Performance analysis of real time embedded systems with space and time partitioning

Guillaume Champagne Michel Dagenais

December 6th, 2018

Polytechnique Montréal

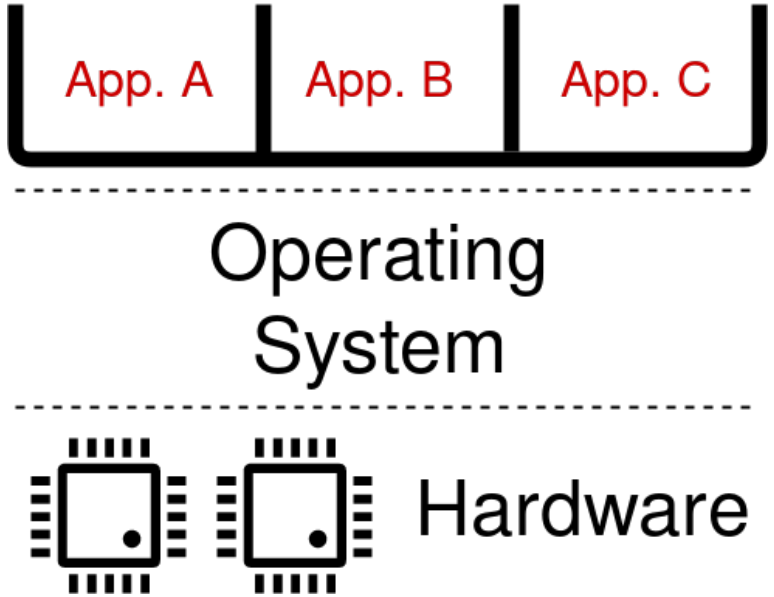
Laboratoire **DORSAL**

Agenda

- Space and Time partitioning with ARINC 653
- Performance in real time systems
- Analysis of ARINC 653 traces
- Conclusion

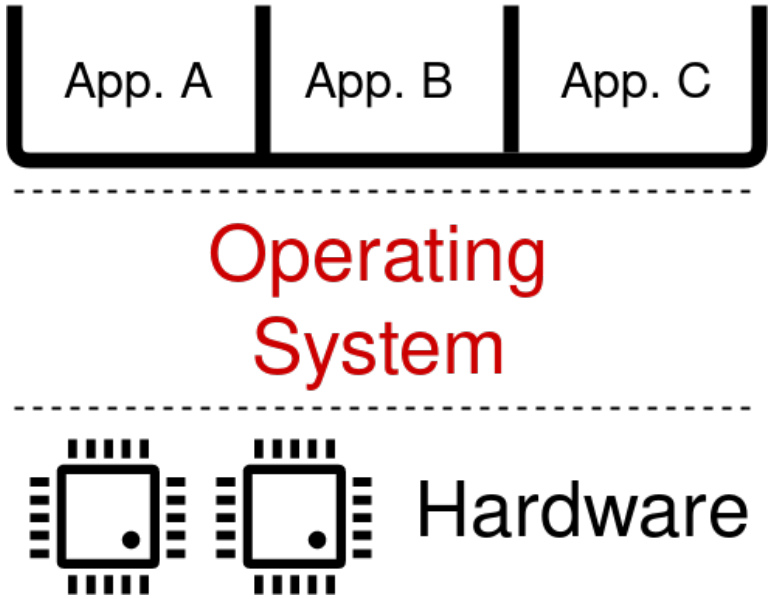
Space and time partitioning

- In **safety critical systems**, modules with different **criticality levels** may cohabit using the **same hardware** resources.
- In avionics, this trend is known as **Integrated Modular Avionics** (IMA).
- Multiple applications may end up sharing the same processor.



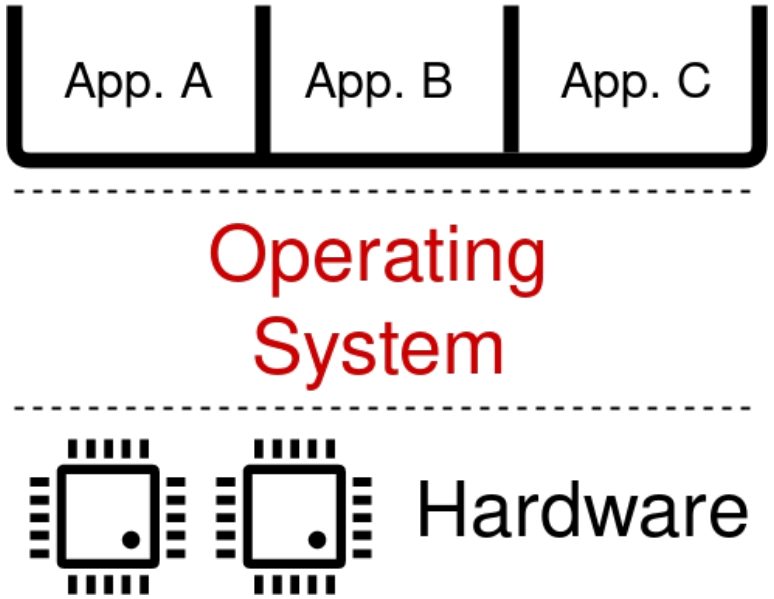
Space and time partitioning

- The operating system has to **isolate** the applications.
- Applications are isolated using space and time partitioning.
  - Space partitioning: Restriction of **memory** between partitions
  - Time partitioning: Attribution of a **fixed scheduled** to each partition.



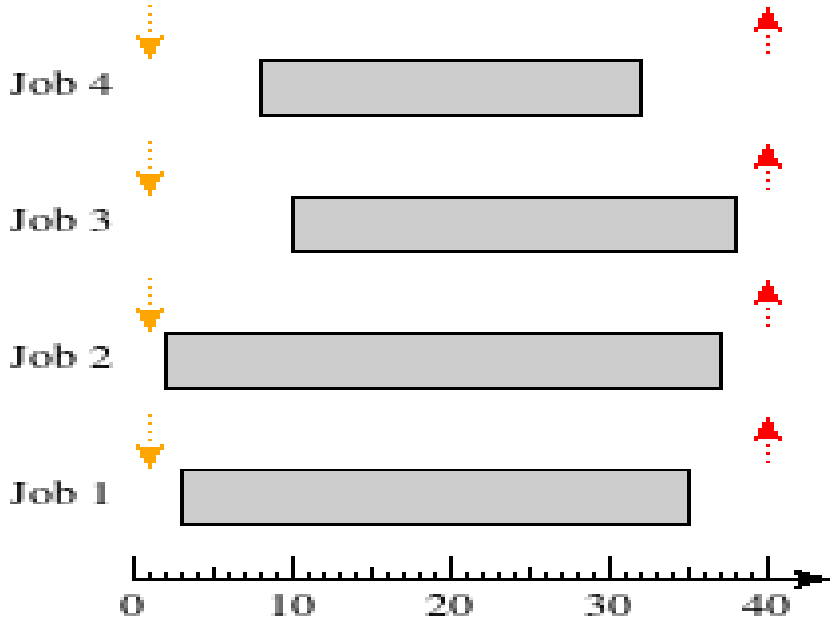
Space and time partitioning

- The operating system has to **abstract** hardware resources, but applications still require common services implemented via hardware (timer, file I/O, etc).
- A common API agreed and used by every contributor would be useful.
  - In commercial avionics, this is the **APEX API** of the ARINC 653 standard.



Performance in real time systems

- Safety-critical real time systems are subject to very stringent **temporal constraints**.
- Performance is not only a matter of speed, but also of **determinism** and **timeliness**.

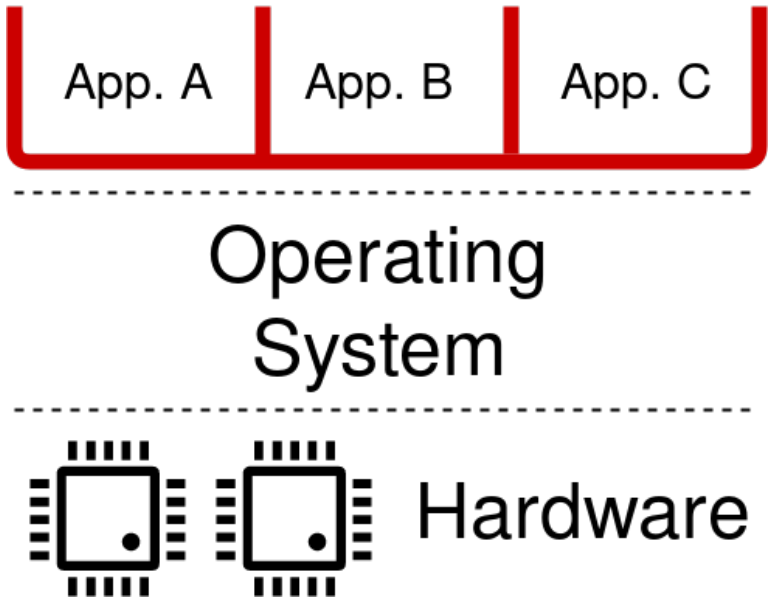


Performance in real time systems

- Troubleshooting performance related issues in real time systems has to be done with **minimal overhead**.
- Issues might occur once in a while and will be hard to reproduce.
- **Tracing** is the appropriate solution for this.

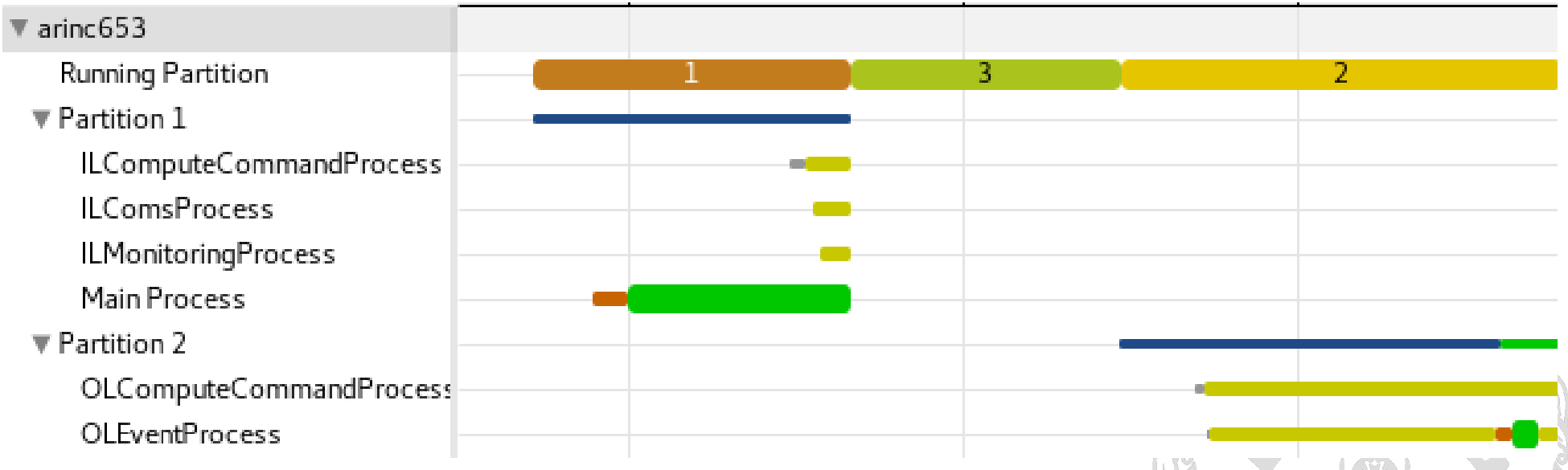
Tracing

- Tracing operating systems with space and time partitioning is not different from traditional operating systems.
- The main difference is **partitions**.
  - Comparable to **containers** or **Linux control groups**, with strict time constraints .



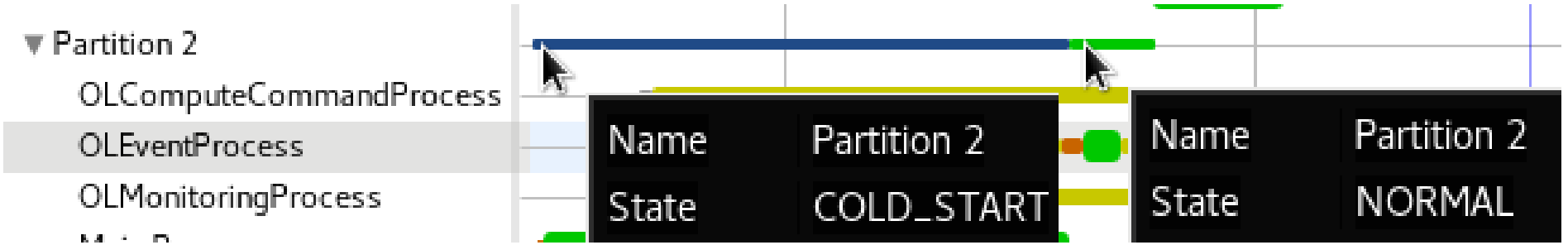
Trace analysis

- An equivalent of the **Control Flow View** in Trace Compass was developed for ARINC 653 operating systems.
- Using a set of events similar to those in Linux, this timeline of the processes can be created.



Trace analysis

- ARINC 653 defines three **partition operating modes**.
  - WARM_START, COLD_START, NORMAL
- The partition operating mode is displayed on the partition line.



Trace analysis

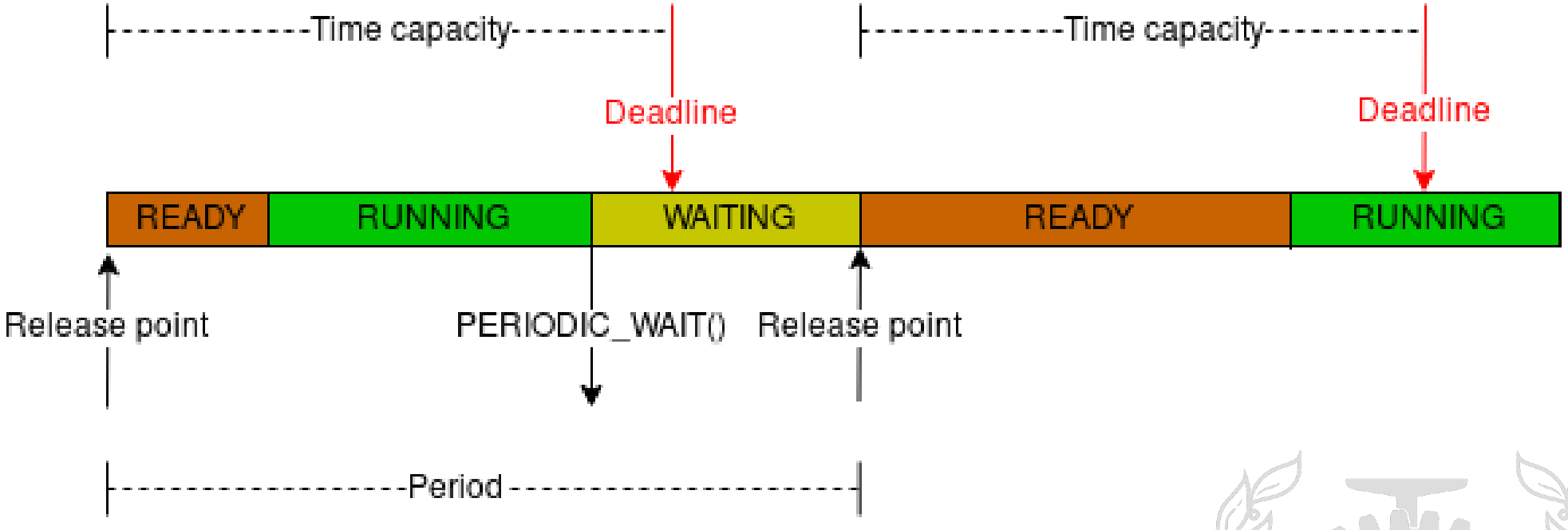
- Applications created for an ARINC 653 operating system will work closely with the APEX API.
- **APEX calls** are displayed using the incoming Trace Compass **overlays.**
- Additional details are provided in the tooltip.

Trace analysis

- How can we help developers detect and remedy to performance issues in ARINC 653 operating systems?
  - **Deadline misses**
  - Latencies
- Detecting real time jobs.
- Showing job executions.
- Show the origin of issues.

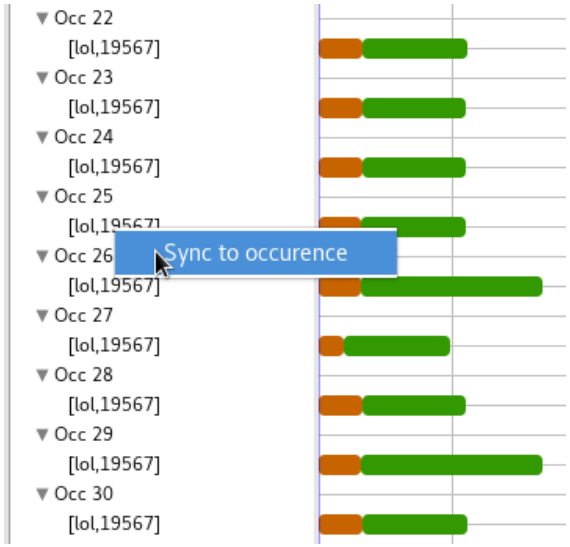
Detecting deadline misses

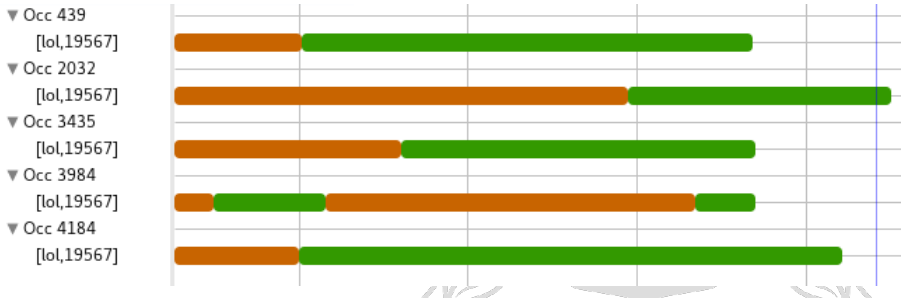
- ARINC 653 defines time management for processes. Each periodic process is associated with a **time capacity**, a **period** and a **deadline**.



Detecting deadline misses

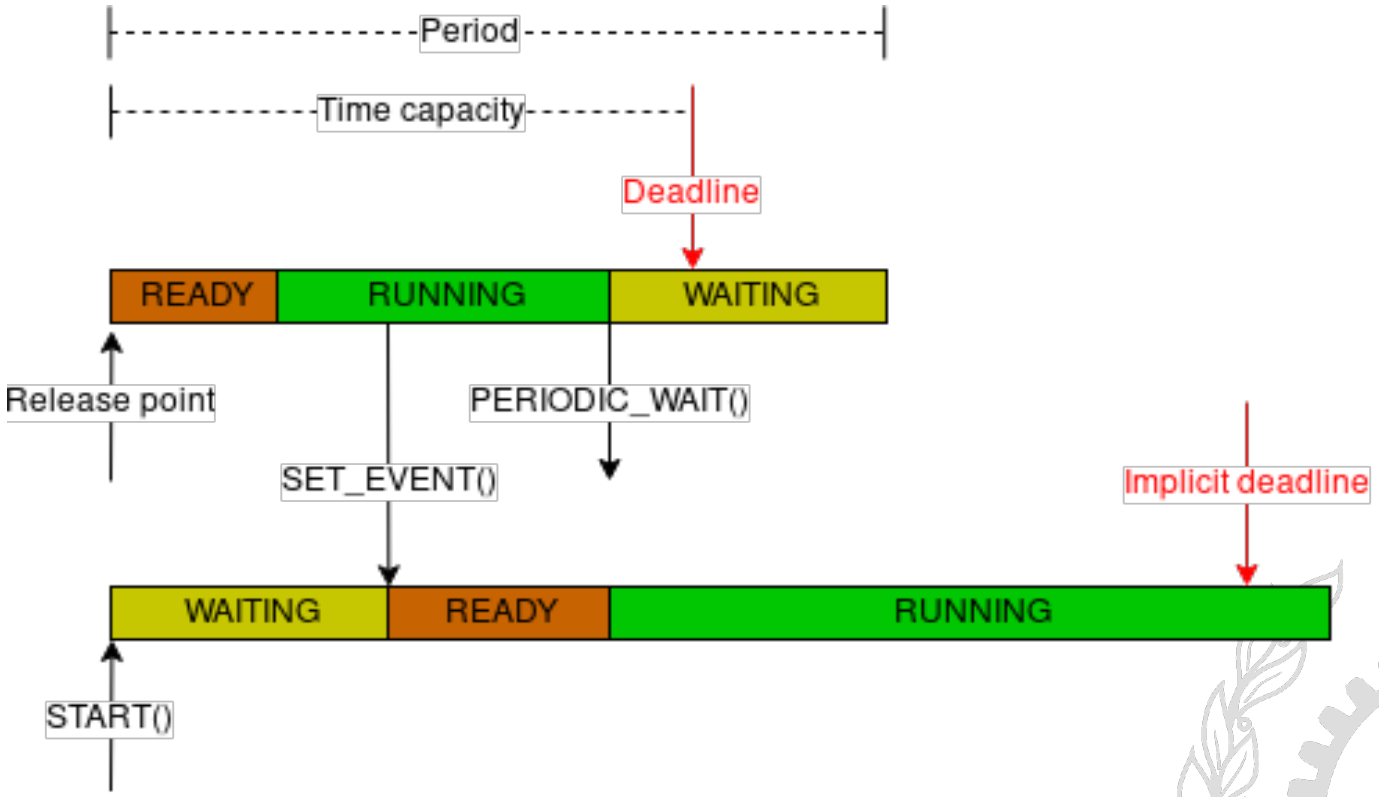
- Simple periodic processes follow a pattern easily detectable if **APEX calls** are traced.
  - Aperiodic processes with simple execution patterns can also be matched.
- A list of every job (occurrence of a task) for a specific process can be generated from the Control Flow View.





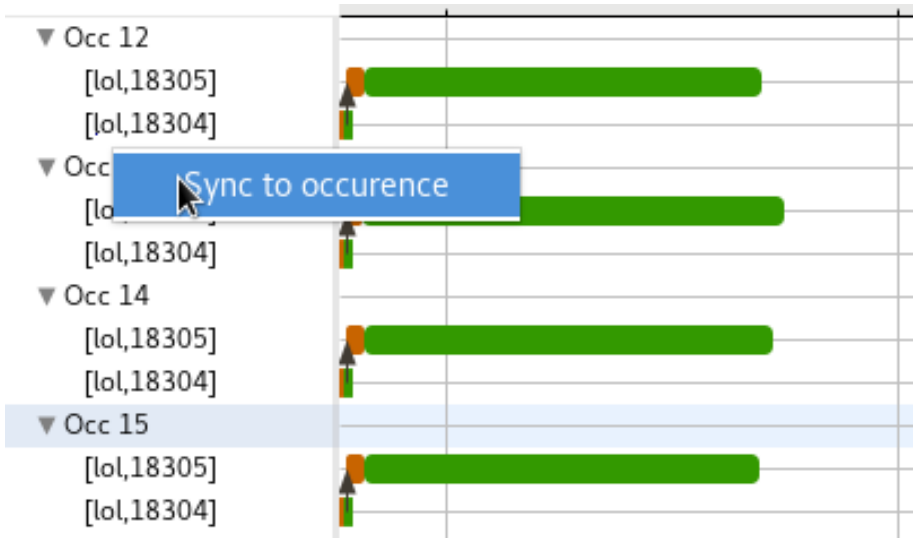
Detecting deadline misses

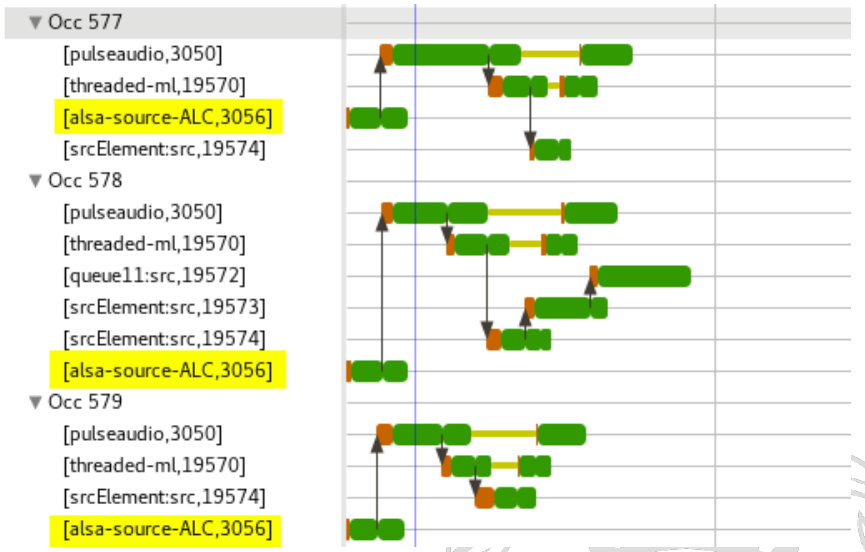
- Periodic processes often release jobs in the system. We must also detect if any job released missed a deadline.
- The released job deadline might not be explicit in the trace.



Detecting deadline misses

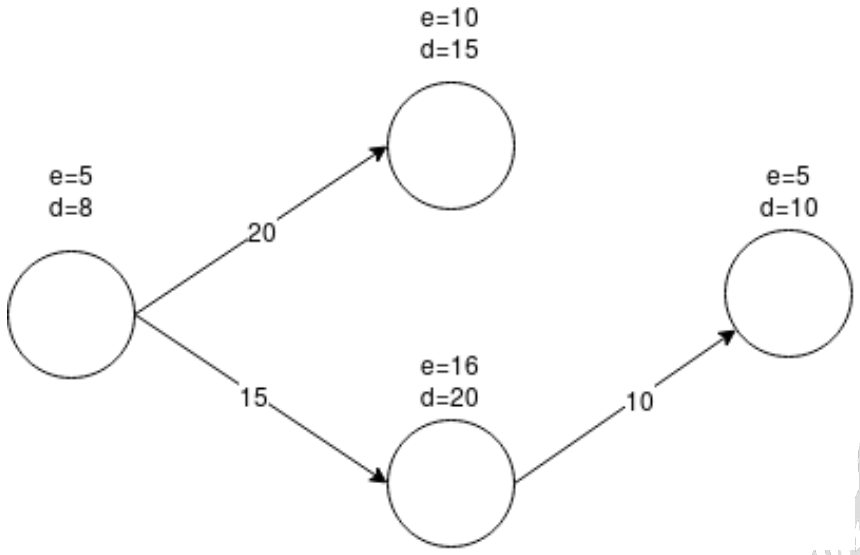
- The process that released the jobs can be detected as a periodic process.
- We can follow the **outgoing vertical edges** in the execution graph of the process that releases the jobs to build a sub graph.
- The jobs can be listed and filtered depending on their execution time.





Limitations

- Different types of jobs with different execution times might be released from the same process.
  - Filtering is not as efficient.
- Real time task models in schedulability analysis can represent complex tasks composed of multiple jobs.



Conclusion

- **ARINC 653** is a standard for space and time partitioning operating systems.
- A plugin was developed to **analyze** their traces.
- We propose real time job detection techniques to find **deadline misses**.

Source: Airbus A380, Wikipedia

Questions?

guillaume.champagne@polymtl.ca