Dependability in distributed applications

Scientific Talk

by Svetlana Slavova

Outline

- The threats (Errors, Faults, Bugs [Bunny ☺])
- Dependability
- Availability
- Reliability
- Fault-tolerance
- Example

The threats in the distributed applications

- Failure – incorrect result; occurs when the system does not provide a correct service
- Error – a human action that produces an incorrect result (ex.: syntax error, logical error)
- Fault – an incorrect step or data in a system; Everything looks correct but we cannot get a result
- Bug – informal word; deviation from the expected result

Dependability

- Dependability includes:
  - Availability – readiness for correct service;
  - Reliability – continuity of correct service;
  - Safety – absence of catastrophic consequences on the user(s) and the environment;
  - Security – confidentiality (absence of unauthorized disclosure of information) & Integrity (absence of improper system state alterations).

Availability

- Availability is the proportion of time a system is in a functioning condition
- Availability is the ratio of the total time a unit is capable of being used during interval to the length of the interval
- Example: Availability of 100/168 if the unit is capable of being used for 100 hours in a week

Reliability

- Definition: “The probability that a system will perform its intended function during a specified period of time under stated conditions.”
- Reliability parameter: mean-time-between-failure (MTBF) – failure rate (number of failures during a given period)

How to achieve availability & reliability

- Fault-prevention – guarantees that the system does not have faulty components
- Fault-tolerance – assumes that although a fault-prevention has been done, there could be faulty components in the system
- Fault-removal – removes faults of the system by using verification (static & dynamic verification)
- Fault-forecasting – estimates the current and the future number of faults in the system, and their consequences for the system

Fault-tolerance (I)

- Fault-tolerance deals with the question: “How to deliver a correct service in the presence of faults?”
- Fault-tolerant system continues working properly in a case of failure in one or more of its components

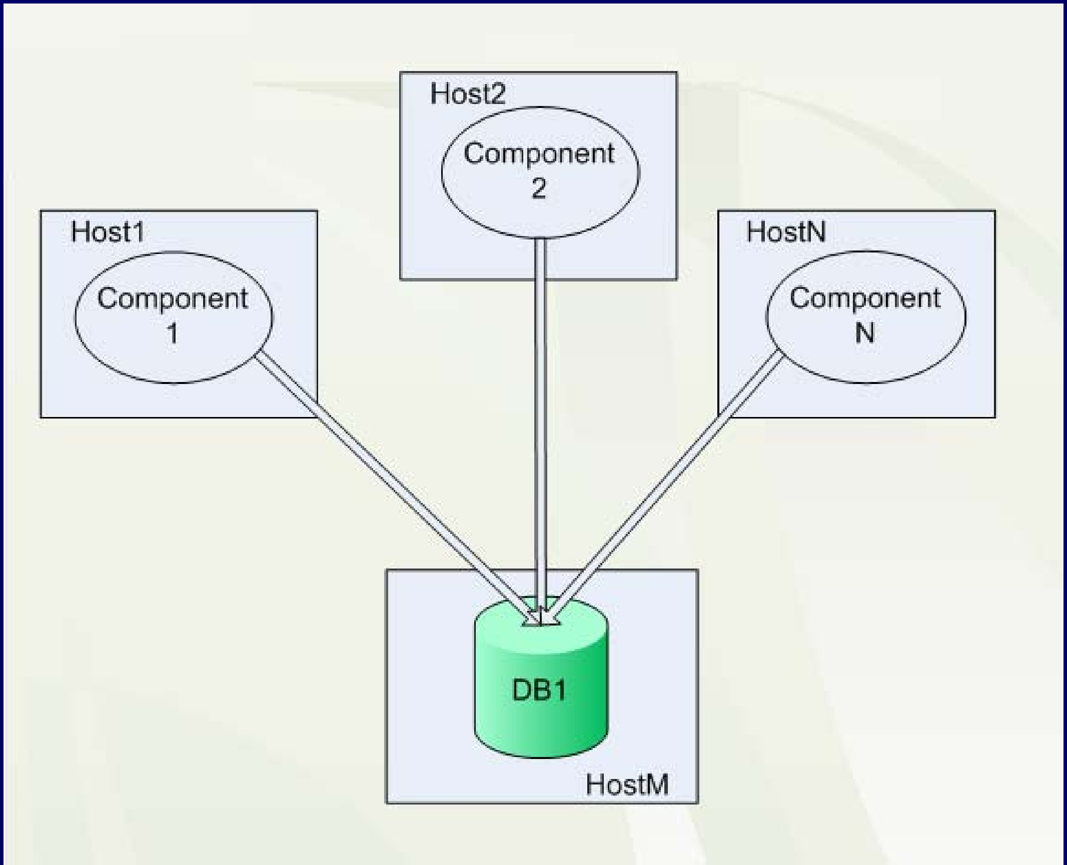
Fault-tolerance (II)

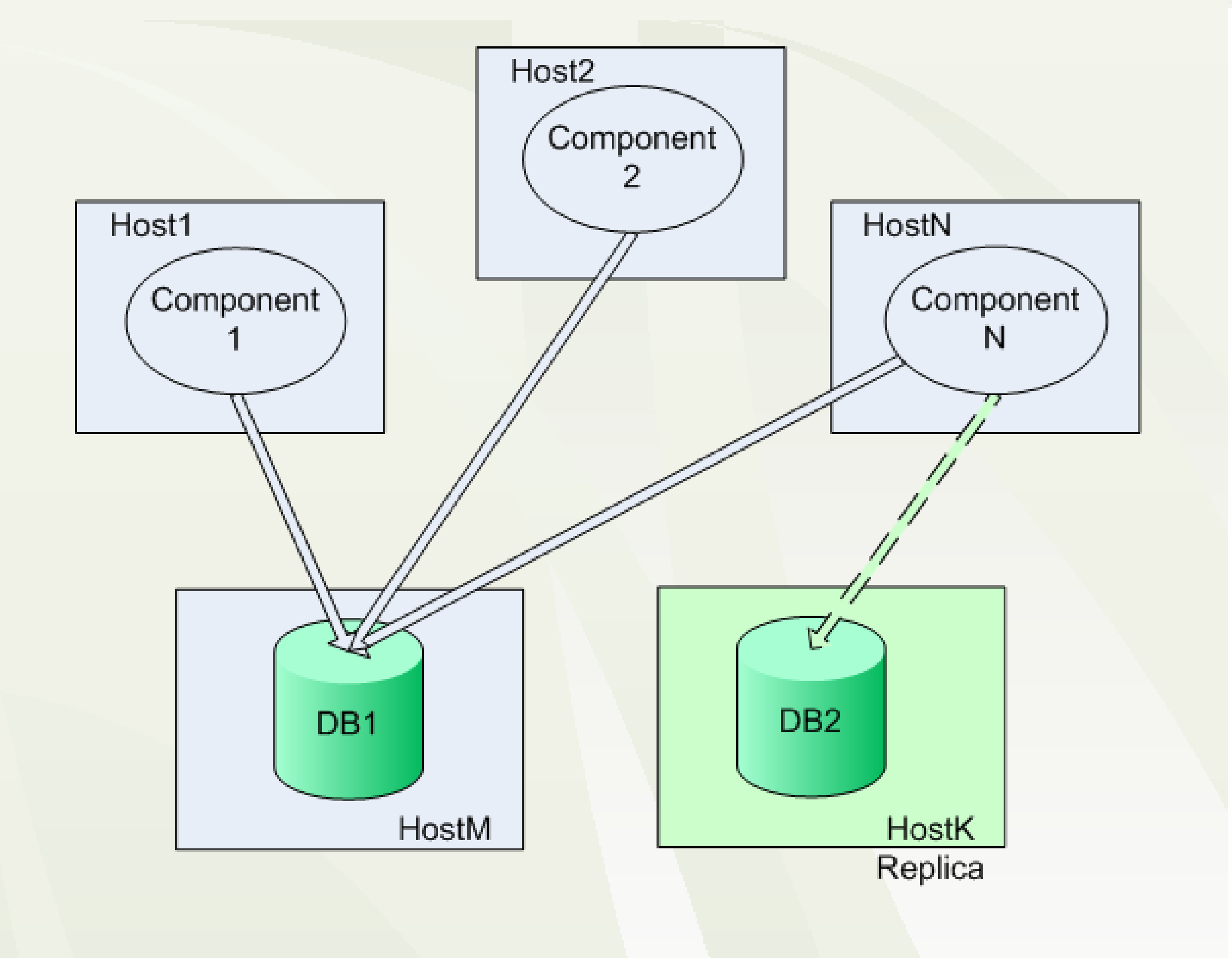
- Achieving fault-tolerance through recovery:
  - Roll-forward
  - Roll-back
- Achieving fault-tolerance through duplication:
  - Replication
  - Redundancy
  - Diversity

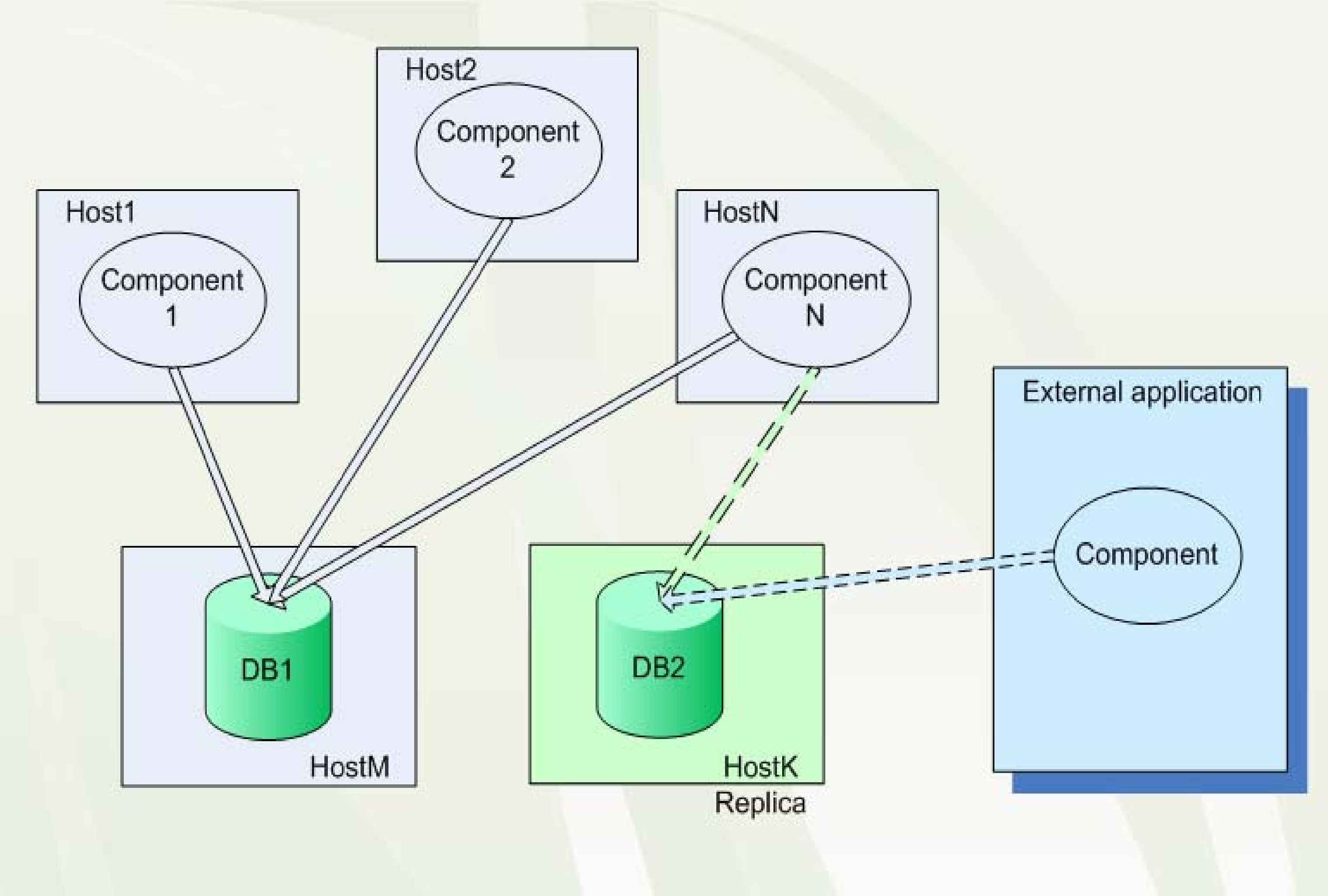
Important issues

- Synchronization of the replicas in order to have the same internal state
  - How to synchronize
  - When to synchronize (Read/Write request)
- System complexity
- System overhead (more communication => more traffic, higher execution time)

Distributed System Example




Host2
Component 2
Host1
Component 1
HostN
Component N
DB1
HostM
DB2
HostK
Replica


Host2
Component 2
Host1
Component 1
HostN
Component N
External application
Component
DB1
HostM
DB2
HostK
Replica

References

- Fundamental Concepts of Dependability, A. Avizienis, J. Laprie, B. Randell
- Building reliable secure computing systems out of unreliable insecure components, J. Dobson, B. Randell
- http://en.wikipedia.org/wiki/Ilities

Thank you for your attention!