Live Migration of a 3D Flash LiDAR System between two Independent Data Processing Systems with Redundant Design

Philipp Stelzer*, Sebastian Reicher*, Georg Macher*, Christian Steger*, and Raphael Schermann*†
*Graz University of Technology, Graz, Austria
{stelzer, reicher, georg.macher, steger}@tugraz.at
†Infineon Technologies Austria AG, Graz, Austria
{raphael.schermann}@infineon.com

***Abstract*—Self-driving and self-flying vehicles have the ability to drive respectively fly independently without the intervention of an operator. For this purpose, these vehicles need sensors for environment perception and data processing systems, which are safety-critical, to process the obtained raw data from these sensors. However, if such safety-critical systems fail, this can have fatal consequences and can affect human lives and/or the environment, especially in the case of highly automated vehicles. A total failure of these systems is one of the worst scenarios in an automated vehicle. Therefore, such safety-critical systems are often designed redundantly in order to prevent a total failure of environment perception. In order to ensure that the operation of the vehicle can continue safely, however, the live migration from one system to the other must be carried out with as little downtime as possible. In our publication, we present a concept for a 3D Flash LiDAR live migration between two independent data processing systems with redundant design. This concept provides a solution for highly automated vehicles to remain fail-operational in case one of the redundant data processing systems fails. The results obtained from the implemented concept, without specifically addressing performance, are also provided to demonstrate feasibility.**

***Index Terms*—3D Flash LiDAR, automated vehicles, live migration, safety, fail-operational**

I. INTRODUCTION

Vehicles, whether on the ground or in the air, are becoming increasingly automated. Already today, there are scenarios in which vehicles perform tasks autonomously [1, 2]. This means that these vehicles are dependent on the data from their environment perception sensors, at least while they are independently carrying out their specified tasks. Should the environment perception fail, then this could have fatal consequences for humans in the vicinity and the environment [3]. Therefore, in addition to reliable and diverse sensors, it is also important that data processing systems are reliable [4]. However, as it must always be expected that a system failure may occur, there must be strategies for highly automated vehicles to maintain the functionality, at least in a reduced form, of holistic safety-critical systems. In order to further research and improve the necessary environment perception and reliability of sensors and systems in highly automated vehicles, the European Union has already launched various projects. These include the PRogrammable sYSTems for INtelligence in automobilEs (PRYSTINE) project [5], where the consortium aimed to achieve a Fail-operational Urban Surround perceptION (FUSION) based on robust Radio Detection And Ranging (RADAR), Light Detection and Ranging (LiDAR) and camera environment perception sensors. At the drone level, for example, the Airborne Data Collection on Resilient System Architectures (ADACORSA) project [6], a representative cover is illustrated in Figure 1, was launched to design and develop, among other things, resilient architectures for flight systems. The resilience of holistic systems is often achieved through redundant design. In this way, functionality can be ensured even if individual components fail. This approach is also taken up in our publication. In our publication, we propose a concept that allows environment perception to be maintained despite a data processing system failure by live migration of a sensor to another redundantly designed data processing system.

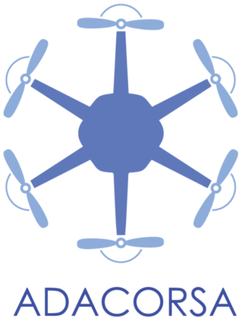


Fig. 1. Airborne Data Collection on Resilient System Architectures [6].

With our contribution we:

- create a possibility to continue the raw data processing of environment perception sensors,
- enhance safety by enabling raw data processing to continue after the failure of the currently responsible data processing system and
- achieve a fail-operational behaviour due to the continuous raw data processing of the sensor data for environment perception.

The remainder of the publication is structured as follows. An overview of related work in the automotive and aerial domain is given in Section II. The novel concept of a 3D Flash LiDAR live migration from one independent data processing system to another redundant data processing system will be introduced in Section III, and the achieved results, including a short discussion, will be provided in Section V. A summary and short discussion of the findings will conclude this publication in Section VI.

II. RELATED WORK

In this section, we refer to related work in the automotive and aerial domain regarding migration, environment perception sensors and proofed safety concepts.

A. Migration

Several concepts for migration in environments including ROS are pointed out in order to clarify the sense of using ROS and at least to be able to show the feasibility. By Khan et al. [7], for instance, a concept of live migration in the automotive domain was investigated. Proposing a framework of live migration and container-based virtualization for the vehicular ad-hoc network, this publication marks out the challenges and the current trend of live migrations. However, this conceptional framework of virtualization and live migration in Vehicle-to-Infrastructure (V2I) is illustrated in Figure 2. For the migration and creation of containers, Checkpoint and Restore In Userspace (CRIU) and Linux Containers (LXC) are being utilized. In another publication [8], Cano et al. looks into dynamic process migration in heterogeneous Robot Operating System (ROS) based environments. By introducing the AnyScale concept, which dynamically allocates tasks to appropriate substrates, system performance can be optimized. This is done by migrating other tasks relying on current resource and performance parameters. With regard to aerial applications, Jeong et al. propose in their publication [9] a platform utilizing the capabilities of ROS to ensure reliability in completing missions of a service drone. Additionally, it is also possible to update an application being still in flight. Extending the observation into the aerial domain, An et al. [10] demonstrates a seamless virtualized controller migration for drone applications, as symbolically illustrated in Figure 3. In this study, a seamless migration scheme is set up in such a way, that a suspension of the drone control is not necessary. This is done by migrating a virtualized drone controller to an edge node that is close to its associated drone. Regarding live video streaming with a remote-controlled drone, Wamser et al. [11] use dynamic cloud service placement in order to manage real-time video streaming and real-time control commands. The objective is to enhance the quality of experience of users for streaming services and real-time control.

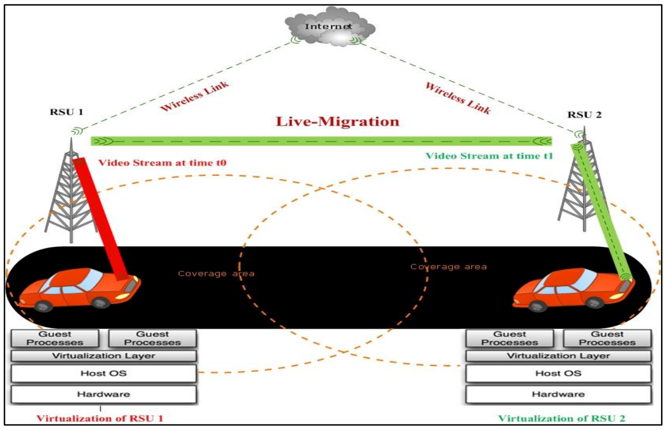


Fig. 2. Conceptual framework of virtualization and live migration in V2I [7].

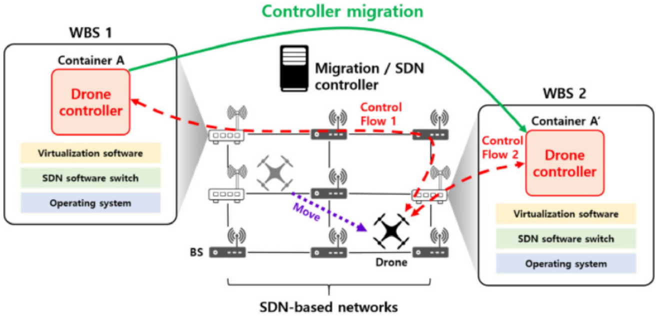


Fig. 3. Seamless migration process in a virtualized edge-computing infrastructure [10].

B. Environment Perception

To perceive the environment, in automated vehicles are often diverse sensor technologies like RADAR, LiDAR and vision cameras are used [5]. Regarding the use of LiDAR sensors in autonomous driving vehicles itself, García et al. [12] and Ratshidaho et al. [13] give examples of possible use cases in this area. Obstacle detection is one of the most important aspects in autonomous driving. García et al. [12] are using ToF (Time of Flight) cameras to study different descriptors and classifiers. Same kind of cameras are also used by Ratshidaho et al. [13] to estimate ego-motion. This is the process of estimating a pose relative to a initial pose using the camera's image sequence. Information regarding localization can be acquired.

C. Redundancy

The concept of redundant design is a proven approach and often applied in safety-critical systems. This involves the redundant design of both, sensors and data-processing systems, depending on the application. Regarding redundancy of sensors in the space of automated driving vehicles, Berk et al. [14], for instance, addressed this topic in order to analyse the reliability of sensor perception. This is done by defining a likelihood function for redundant binary sensor data using no reference truth. By proposing a Gaussian copula, dependent sensor errors can be modelled. To increase reliability, sensor systems are often designed redundantly. Safronov et al., for instance, describe in their publication [15] how they include flight control computer state synchronisation in a mission execution system to achieve redundancy in flight control for aerial vehicles. In most of the fault tolerant control systems redundancy is required as mentioned by Huang et al. [16] in their publication. This is intended to fulfil safety and reliability requirements in an uncomplicated way. As shown by the steer-by-wire example of Huang et al. [16], there are various forms of redundancy.

III. 3D Flash LiDAR Live Migration Concept

This section introduces the new concept for an environment perception sensor live migration using a 3D Flash LiDAR. The proposed setup in this publication is limited to the processing and display of 3D Flash LiDAR data and a live migration between two independent redundant data processing systems. As can be seen in Figure 4, the 3D Flash LiDAR system is connected to a safety controller via USB, which forwards the data to host 1 and host 2 via TCP/IP. On host 1 and host 2 the Robot Operating System (ROS) application is running. The data of the 3D Flash LiDAR system are displayed in the ROS. The safety controller monitors the current status of the active host (e.g. host 1). In the case, that the active host cannot continue to process incoming 3D Flash LiDAR data (e.g. anomaly in the memory area due to the execution of defective programs), the safety controller triggers a live migration to the redundant host (e.g. host 2). The current status is transferred from host 1 to host 2 and host 2 takes control of the 3D Flash LiDAR system. A sequence diagram shown in Figure 5 visualizes the chronology of these processes. After the live migration from host 1 to host 2, the experiment is completed. The sequence diagram shows a theoretical endless continuation of the experiment, by execution of another migration from host 2 to host 1. This presumes a fixing of the initial problem of host 1. At this point, the experiment finds itself in the initial starting situation and could theoretically repeat itself. The main objective of this experimental setup is to ensure a fail-operational behaviour by preventing a total failure of the 3D Flash LiDAR system. As described, the live migration ensures the continuation of the system's working capabilities on another host. Ideally, this procedure executes in such a way that the downtime during a live migration is not even noticeable to the end user. The success of this so called seamless live migration is dependent on different factors. First of all, the detection of the running host's failure has to occur sufficiently early by the safety controller. Beyond that, the live migration itself has to perform fast enough to not interfere with the performance of the system's task. In our use case, the 3D Flash LiDAR system produces a high amount of data output in the form of 3D point clouds. If the live migration cannot keep up with the huge exchange of data, downtime is inevitable. In the case of safety-critical systems, this resulting downtime can lead to drastic consequences. In order to combat that, a possible solution is to reduce the amount of exchanged data by decreasing the resolution and/or the frame rate of the 3D Flash LiDAR system. This action certainly influences the efficiency of the system, due to that, a acceptable trade-off has to be found.

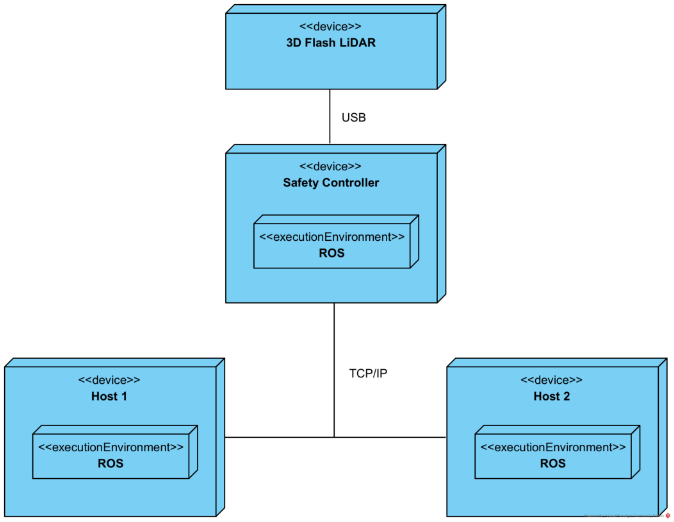


Fig. 4. Deployment diagram of the proposed, fail-operational system.

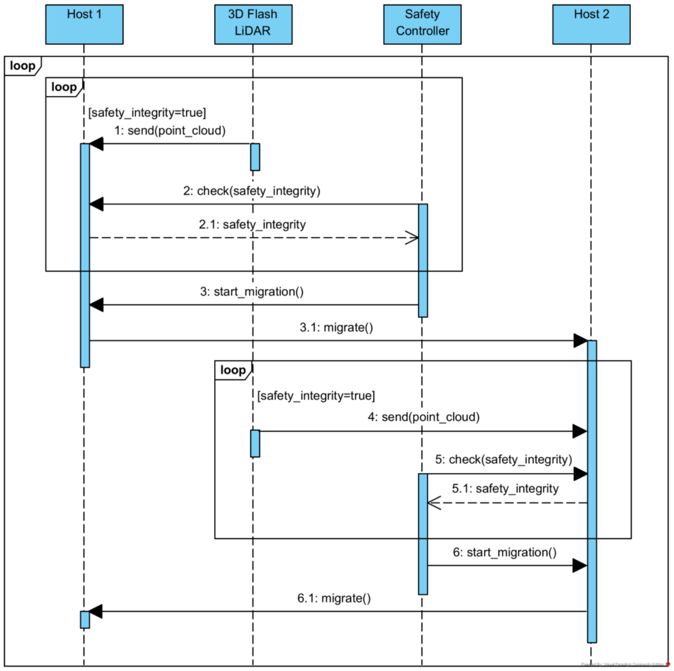


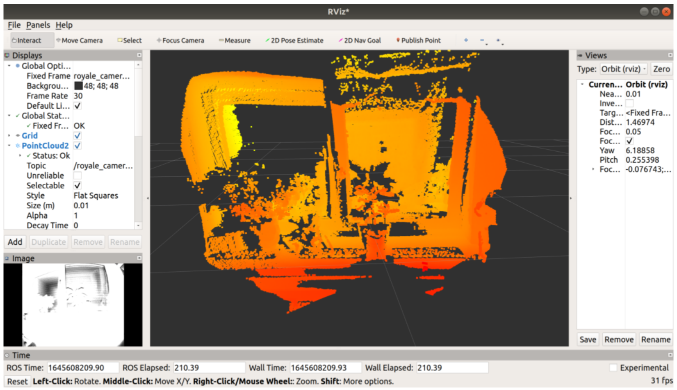
Fig. 5. Sequence diagram of the proposed, fail-operational system.

IV. Implementation and Setup

This section describes the implementation and the experimental setup. As can be seen in Figure 6, the experimental setup consists of two Raspberry Pi 4 Model B (4 GB RAM) representing host 1 and host 2, both running Ubuntu 18.04 natively. This provides a straightforward, inexpensive setup, although the graphical capabilities are being quite exhausted. The Safety Controller is implemented on a Lenovo X1 Carbon running Ubuntu 18.04 natively. For the 3D Flash LiDAR camera, a CamBoard pico monstar is being used. This 3D camera development kit is based on pmd Time-of-Flight (ToF) technology. Powered by USB 3.0, it features a Field of View (FoV) of 100° x 85° degrees, a pixel count of 352 x 287 pixels and covers 6 meter measurement range. The pico monstar supports a variety of operation modes up to 60 frames per second (fps). The ROS is being used to transmit the sensor data

Fig. 6. Setup for the experiment.

provided by the 3D Flash LiDAR camera. This open-source middleware suite for robotics is not an operating system, but a collection of software frameworks for robot software development. ROS contains services for heterogeneous computer clusters, e.g. message passing between processes, low-level device control or hardware abstraction. Processes of ROS are running in nodes. These nodes communicate with each other by sending and receiving messages. They are built up in a graph like architecture. The visualization of the 3D point cloud generated of the 3D Flash LiDAR data is done by a program called RViz. Figure 7 shows RViz illustrating a depth map of the captured scene, in this case a window board. Red points indicate closer objects, whereas yellow points indicate objects further away. For the live migration itself, a Linux software tool called Checkpoint/Restore In Userspace (CRIU) is being utilized. CRIU is able to freeze a running application (or container) and save that check-pointed state as a collection of files. The application (or container) can be restored in the exact same state as it was frozen by using these files.



Global Opti...
Fixed Frame royale_camer...
Backgrou... 48; 48; 48
Frame Rate 30
Default Li... ✓
Global Stat...
Fixed Fr... OK
Grid ✓
PointCloud2 ✓
Status: Ok
Topic /royale_camer...
Unreliable ☐
Selectable ✓
Style Flat Squares
Size (m) 0.01
Alpha 1
Decay Time 0

Type: Orbit (rviz) Zero
Curren... Orbit (rviz)
Nea... 0.01
Inve... ☐
Targ... <Fixed Fra...
Dist... 1.46974
Foc... 0.05
Foc... ✓
Yaw 6.18858
Pitch 0.255398
Foc... -0.076743;...

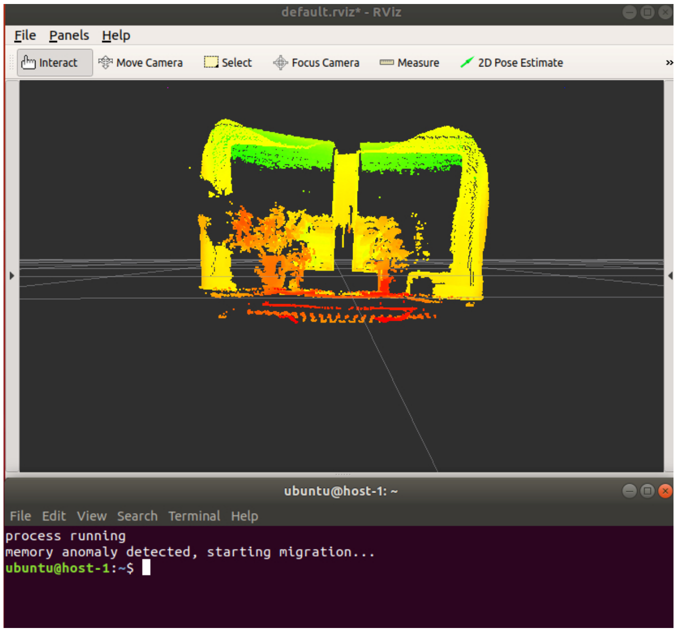
ROS Time: 1645608209.90 ROS Elapsed: 210.39 Wall Time: 1645608209.93 Wall Elapsed: 210.39 ☐ Experimental

Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options. 31 fps

Fig. 7. 3D Flash LiDAR data visualized in RViz.

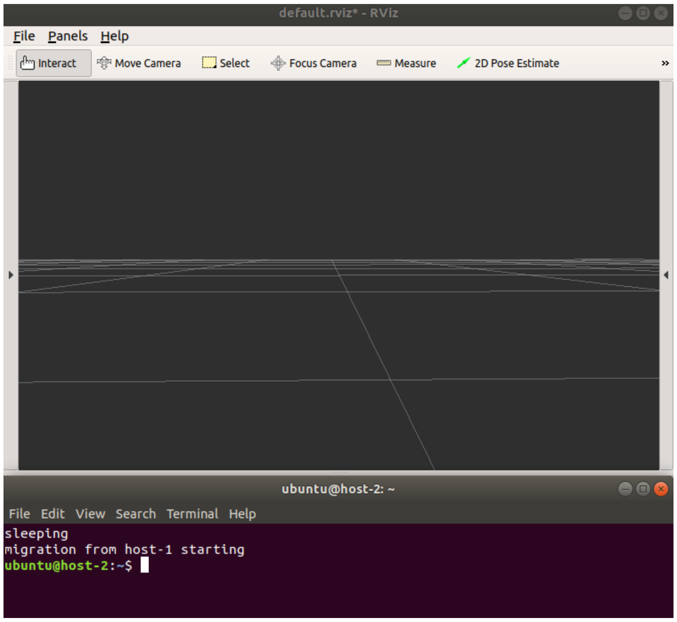
V. RESULTS

In this section we provide the experimental results of a 3D Flash LiDAR live migration, which has been introduced in Section III. The performance of the live migration is highly dependent on the actual use case of the 3D Flash LiDAR. While this test setup is limited to the processing and display of the 3D Flash LiDAR raw data, related tasks like object detection or ego-motion estimation demand a higher amount of resources which leads to an extended migration time. Besides the use case, the actual type of transmitted data influences the live migration. This test setup makes use of a pico monstar operating in ROS. For an average frame rate of 30 frames per second, the output of the overall camera data adds up to around 10.5 megabytes per seconds. Attributable to this, the data of the 3D point cloud makes up around 8 megabytes per second. Considering only the depth image, it



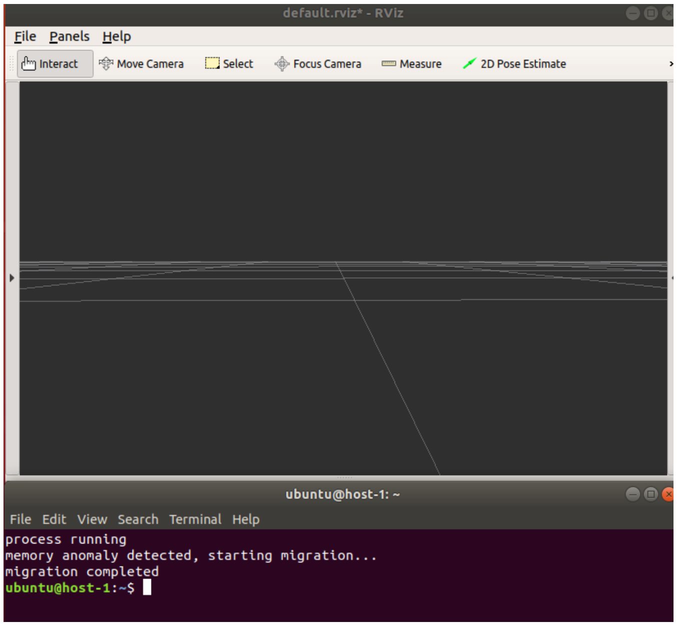
```
process running
memory anomaly detected, starting migration...
ubuntu@host-1:~$
```

Fig. 8. 3D Flash LiDAR data visualized at host 1 while a memory anomaly is detected and a live migration to host 2 is started.



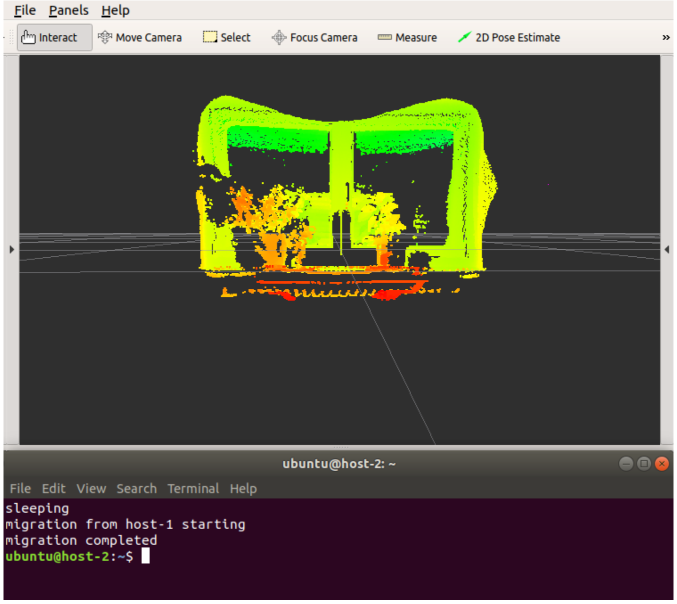
```
sleeping
migration from host-1 starting
ubuntu@host-2:~$
```

Fig. 9. Host 2 is sleeping while a memory anomaly is detected on host 1 and a live migration is started.



```
process running
memory anomaly detected, starting migration...
migration completed
ubuntu@host-1:~$
```

Fig. 10. Host 1 has completed the migration from the 3D Flash LiDAR to host 2.



```
sleeping
migration from host-1 starting
migration completed
ubuntu@host-2:~$
```

Fig. 11. 3D Flash LiDAR data visualized at host 2 after migration has been successfully completed..

adds up to about 2 megabytes per second. The live migration benefits from a low data exchange, so choosing minimum data-requirements is, at least for the live migration performance itself, the best option. At the beginning of the experiment, the raw data of the 3D Flash LiDAR is processed and displayed on host 1. However, after an anomaly has been detected in the memory, the migration process is started by the safety controller. The start of the migration is shown in Figure 8. The aforementioned memory anomaly is artificially caused by filling up the memory, e.g. by a malware. While host 2 is not needed for processing and displaying the raw data, it remains in sleep mode. As soon as it is woken up by the safety controller, the migration process also starts on host 2. As can be seen in Figure 9, no data is visualised on host 2 in the meantime. After the migration process as described in Section III has been completed, no more raw data from the 3D Flash LiDAR is processed on host 1. Consequently, as shown in Figure 10, nothing is visualised on host 1. In contrast, these tasks have now been migrated to host 2. In Figure 11 it can be seen how the processed raw data is now visualised on host 2. Our implementation displays a possible solution to apply system redundancy in order to maintain functionality of a safety-critical system. Under our test conditions, the live migration performs in a timespan of less than a second. A seamless migration, which is a migration, where the downtime is not even noticeable by the end user, seems under the current conditions of our experimental setup not feasible. Our tests have shown, that dropping at least a few frames during live migration is inevitable.

VI. CONCLUSION

In this publication, a concept was proposed to continue to ensure raw data processing in an overall safety-critical system. With this approach, it can be ensured that even in the event of a failure of a data processing system, the environment perception can still be maintained with a short delay if a redundant system is available. However, as this is a live migration, it cannot be guaranteed that no frames will be lost during the migration process and thus the resolution will be lower during the migration process. This state will last longer or shorter depending on the amount of data. However, since this publication was primarily concerned with proving feasibility, performance was of secondary importance. Although it was shown that it is possible to maintain operation, despite restrictions, and thus obtain fail-operational behaviour. Especially for highly automated and autonomous vehicles on land or in the air, it is becoming increasingly important that the environment perception is fail-operational. Therefore, the approach presented can contribute to ensure that the safety in the environment of such highly automated vehicles remains guaranteed and that there is no immediate danger to people in the vicinity and for the environment.

ACKNOWLEDGMENTS

The authors would like to thank all national funding authorities and the ECSEL Joint Undertaking, which funded the ADACORSA project under the grant agreement number 876019.

ADACORSA is funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT) under the program ICT of the Future between May 2020 and April 2023 (grant number 877585). More information: https://iktderzukunft.at/en/.

REFERENCES

[1] J. A. Besada, L. Bergesio, I. Campaña, D. Vaquero-Melchor, J. López-Araquistain, A. M. Bernardos, and J. R. Casar, “Drone mission definition and implementation for automated infrastructure inspection using airborne sensors,” *Sensors*, vol. 18, no. 4, p. 1170, 2018.

[2] K. Divya and G. Girisha, “Autonomous Car Data Collection and Analysis,” 2021.

[3] S. Safavi, M. A. Safavi, H. Hamid, and S. Fallah, “Multi-Sensor Fault Detection, Identification, Isolation and Health Forecasting for Autonomous Vehicles,” *Sensors*, vol. 21, no. 7, p. 2547, 2021.

[4] P. Rahman and E. Y. Bobkova, “Reliability model of fault-tolerant data processing system with primary and backup nodes,” in *IOP Conference Series: Materials Science and Engineering*, vol. 124, no. 1. IOP Publishing, 2016, p. 012023.

[5] N. Druml, G. Macher, M. Stolz, E. Armengaud, D. Watzenig, C. Steger, T. Herndl, A. Eckel, A. Ryabokon, A. Hoess, S. Kumar, G. Dimitrakopoulos, and H. Roedig, “PRYSTINE - PRogrammable sYSTems for INtelligence in AutomobilEs,” in *2018 21st Euromicro Conference on Digital System Design (DSD)*, Aug 2018, pp. 618–626.

[6] ADACORSA Project. (2022) Airborne Data Collection on Resilient System Architectures. https://www.adacorsa.eu.

[7] Z. Khan, F. Abbas *et al.*, “A Conceptual Framework of Virtualization and Live-Migration for Vehicle to Infrastructure (V2I) Communications,” in *2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN)*. IEEE, 2019, pp. 590–594.

[8] J. Cano, E. Molinos, V. Nagarajan, and S. Vijayakumar, “Dynamic process migration in heterogeneous ROS-based environments,” in *2015 International Conference on Advanced Robotics (ICAR)*. IEEE, 2015, pp. 518–523.

[9] H. Jeong, J. Baik, S. Min, and K. Kang, “Improved drone reliability using a robot operating system,” in *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2016, pp. 151–153.

[10] N. An, S. Yoon, T. Ha, Y. Kim, and H. Lim, “Seamless virtualized controller migration for drone applications,” *IEEE Internet Computing*, vol. 23, no. 2, pp. 51–58, 2018.

[11] F. Wamser, F. Loh, M. Seufert, P. Tran-Gia, R. Bruschi, and P. Lago, “Dynamic cloud service placement for live video streaming with a remote-controlled drone,” in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2017, pp. 893–894.

[12] A. R. García, L. R. Miller, C. F. Andrés, and P. J. N. Lorente, “Obstacle detection using a time of flight range camera,” in *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. IEEE, 2018, pp. 1–6.

[13] T. Ratshidaho, J. R. Tapamo, J. Claassens, and N. Govender, “ToF camera ego-motion estimation,” in *2012 5th Robotics and Mechatronics Conference of South Africa*. IEEE, 2012, pp. 1–6.

[14] M. Berk, O. Schubert, H.-M. Kroll, B. Buschardt, and D. Straub, “Exploiting redundancy for reliability analysis of sensor perception in automated driving vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 5073–5085, 2019.

[15] E. Safronov, M. Vilzmann, D. Tsetserukou, and K. Kondak, “Asynchronous behavior trees with memory aimed at aerial vehicles with redundancy in flight controller,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3113–3118.

[16] C. Huang, F. Naghdy, H. Du, and H. Huang, “Fault tolerant steer-by-wire systems: An overview,” *Annual Reviews in Control*, vol. 47, pp. 98–111, 2019.