Dependability aspects of Operating Systems and Middleware

Non-functional properties in Operating Systems and Middleware

Seminar topics 2016
Driver verification

- Exhaustive verification has become feasible for small software systems, such as device drivers
  - Concurrency, state space explosion
  - Abstraction of the C programming language needed
- What aspects of real world programs can be proven correct and how?

Proactive recovery and software rejuvenation

• Software aging: progressive degradation of a running system
  • Due to resource exhaustion
  • Due to fragmentation
  • Due to error accumulation

• Proactive approaches: health monitoring, restart, reboot, ...

• How can aging-related failures be prevented?


Fault tolerance with microkernels

• Operating system reliability still a major issue
• Microkernels can enhance dependability by
  • A smaller and therefore less faulty kernel
  • Shorter error propagation
  • Easy and fast restart of failed servers
• What are the trade-offs when using microkernel architectures for fault tolerance?

• CapROS: The Capability-based Reliable Operating System http://www.capros.org/
Byzantine fault tolerance (BFT) in practice

• Byzantine fault model: faulty nodes may present different results to different observers
• Reaching consensus is hard, theoretically complex
• How is BFT implemented in modern real-world middleware?

• UpRight library https://code.google.com/archive/p/upright/
Case studies / post mortems

• Distributed systems fail in complex ways
• DevOps as an increasingly hard challenge
• How well do fault tolerance mechanisms work in practice?
  How does monitoring and recovery work?

• CSC outage post-mortem https://csc.fi/web/blog/post/-/blogs/the-largest-unplanned-outage-in-years-and-how-we-survived-it


• Azure downtime due to leapday bug https://azure.microsoft.com/de-de/blog/summary-of-windows-azure-service-disruption-on-feb-29th-2012/

• ... https://Failure.wiki
Dependable Tandem systems

• Fault tolerant server systems since the 70s
• Fail fast design pattern
• Redundancy at every layer in HW and SW
• What can we learn from early fault tolerant operating systems?

