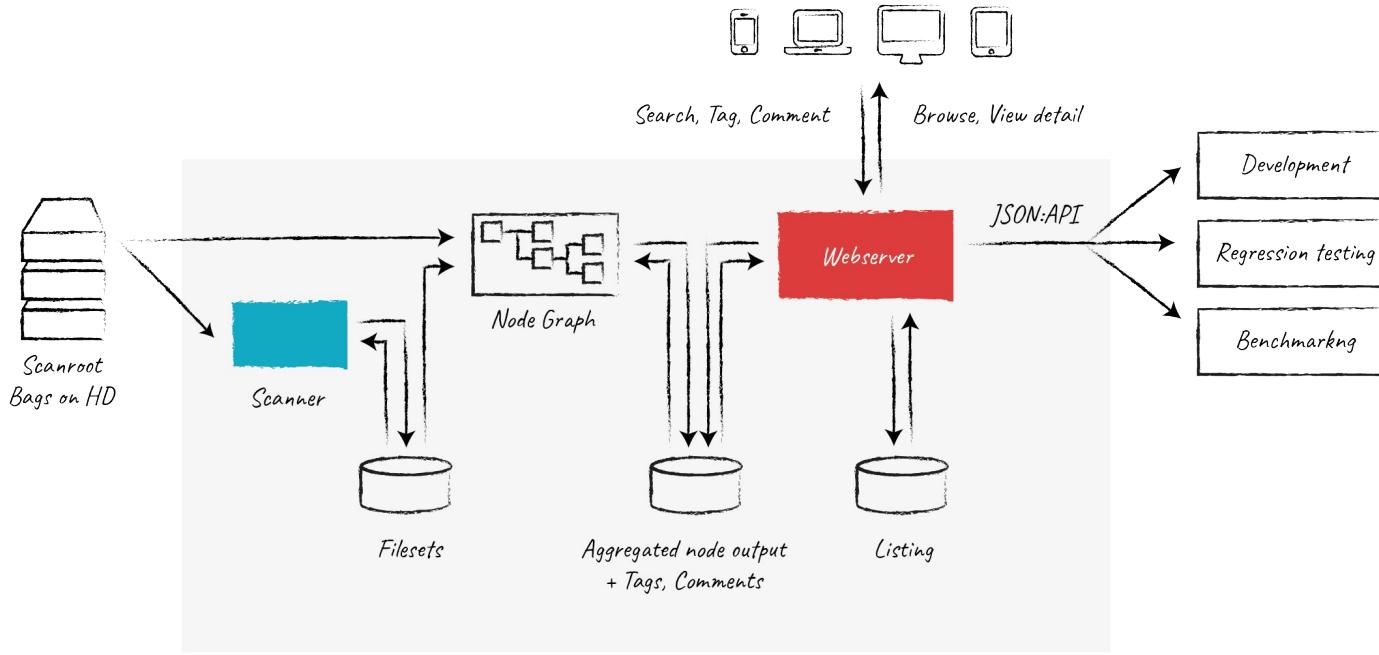
- Overview
- Filters
- Nodes
- Tagging/commenting
- API
- Downloading
- Full text search

<https://ternaris.com/marv-robotics/demo/>

The screenshot shows a web-based application interface for managing ROSbag files. At the top, there's a header bar with the URL <https://ternaris.com/marv-robotics/demo/>. Below the header is a dark navigation bar with the "MARV ROBOTICS" logo. The main content area is divided into sections:

- Summary**: Shows statistics: 9 filesets, 3.93 GB size, and 07m 15s duration.
- Data sets (9 found)**: A list of data sets with checkboxes and various actions like tag, comment, copy file paths, and copy URLs.
  - gps\_track\_\_gps\_fix\_orientation
  - gps\_track\_\_leica\_trimble\_imu
  - navsatfix-example2
  - navsatfix-example1
  - recording\_lhinger-hof\_Plot 3\_2016-03-02-09-59-57-926422\_camB
  - diagnostics\_agg
  - zr\_as20150818\_day12\_jai\_parzelle02
  - zr\_as20150818\_day07\_jai\_parzelle02
  - test

# Overview diagram



# How to work with MARV?

## Install

**TERNARIS**

About MARV Robotics Contact

You create robots that will shape the world of tomorrow? Supercharge your workflows with MARV Robotics, a powerful and extensible data management platform.

Try out the demo ➤

<https://ternaris.com/marv-robotics/>

Rich dynamic web interface

MARV Robotics allows you to make vast amounts of data accessible by web. Your team can access all data and get an overview in a dynamic web GUI. Filter and find information, comment on and tag it for future reference.

**MARV ROBOTICS**

Summary  
9 files 3.93 GB 0m 15s

Data sets (9 found)

Name	Size	Status	Tags	Start time	End time	Duration
gei_reach_gte_0_annotation	249.71 KB	13/08/13	2013/08/13	2013/09/30	21%	
gei_reach_lte_minimal_mju	1.80 MB	20/08/13	2013/08/13	2013/09/30	42%	
remote_reach	409.90 KB	19/07/13	2013/07/21	2013/07/21	62m 40s	
rendezvous	171.11 KB	20/07/13	2013/07/21	2013/07/21	42%	
rendezvous_hol_Plot_1_2013-03-09-09-57-	49.63 MB	20/08/13	2013/07/02	2013/07/02	0%	
SO3421_cmuB	1.00 MB	10/09/13	2013/09/13	2013/09/13		
logistics_RSE	33.32 KB	20/08/13	2013/07/28	2013/07/28	4%	

UI | Config

<https://github.com/ternaris/marv-robotics>

ternaris / marv-robotics

Code Issues Pull requests Projects

No description or website provided.

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request

chaoflow MARV Robotics release 2.0.0b1 ... Latest commit 177c60e an hour ago

marv-robotics MARV Robotics release 2.0.0b1 an hour ago

README.rst MARV Robotics release 2.0.0b1 an hour ago

README.rst

## MARV Robotics (beta)

MARV Robotics is an extensible data management platform for robot logs. New robot logs are found by a scanner and configured nodes are run to extract, filter and process data from them.

The robot logs are visualized in a web-based application that features a listing view with filters and summary, and detailed views of individual log files.

So far MARV Robotics supports the Robot Operating System (ROS) Bag format used by systems running ROS to record sensor data and system state.

### Robot Operating System (ROS)

MARV Robotics is meant to support all alive ROS releases. Currently these are indigo, jade, and kinetic -- if you encounter difficulties, please report back! The following instructions are written for ROS kinetic, adjust to your needs.

# How to work with MARV?

## Configure

- Scan root of bag files
- Fileset type
- Listing Columns
- Filters
- Details
- Details for nodes

[https://github.com/ternaris/marv-robotics/blob/master/marv-robotics/marv\\_robotics/marv\\_profile/marv.conf.in](https://github.com/ternaris/marv-robotics/blob/master/marv-robotics/marv_robotics/marv_profile/marv.conf.in)

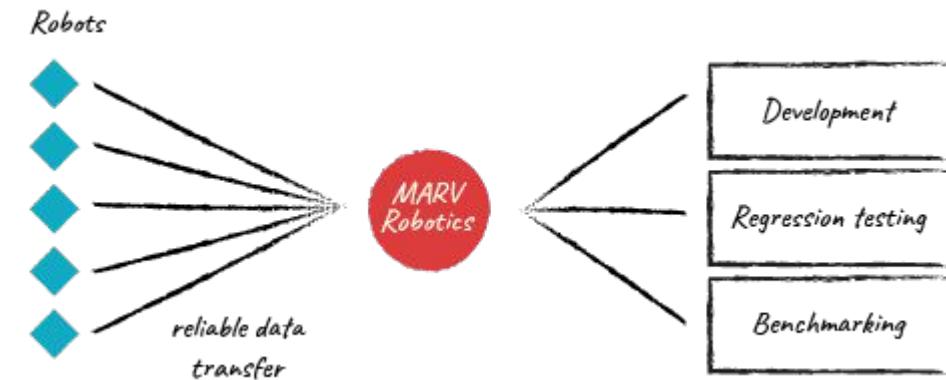
```
# [auth]
# authfunc = marv.app:authenticate
# userdb = ./userdb

[collection]
scanroot = ./scanroot
fileset = marv_robotics.bagset
nodes =
    marv.nodes:md5
    marv_robotics.bagset:messages
    marv_robotics.nodes.bagset_name:bagset_name
    marv_robotics.nodes.bagmeta:bagmeta
    marv_robotics.nodes.camera_frames:camera_frames
    marv_robotics.nodes.diag_count:diag_count
    marv_robotics.nodes.fulltext:fulltext
    marv_robotics.nodes.gnss:gnss_plots
    marv_robotics.nodes.osm:navsatfix
    marv_robotics.nodes.osm:geo_json_trajectory
detail_title = bagset_name
detail_summary =
    size      | filesize   | fileset      | sum(x.size for x in fileset.files)
    files     | string     | fileset      | len(fileset.files)
    start_time | datetime  | bagmeta     | bagmeta.start_time
    end_time   | datetime  | bagmeta     | bagmeta.end_time
    duration    | timedelta | bagmeta     | bagmeta.duration
    robot      | string     | deepfield_meta | deepfield_meta.robot_name
    use_case    | string     | deepfield_meta | deepfield_meta.use_case
details =
    bag_meta
    topics
    diagnostics
    camera_frames
    gnss_plots
    trajectory
```

# How to work with MARV?

## Record/Upload/Enjoy

- Instructions in [README](#)
- Copy data to the scanroot (we recommend rsync)
- Scan filesets (partial bags are aggregated)
- Query hashes of known files and delete successfully synced ones
- Run nodes
- View/Analyse/Debug :)



# How to work with MARV?

## How to write nodes?

- Image lapse node
- Nodes are written in Python
- Write logic only, let marv do the rest

[https://github.com/ternaris/marv-robotics/blob/master/marv-robotics/marv\\_robotics/nodes/camera\\_frames.py](https://github.com/ternaris/marv-robotics/blob/master/marv-robotics/marv_robotics/nodes/camera_frames.py)

```
@marv.node()
@marv.input('bagmeta')
@marv.input('messages', filter=['*:sensor_msgs/Image'])
@marv.param('image_width', help="Scale to image_width, keeping aspect ratio")
@marv.param('max_frames', help="Maximum number of frames to extract")
def camera_frames(bagmeta, messages, image_width=320, max_frames=50):
    """Extract camera frames from sensors_msgs/Image streams

    Images are scaled to image_width while keeping the aspect ratio
    and a maximum of max_frames equidistantly spread frames is
    extracted from each stream.
    """

    # Message counts and desired intervals for all topics. All topics,
    # as we cannot know which topics a user configured.
    message_counts = {k: v['message_count'] for k, v in bagmeta.topics.items()}
    intervals = {topic: int(math.ceil(message_count / max_frames))
                 for topic, message_count in message_counts.items()}

    # Keep track of per-topic message indices and images generated
    msg_indices = defaultdict(int)
    images = defaultdict(list)
    for topic, msg, _ in messages:

        # Only use messages in desired per-topic intervals
        idx = msg_indices[topic]
        msg_indices[topic] = idx + 1
        if idx % intervals[topic]:
            continue

        try:
            img = imgmsg_to_cv2(msg, "rgb8")
        except:
            import traceback
            marv.log_error('On topic %r: %s', topic, traceback.format_exc())
            continue
```

# How to work with MARV?

## Integrate MARV in your system

- RESTful API
- JSON response

```
import requests
import json

url = 'http://127.0.0.1:8000/marv/api/2/listing'
headers = {'Accept': 'application/vnd.api+json'}

filters = [dict(name='name', op='like', val='%gps%')]
params = {'filter[objects]': json.dumps(filters)}

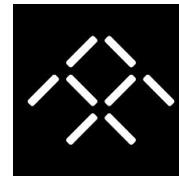
response = requests.get(url, params=params, headers=headers)
assert response.status_code == 200
print(response.json())
```

# Demo, licensing and support

- Demo: <https://ternaris.com/marv-robotics/demo>
- Community edition: AGPLv3 license
- Enterprise edition: <https://ternaris.com/marv-robotics>

Contact&Support: <https://ternaris.com>

# roboception



## Selected Enterprise Edition Features:

- Browser-based download of partial robot logs
- Streaming of partial robot logs
- Browser-based video playback of camera data
- User-management and access control
- Regular updates
- Priority support (review and development of custom nodes and widgets, integration in your workflows)
- Embeddable into your products