Distributed Algorithms

Introduction

Univ.-Prof. Dr.-Ing. habil. Gero Mühl

Architecture of Application Systems (AVA)
Faculty for Informatics and Electrical Engineering
University of Rostock

Overview

> Definition of a distributed system

> Motivation for the use of distributed systems

> Characteristics of distributed systems

> Conceptual problems in distributed systems

> Basic models for distributed systems and algorithms

Distributed System – Some Definitions

"A distributed computing system consists of multiple autonomous processors that do not share primary memory, but cooperate by sending messages over a communication network."
-- *Henri Bal*

"A distributed system is a collection of independent computers that appears to its users as a single coherent system"
-- *Andrew S. Tanenbaum*

"A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable."
-- *Leslie Lamport*

Distributed System – Definition & Demarcation

- Distributed systems
  - Consist of *autonomous* computers, that do not share primary memory, are connected *loosely* by a network, and communicate through *message exchange* to achieve a common functionality
- Computer networks
  - Focus is on connecting the computers to get access to other computers, but not on the cooperation of the computers
  - Computer networks enable, e.g., the access to a remote computer via RDP or SSH
- Parallel computers
  - Usually share common physical memory

Distributed Systems – Goals and Advantages

- Capacity gain through concurrent processes
  - If application is well parallelizable, an otherwise unreachable capacity is possible
    - Cracking a cryptographic key
    - Searching for the prime factors of a large number
    - …
- Flexible and incremental expandability
  - Including new functionality or scaling of the system by adding more computing nodes

Distributed Systems – Goals and Advantages

- Gain access to remote data and services
  - Access to several databases (e.g., library catalogue)
  - Usage of unused computer capacity (e.g. SETI@home)
- Profitability
  - Connected PCs usually offer better cost-benefit ratio than a similar expensive supercomputer
  - ⇒ Distribute application on many small computers if possible
- Fault tolerance
  - Redundant storage of data
  - Primary/Backup Server (backup takes over if primary breaks down)

Distributed Systems – Characteristics

- Concurrency
  - Many processes with different execution speeds
  - Many events happen at the same time
  - Actions are not reproducible → Non-determinism
  - Mechanisms for coordination and synchronization of activities (e.g., for exclusive access on resources)
- No shared primary memory
  - Since the state is distributed to the nodes, no node has a global view on the complete state of the system
  - Mechanisms for a consistent view on the state (e.g., for the secure detection of termination)

Distributed Systems – Characteristics

- Communication only through message exchange
  - Message delay varies unpredictably and is usually not limited (e.g., delay in an Ethernet depends on the current network load)
- Communication is error-prone
  - Message loss, duplication, and corruption is possible (e.g., due to network overload or disruption)
  - ⇒ Mechanisms for detection and handling of communication errors
- Communication is inherently insecure
  - A malicious attacker may intercept, adulterate, add, or suppress messages
  - ⇒ Security mechanisms

Distributed Systems – Characteristics

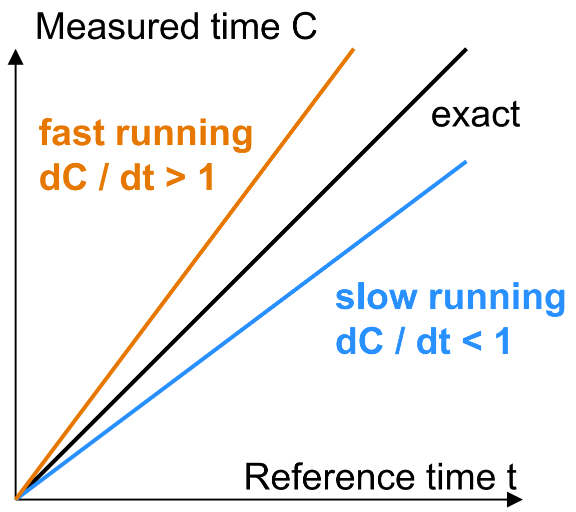
- Computers and network connections can fail independent from each other → partial break down
  - In large systems, failures of computers or network links are very likely to occur quite frequently
  - A failure of a single component should not lead to a breakdown of the whole system
  - ⇒ Fault tolerance mechanisms
- No reliable error detection possible in the general case
  - Slow processes or connections cannot be distinguished from those that failed
  - ⇒ Mechanisms to deal with uncertainty

Distributed Systems – Characteristics

> Independent physical system clocks with different speeds clock apart
  ⇒ Clock synchronization mechanisms

> Clock resolution and accuracy of clock synchronization are limited

> If the relative order of events in reference to each other is more important than the exact time of their appearance, logical clocks are a possible alternative



Distributed Systems – Characteristics

> Different administrative domains
  > Centralized administration impossible in large systems
  > For each domain, a different administrator is responsible
  ⇒ Decentralized administration and automated administration

> Components are heterogeneous
  > Interoperability between components is difficult to guarantee in heterogeneous systems
  ⇒ Protocol and interface standardization

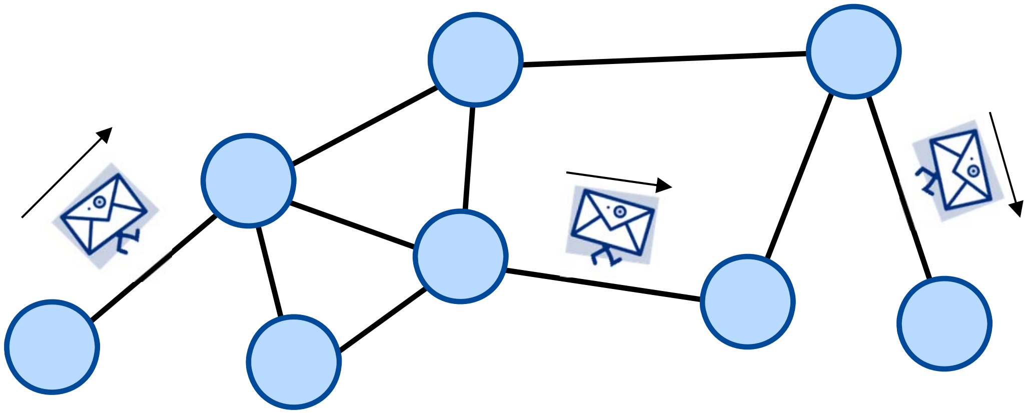
Distributed Systems – Conclusions

- Designing, implementing, testing and securing of a distributed system is much more complex than in the case of a centralized system
- Only if the distributed system is configured adequately, it can be more powerful, more scalable, more fault tolerant, and perhaps also similarly secure than a centralized system
- Precondition for that is the understanding of occurring phenomena in distributed systems, the resulting problems, and the knowledge about possible solutions
- The lecture deals with models, concepts and algorithms contributing to the understanding and controlling of the phenomena in distributed systems

Basic Models for Distributed Systems

Distributed System – An abstract view

- A distributed system is a connected graph consisting of a set of nodes and a set of edges
  - nodes are also denoted as processes, computers etc.
  - edges are often denoted links, channels etc.
- Nodes can exchange messages with their respective direct neighbors over channels
- Arbitrary nodes can communicate through message routing

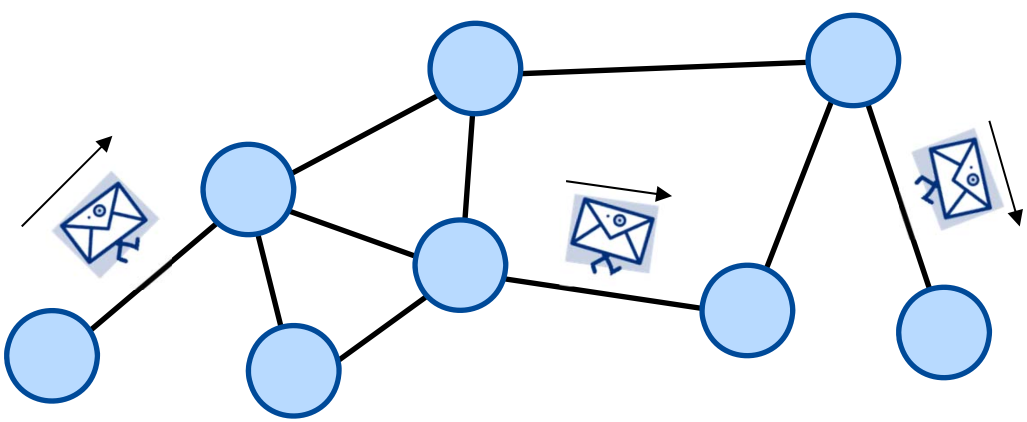


Default for the lecture

Network-Topologies

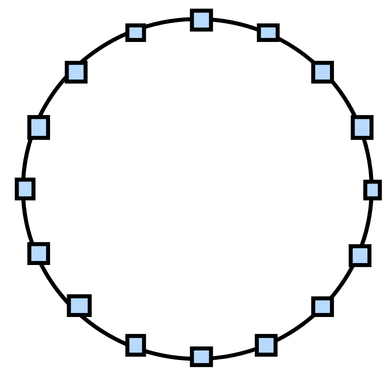
- A topology is an undirected graph $G = (V, E)$
  - Node set $V = \{v_0,\ldots,v_{n-1}\}$
  - Edge set $E = \{e_0, \ldots, e_{m-1}\}$, $e_i = (v_{i\_1},v_{i\_2})$

- Static vs. dynamic topology
  - Static node- and edge set do not change over time
  - Dynamic node- and edge set change over time

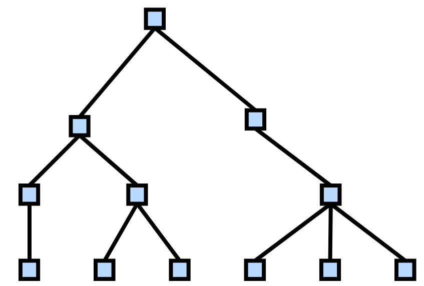


Special Topologies

- Ring
  - Number of nodes = number of edges ($n = m$)
  - $E = \{(v_i, v_j) \mid j = (i + 1) \bmod n\}$
  - Node degree (neighbors per nodes): 2
  - Node degree constant with scaling
  - Diameter (longest shortest path): $\lfloor n / 2 \rfloor$

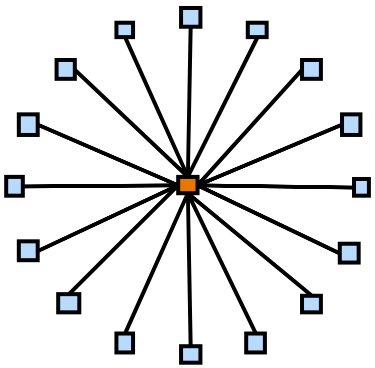


- Tree
  - Connected graph without cycles
  - $m = n - 1$

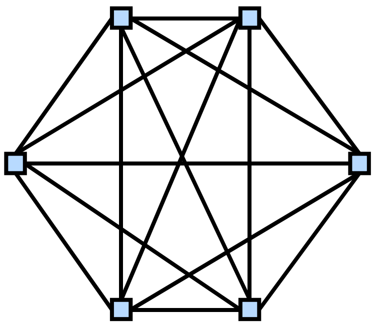


Special Topologies

> ## Star
>
> - Special tree with central node v_0
> - $E = \{(v_0, v_i) \mid i \neq 0\}$
> - Neighbors per node: 1 or $n - 1$
> - Diameter: 2
> - Not (infinitely) scalable

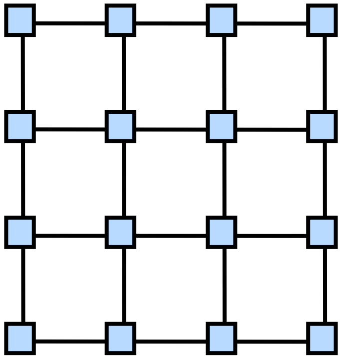


> ## Complete graph / fully meshed graph
>
> - $m = n (n - 1) / 2$
> - $E = \{(e_{i\_1}, e_{i\_2}) \mid i\_1 < i\_2\}$
> - Neighbors per node: $n - 1$
> - Diameter: 1
> - Not (infinitely) scalable

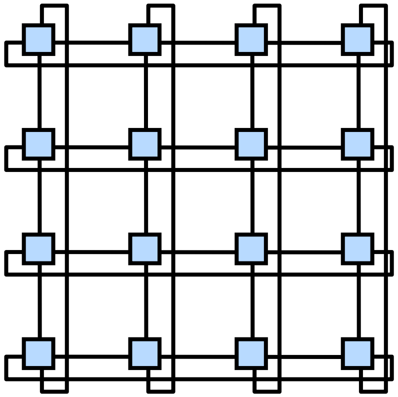


Special Topologies

- Meshes
  - Despite 2D meshes, also 3D meshes are possible
  - Constant node degree with scaling
  - Diameter increases with root of node number
  - Expandable in small increments (e.g., additional row or column)
  - Good support of algorithms with local communication structure (e.g., modeling of physical processes)



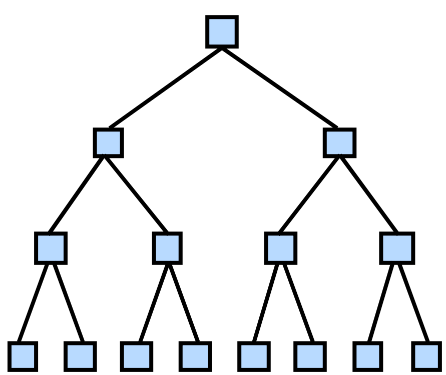
4x4-Mesh



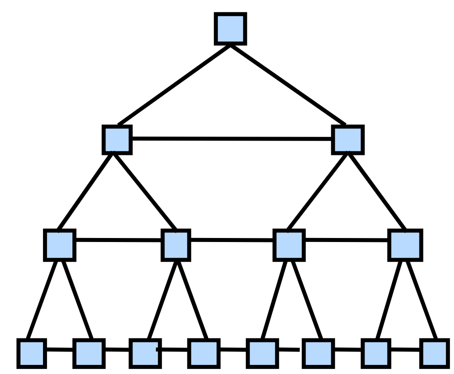
4x4-Torus

Special Topologies

- Complete k-ary tree and X-Tree
  - Constant node degree with scaling
  - $h = \lfloor \log_k n \rfloor$ (logarithmic height in node number)
  - Expandable in powers of k
  - Node degree: maximal $k + 1$ (or maximal $k + 3$)



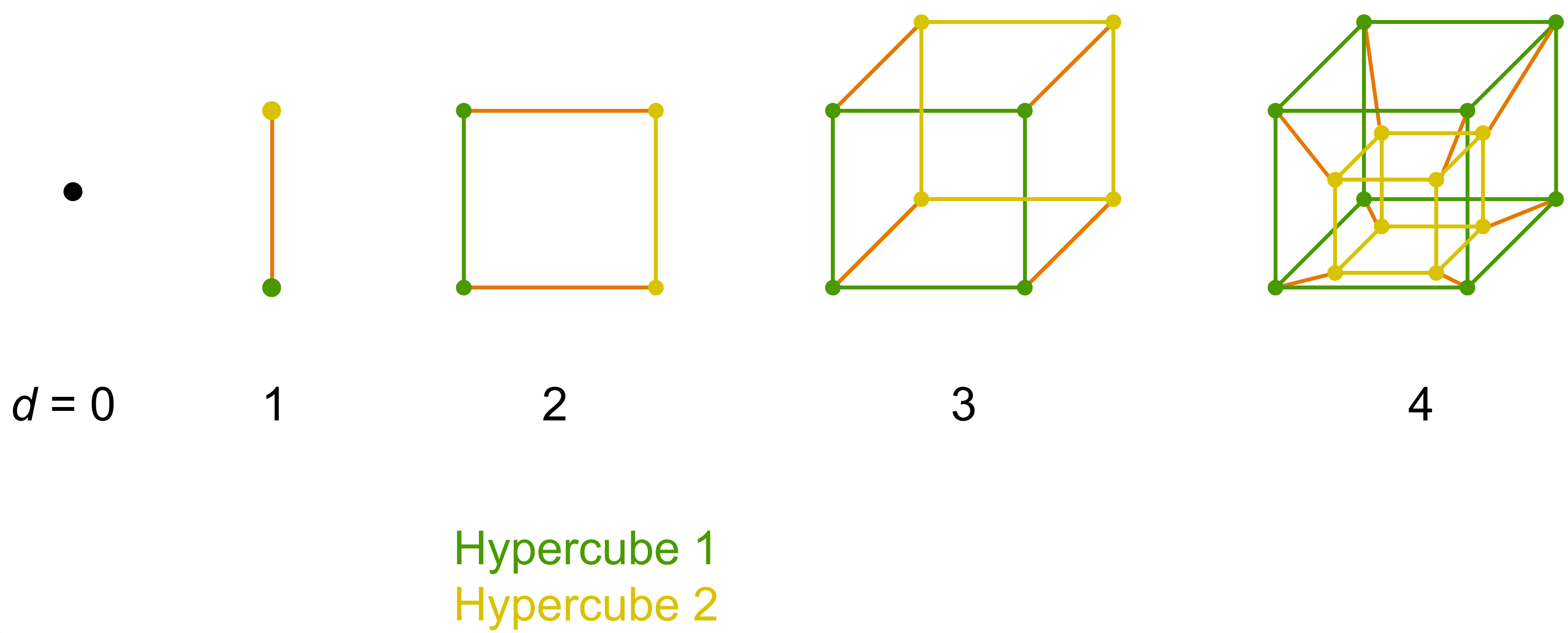
Binary Tree



Binary X-Tree

Special Topologies

- A hypercube is a cube of dimension *d*
- Recursive construction
  - Hypercube of dimension 0: single node
  - Hypercube of dimension *d* + 1: *„Take two cubes of dimension d and connect the corresponding nodes“*



Characteristics of Hypercubes

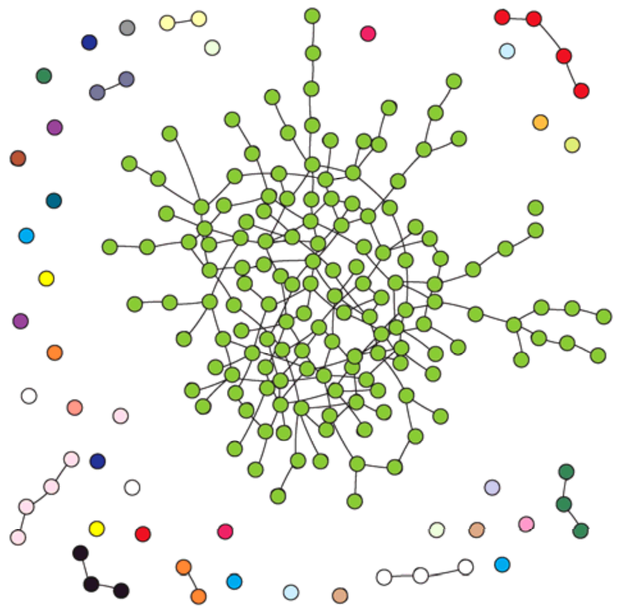
- Number of nodes $n = 2^d$
- Number of edges $m = d \cdot 2^{d-1} = (\log n) \cdot n / 2$ → $O(n \log n)$
- Diameter, i.e., longest shortest path length between two nodes (occurs with diagonal opposite nodes)
  $d = \log n$ → $O(\log n)$
- Many path alternatives fosters fault tolerance
- Node degree = d → not constant with scaling
- Average path length = $d / 2$
- Simple routing of messages
  - XOR of send address and destination address (one bit vector with d bits each)
  - Dimensions whose bits equals 1 in the result vector are traversed successively. Does the order matter at all?

Characteristics of Real-World Networks

- Short Paths
  - Every node is connected to every other node by a relatively short path
- High Clustering Coefficient
  - If *A* is connected to *B*, and *B* is connected to *C*, the probability is high that *A* is also connected to *C*
  - If the probability equals 1, the connection relation is transitive
- Scale-Free
  - Distribution of the number of neighbors of a node follows a power law
  - Many nodes with few neighbors and only a few nodes with many neighbors

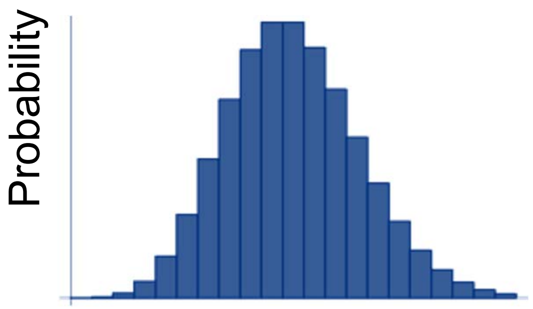
Random Networks

> Generation of a random connected network with n nodes and m edges
>
> > Choose m-times two nodes randomly and connect them



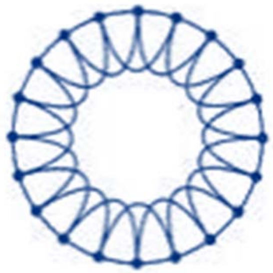
Source: [1]

> Characteristics
>
> > For $m > n / 2$, a large component with short paths is formed
> >
> > Clustering coefficient is low
> >
> > Node degree is approximately Poisson-distributed



Small World-Networks

- Characteristics
  - Short paths
  - High clustering-coefficient
- Generation according to Watts and Strogatz [2]
  - Starting point is a ring in which each node is connected with his k succeeding nodes (k = 2 in Fig.)
  - Choose a new node randomly for each edge with probability p
  - Here: Average node degree is k



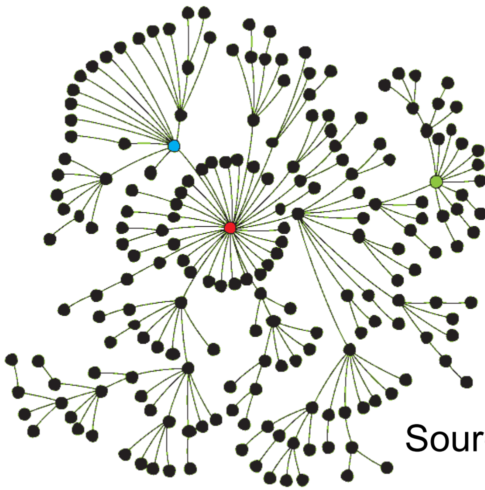
regular
($p = 0$)

Small World
($0 < p \ll 1$)

random
($p = 1$)

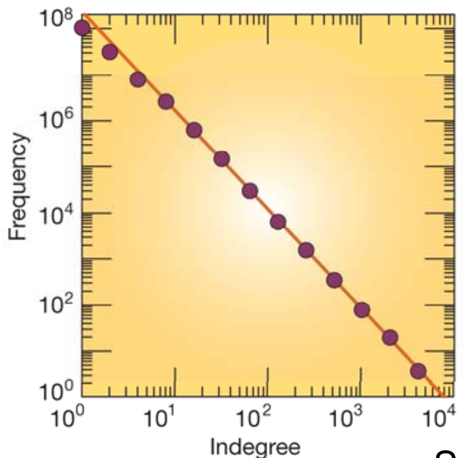
Scale-Free Topologies

- Additional characteristics to those of Small World-topologies
  - Distribution of the number of neighbors follows a power law, i.e., $P(k) \sim k^{-\gamma}$
  - ⇒ Many nodes with few neighbors
  - ⇒ Few nodes (*hubs*) with many neighbors
- Close to many real networks (e.g., Internet, WWW, social networks)



Source: [1]

Ingoing degree of web-pages in the WWW



Source: [1]

Scale-Free Topologies

- Generation through *Preferential Attachment*
  - Start with a small number of nodes n_0
  - With every step, a node is added and connected with $n_1 \leq n_0$ other nodes
  - Hereby, a node i with degree k_i is chosen with probability

$$P(k_i) = k_i / \sum_j k_j$$

  - ⇒ Thus, a node is chosen with a higher probability if it already has many neighbors
  - ⇒ „The rich get richer“
  - For a topology generated with those rules: $\gamma = 3$

Thank you for your kind attention!

Univ.-Prof. Dr.-Ing. habil. Gero Mühl

`gero.muehl@uni-rostock.de`
`http://wwwava.informatik.uni-rostock.de`