Verifying a Distributed System with Combinatorial Topology

CodeMesh 2018

Verónica López
Sr. Software Engineer
@maria_fibonacci

Verifying a Distributed System with Combinatorial Topology

CodeMesh 2018

Verónica López
Sr. Software Engineer
@maria_fibonacci

whoami

- Academy & Industry: From Physics to Distributed Systems
- Software Engineer: Go & Kubernetes, Containers, Linux
- Personal preference: Elixir (BEAM)
- Before: Big Latin American systems: many constraints
- Technology as a means of social progress

Agenda

- Distributed Systems
- Graph Theory
- Topology

Topology: the math term, not the (pretentious) engineer term for any systems design diagram

All these concepts have connectivity in common

Distributed Systems

Famous -and overused- quote about distsys…

“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

Ideal Distributed System

- Fault Tolerant
- Highly available
- Recoverable
- Consistent

- Scalable
- (Predictable)
  Performance
- Secure

Design for Failure

If the probability of something happening is one in 10^13, how often will it really happen?

“Real life”: never

Physics: all the time

Think about servers (infrastructure) at scale
Or in terms of downtime

Verification of a Distributed System

Hard Problem:

- Have control and visibility over all the interconnections of our systems
- Solutions: Monitoring, Chaos Engineering, On-Call rotations, Testing in Production, etc.

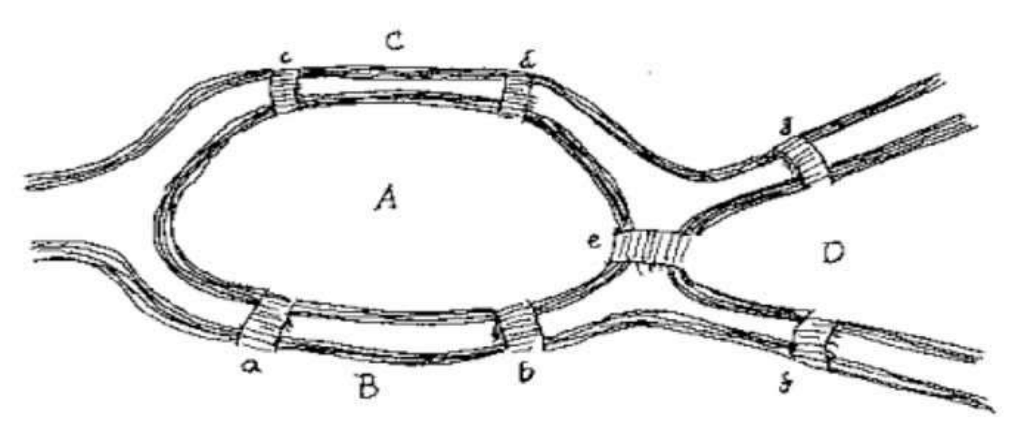
Formal Verification

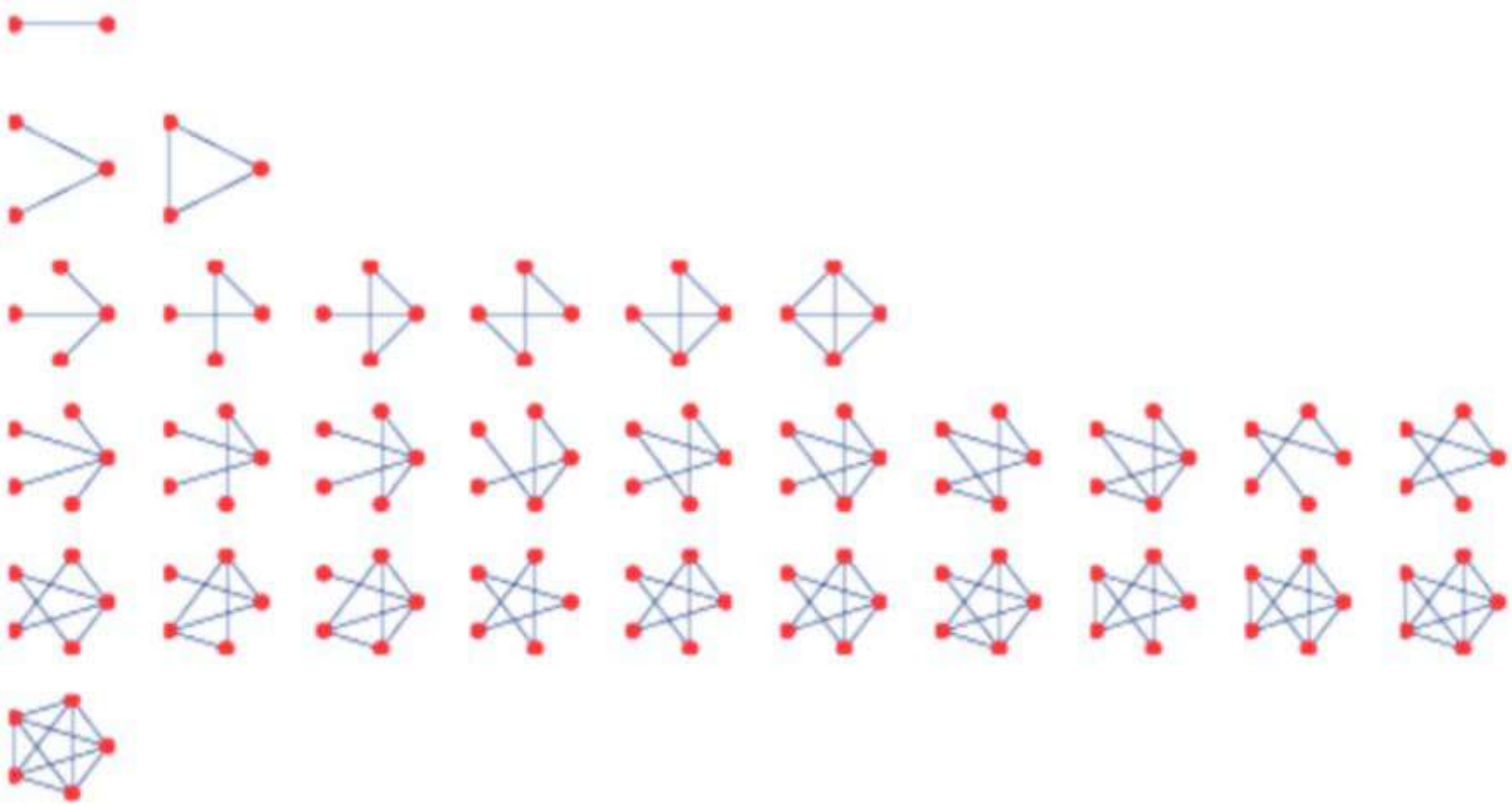
- Formal specification languages & model checkers
- Still requires the definition of the program, possible failures, correctness definitions

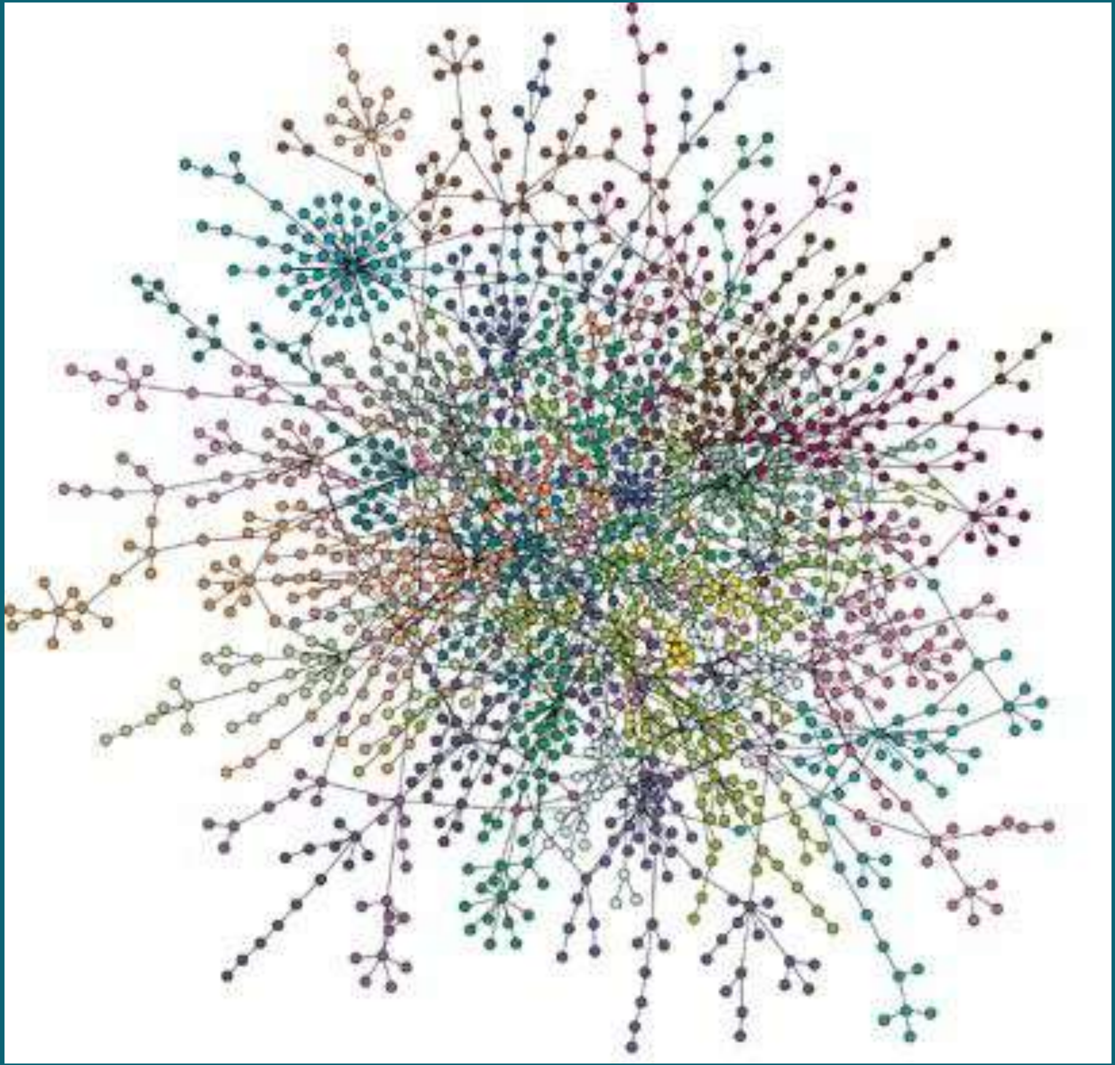
What if we had something that allowed us to see all these possibilities *at once*

Graph Theory

- The mathematical structures used to model pairwise relations between objects.
- Seven Bridges of Könisberg (1736, Euler) is the first paper in history of graph theory
- K-connectedness: how many nodes we need to disconnect a graph (a system)
- Verify points of failure





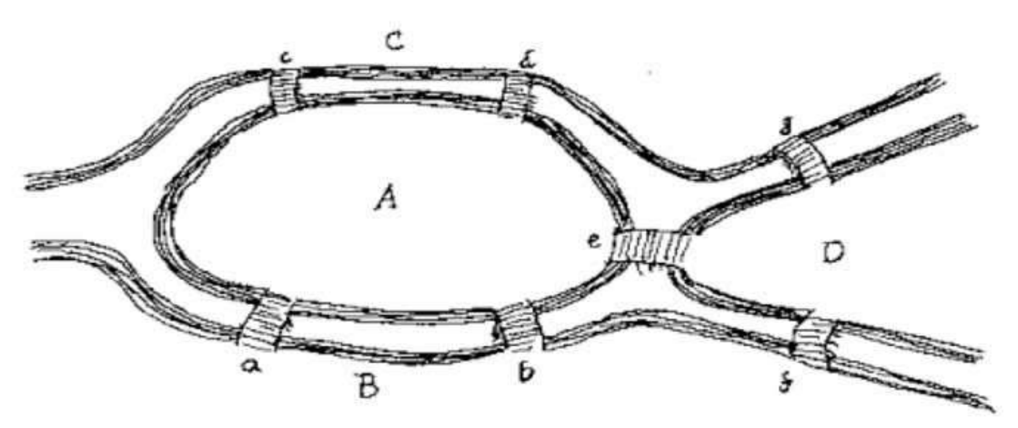


Describing the adjacencies (interactions) of distributed systems gets messier with graphs

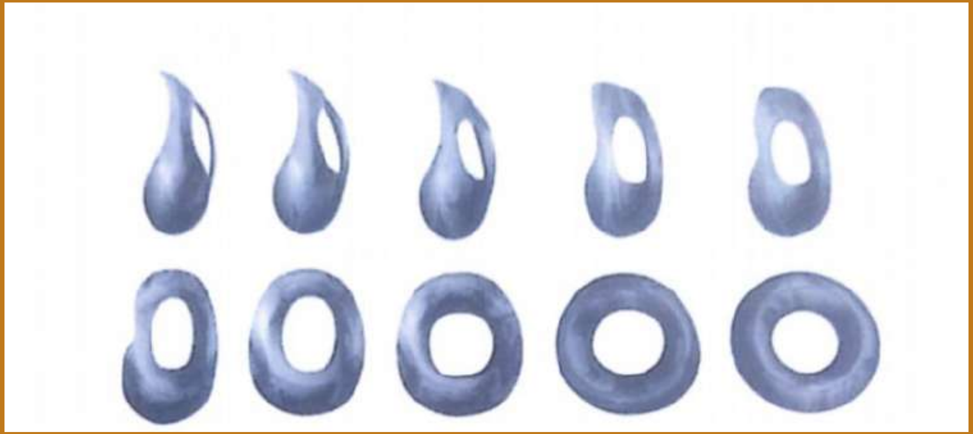
Topology

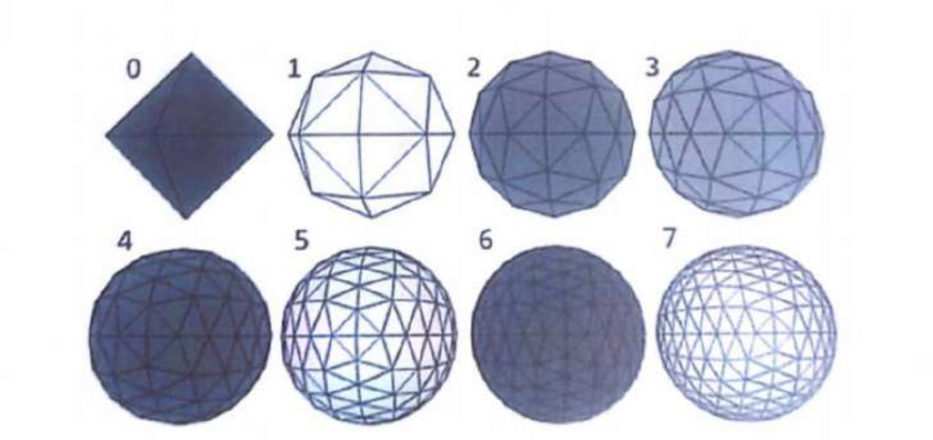
The study of geometric properties and spatial relations unaffected by the continuous change of shape or size of figures.

The paper on the Seven Bridges of Königsberg is also considered the first paper in history of Topology



Properties remain invariant under continuous stretching and bending of the object (different partitions)





Herlihy, Maurice, et al. *Distributed Computing through Combinatorial Topology*. Morgan Kaufmann, 2014.

A topologist is a person who cannot tell the difference between a coffee mug and a donut

A topologist is a person who cannot tell the difference between a coffee mug and a donut

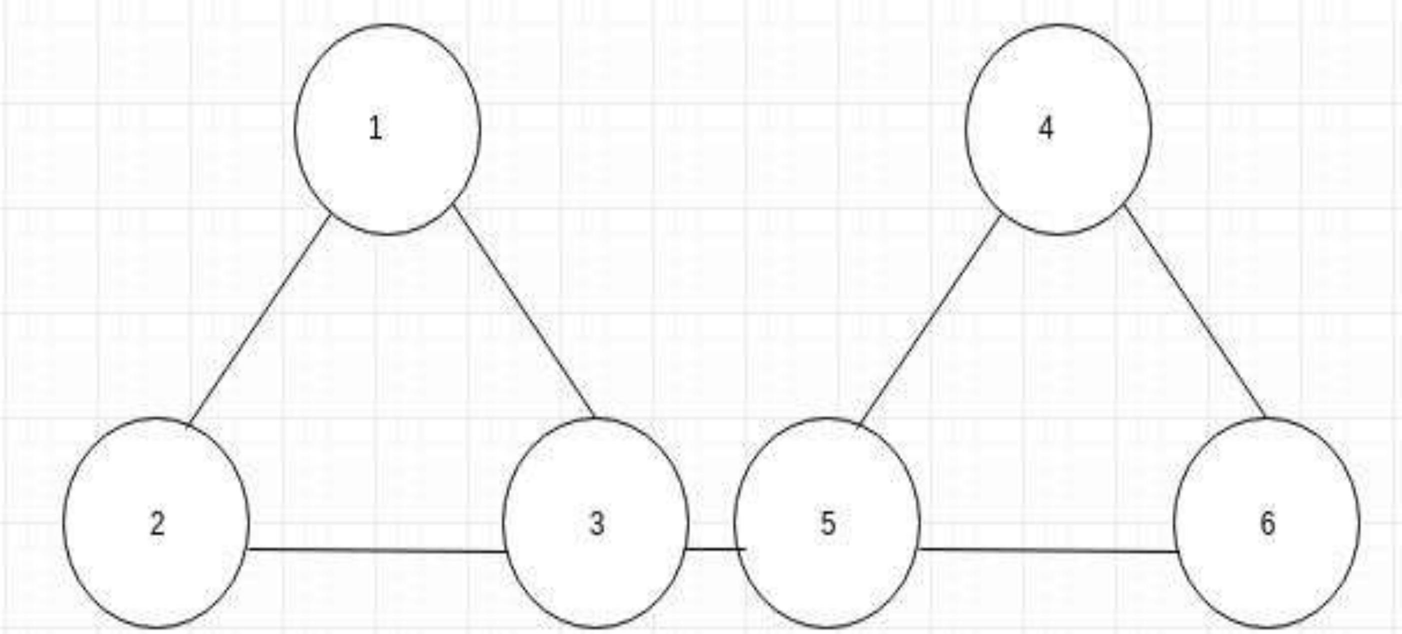
Combinatorial (Algebraic) Topology

- Studies spaces that can be constructed with discretized spaces
- Allows to have all the (system) perspectives (of a node) available at the same time
- Perspectives evolve with communication
- Perspective = the view from a single node

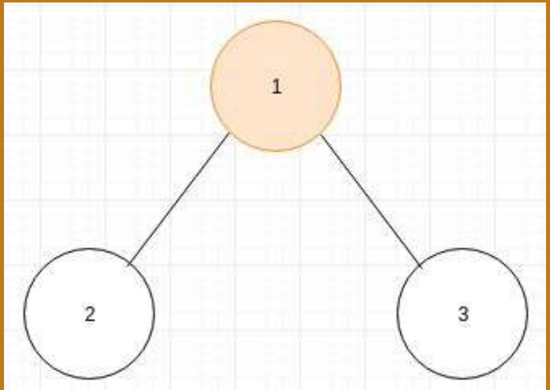
Combinatorial (Algebraic) Topology

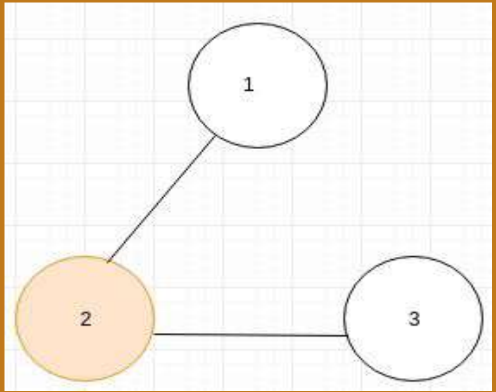
- Branches of topology differ in the way they represent spaces and in the continuous transformations that preserve properties.
- Spaces made up of simple pieces for which essential properties can be characterized by **counting**, such as the sum of the degrees of the nodes in a graph.
- Countable items allow combinations (interactions)

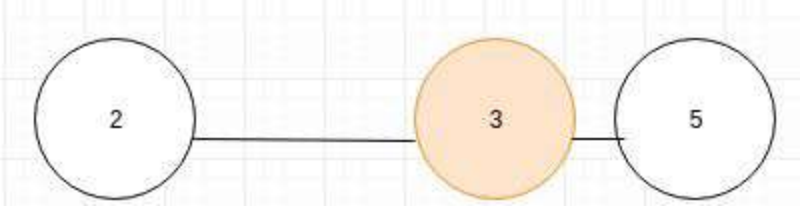
Views: each set of interactions has its own perspective of the system.
Views can be later put together to describe the system.



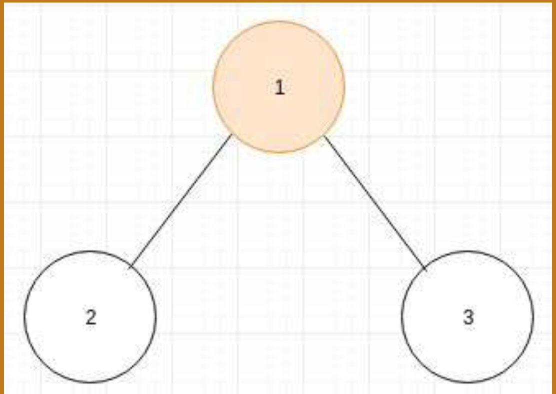
Views: each set of interactions has its own perspective of the system.
Views can be later put together to describe the system.

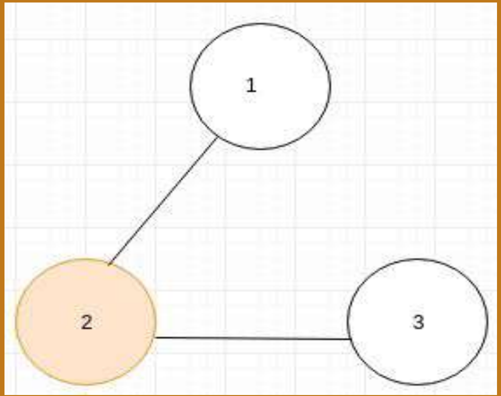


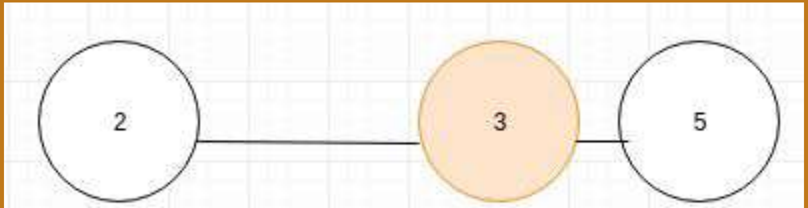




Views: each set of interactions has its own perspective of the system.
Views can be later put together to describe the system.

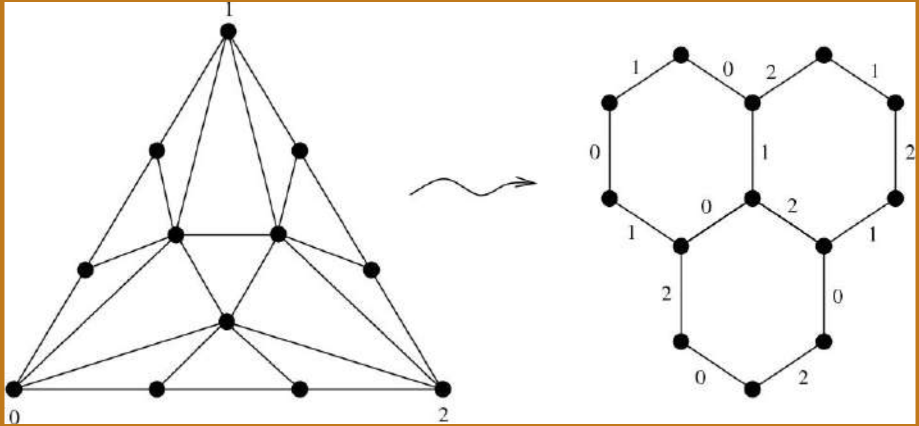


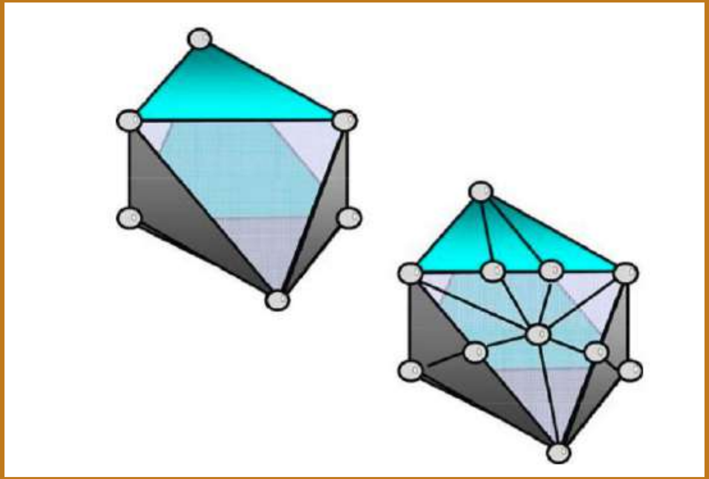




Perspective	Node	Connected to
P1	1	2,3
P2	2	1,3
P3	3	2,5

Subdivisions





Herlihy, Maurice, et al. *Distributed Computing through Combinatorial Topology*. Morgan Kaufmann, 2014.

Not every continuous map A->B has a simplicial approximation.

Verifying a Distributed System with Combinatorial Topology

Thesis

Distributed systems can be formally verified by treating them as (a set of) topological entities that are subject to (valid) subdivisions, analysis of the persistence and consistency of their interconnections (paths), offering a comprehensive set of states of the world

Step 1

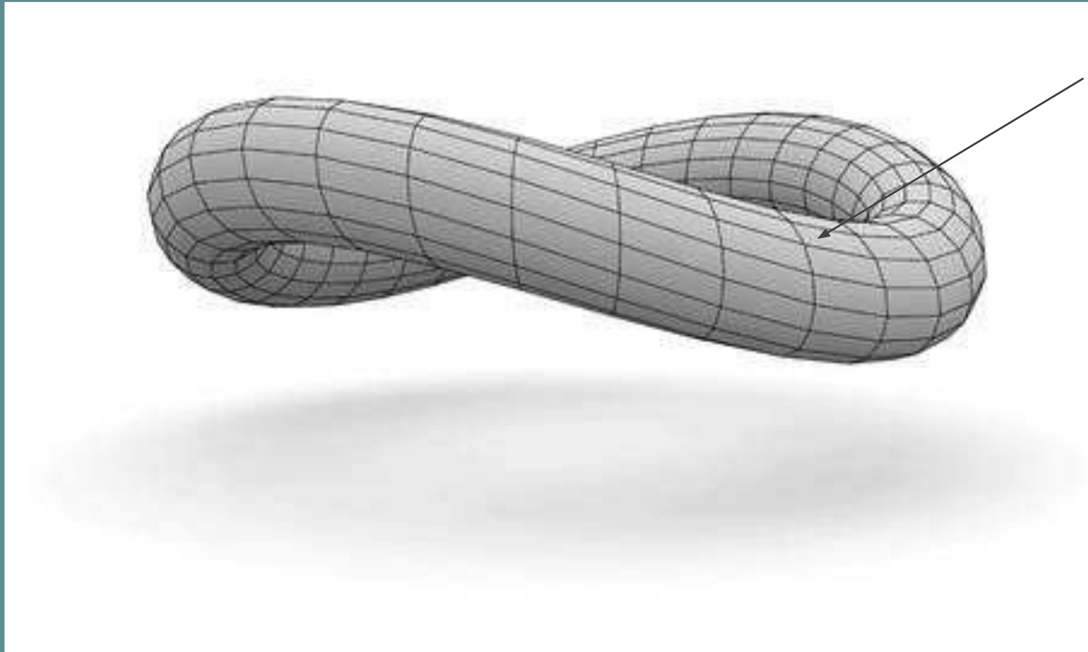
If your system can be described as a graph, it can also be described as a topological object (if the connections are preserved)

Theorem:

A topology on V is compatible with a graph G(V,E) if every induced subgraph of G is connected if and only if its vertex set is topologically connected (too).

Step 2

Describe our systems as a topological object:



Every node is an elemen of our system: compute server, cluster, etc.

Step 3

Prove connectivity -> Verifying the system

Analyze the connections and interactions (in terms of formal Connectivity)

Get all the possible states of the world (use cases; paths)

Once all the connections are topologically correct, we can say that the system is verified.

Resources

1. **Algebraic topology and distributed computing a primer**

https://link.springer.com/chapter/10.1007%2FBFb0015245

2. **The Topology of shared-memory adversaries**

https://dl.acm.org/citation.cfm?doid=1835698.1835724

3. **Distributed Computing Through Combinatorial Topology**

https://www.elsevier.com/books/distributed-computing-through-combinatorial-topology/herlihy/978-0-12-404578-1

Thank you!

whoami

- Academy & Industry: From Physics to Distributed Systems
- Software Engineer: Go & Kubernetes, Containers, Linux
- Personal preference: Elixir (BEAM)
- Before: Big Latin American systems: many constraints
- Technology as a means of social progress

Agenda

- Distributed Systems
- Graph Theory
- Topology

Topology: the math term, not the (pretentious) engineer term for any systems design diagram

All these concepts have connectivity in common

Distributed Systems

Famous -and overused- quote about distsys...

“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

Ideal Distributed System

- Fault Tolerant
- Highly available
- Recoverable
- Consistent

- Scalable
- (Predictable)
  Performance
- Secure

Design for Failure

If the probability of something happening is one in 10^13, how often will it really happen?

“Real life”: never

Physics: all the time

Think about servers (infrastructure) at scale
Or in terms of downtime

Verification of a Distributed System

Hard Problem:

- Have control and visibility over all the interconnections of our systems
- Solutions: Monitoring, Chaos Engineering, On-Call rotations, Testing in Production, etc.

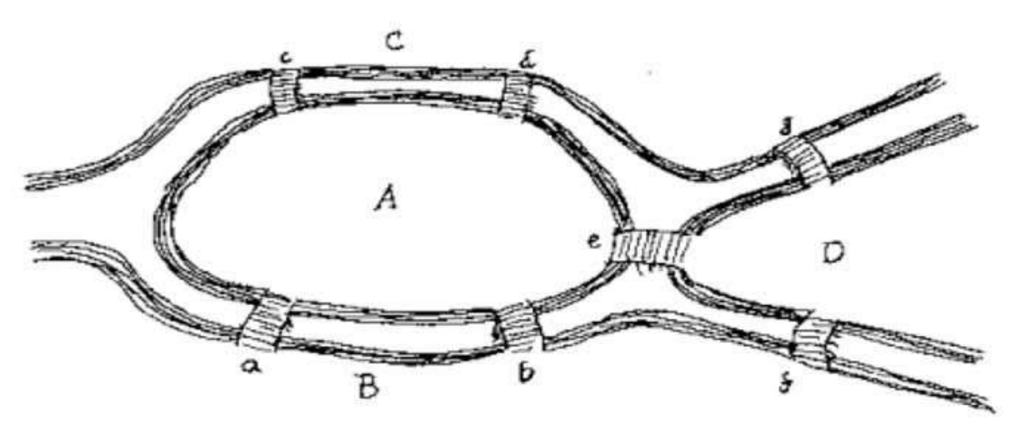
Formal Verification

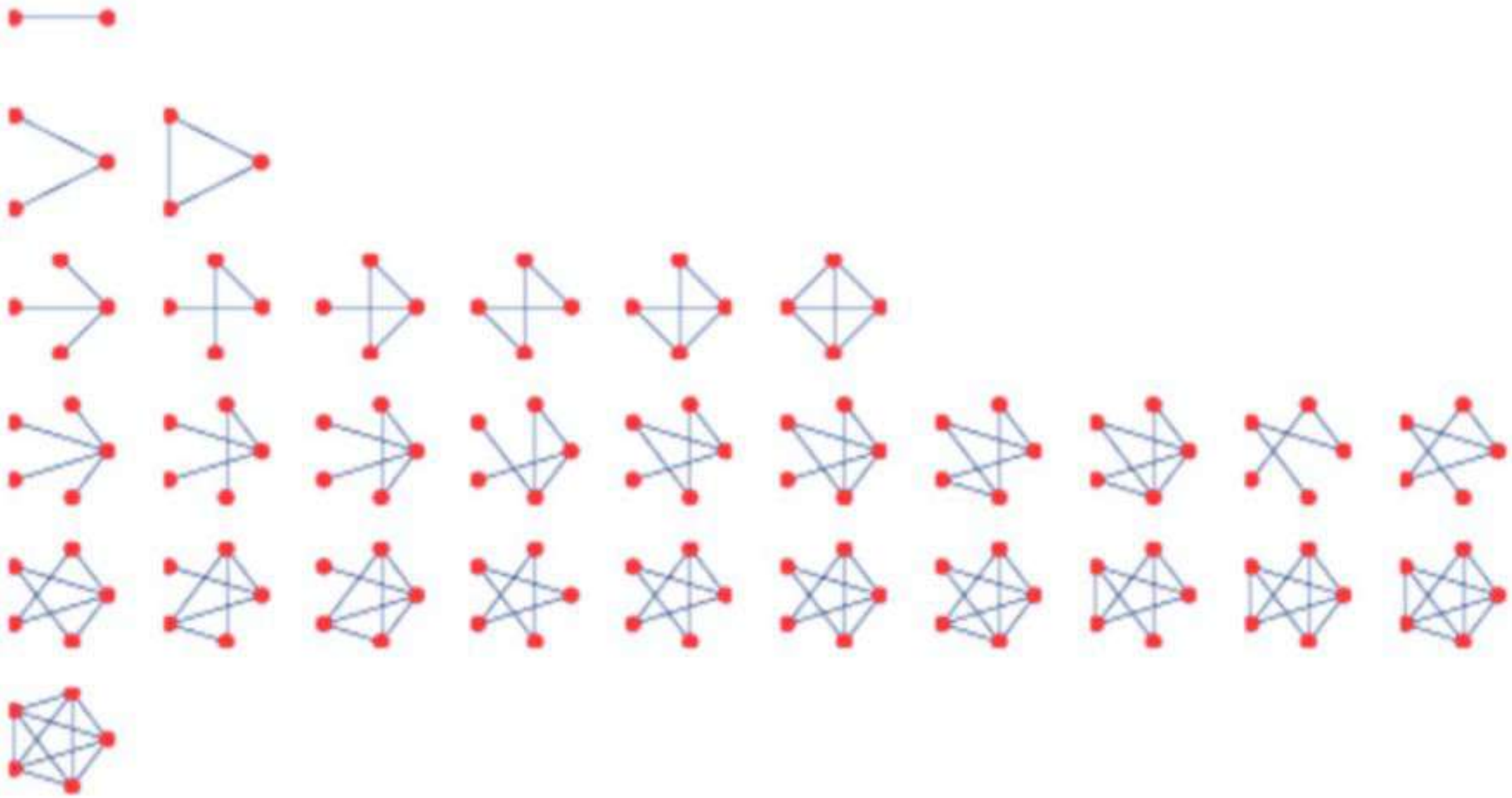
- Formal specification languages & model checkers
- Still requires the definition of the program, possible failures, correctness definitions

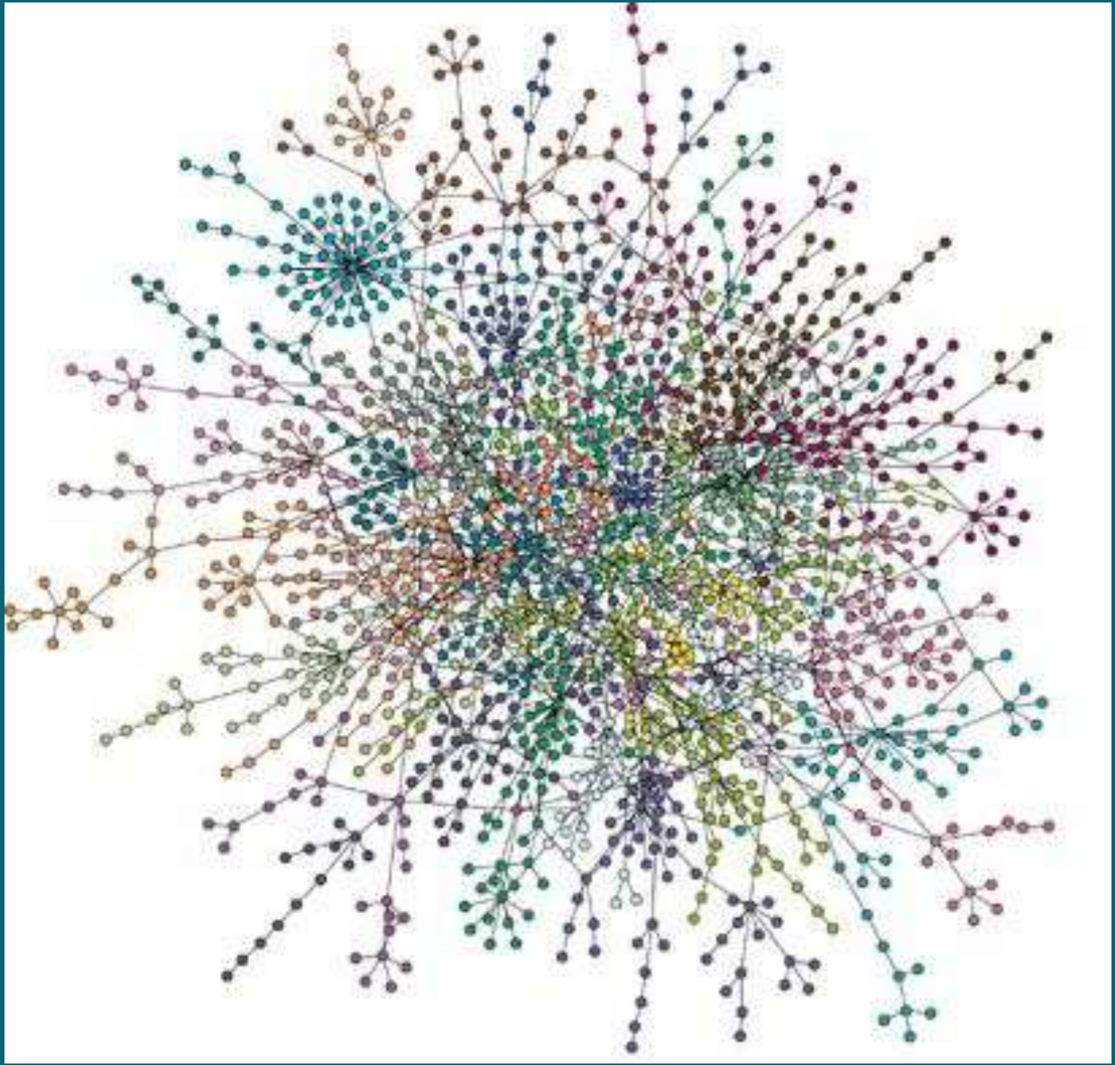
What if we had something that allowed us to see all these possibilities *at once*

Graph Theory

- The mathematical structures used to model pairwise relations between objects.
- Seven Bridges of Könisberg (1736, Euler) is the first paper in history of graph theory
- K-connectedness: how many nodes we need to disconnect a graph (a system)
- Verify points of failure





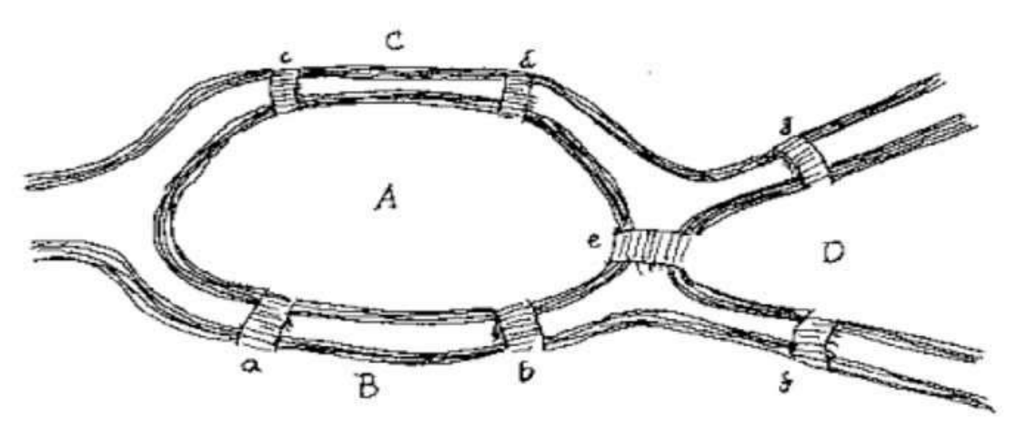


Describing the adjacencies (interactions) of distributed systems gets messier with graphs

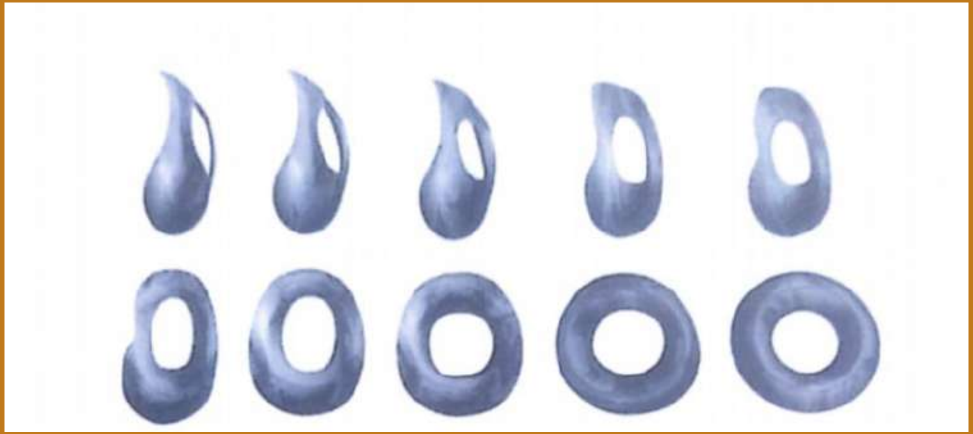
Topology

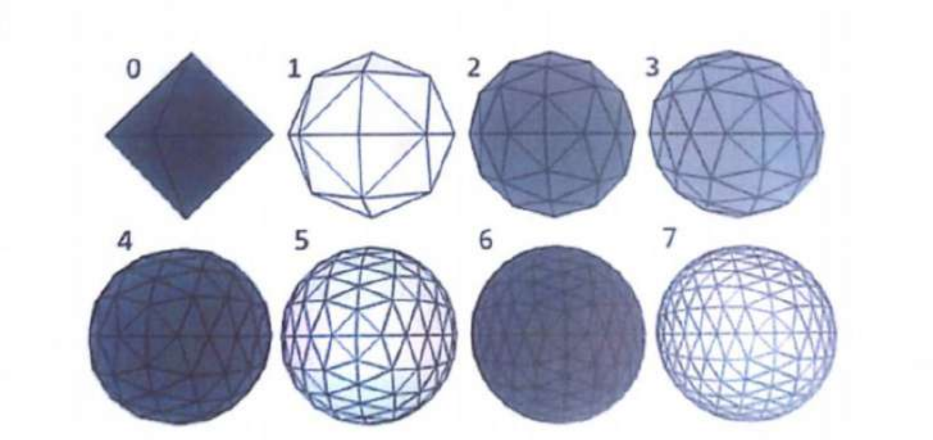
The study of geometric properties and spatial relations unaffected by the continuous change of shape or size of figures.

The paper on the Seven Bridges of Königsberg is also considered the first paper in history of Topology



Properties remain invariant under continuous stretching and bending of the object (different partitions)





Herlihy, Maurice, et al. *Distributed Computing through Combinatorial Topology*. Morgan Kaufmann, 2014.

A topologist is a person who cannot tell the difference between a coffee mug and a donut

A topologist is a person who cannot tell the difference between a coffee mug and a donut

Combinatorial (Algebraic) Topology

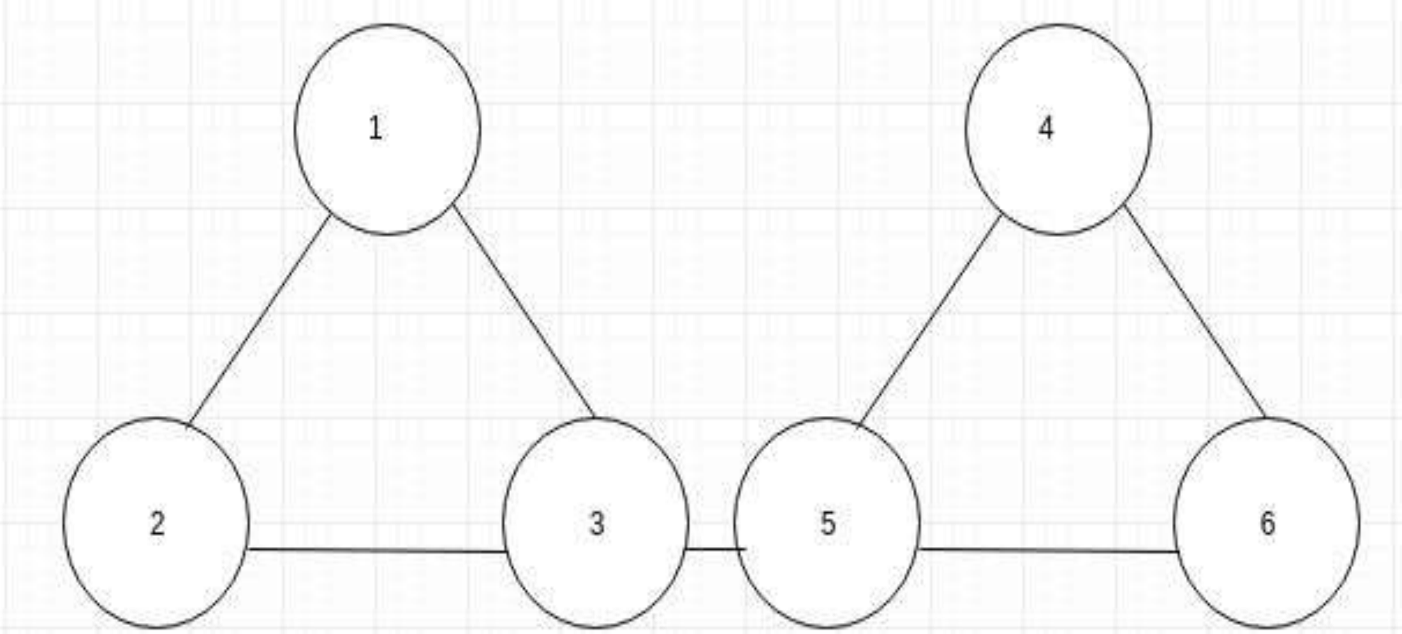
- Studies spaces that can be constructed with discretized spaces
- Allows to have all the (system) perspectives (of a node) available at the same time
- Perspectives evolve with communication
- Perspective = the view from a single node

Combinatorial (Algebraic) Topology

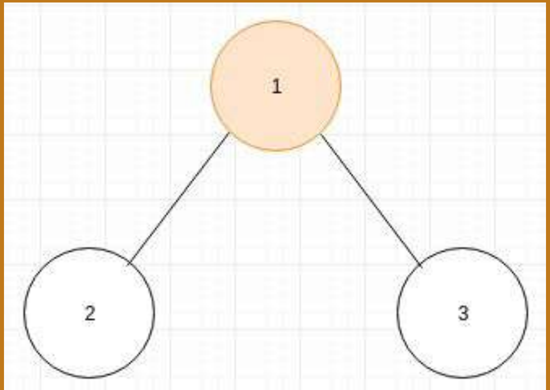
- Branches of topology differ in the way they represent spaces and in the continuous transformations that preserve properties.
- Spaces made up of simple pieces for which essential properties can be characterized by **counting**, such as the sum of the degrees of the nodes in a graph.
- Countable items allow combinations (interactions)

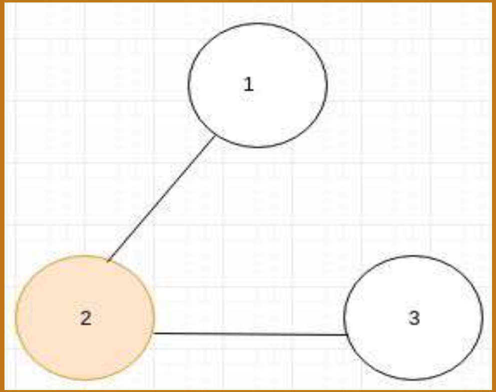
Views: each set of interactions has its own perspective of the system.

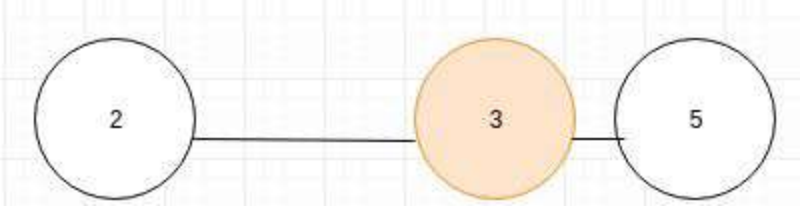
Views can be later put together to describe the system.



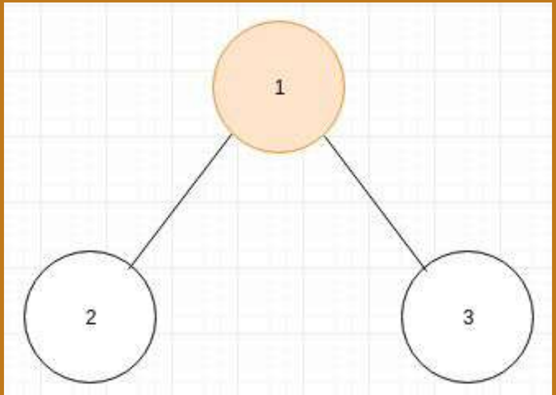
Views: each set of interactions has its own perspective of the system.
Views can be later put together to describe the system.

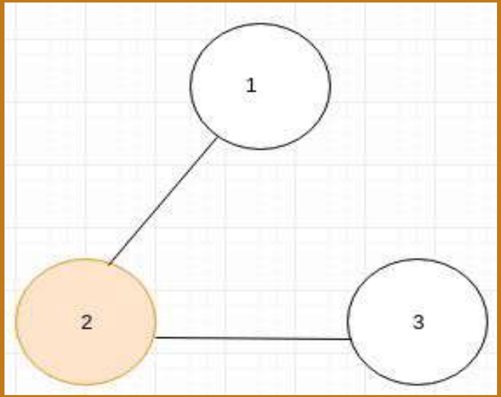


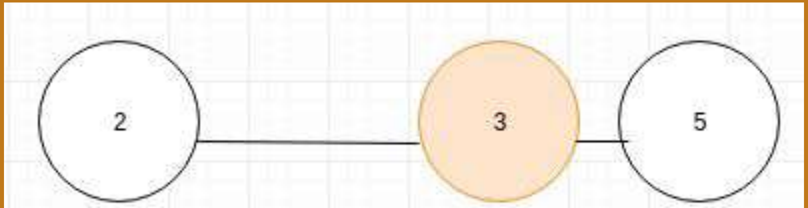




Views: each set of interactions has its own perspective of the system.
Views can be later put together to describe the system.

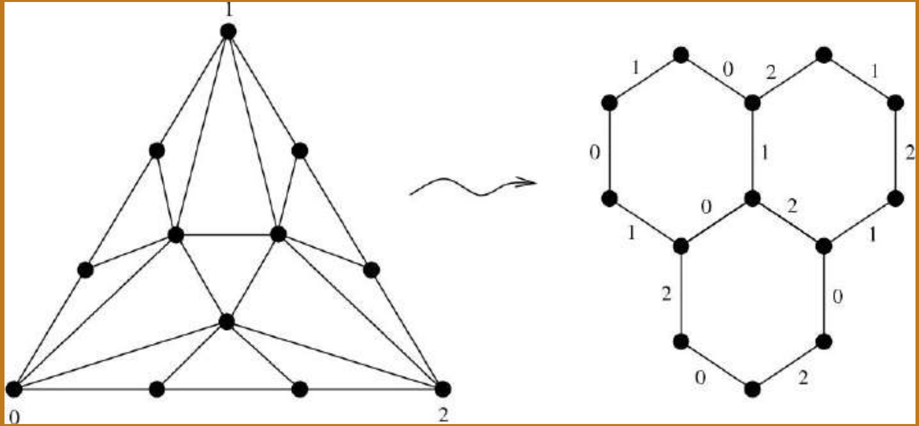


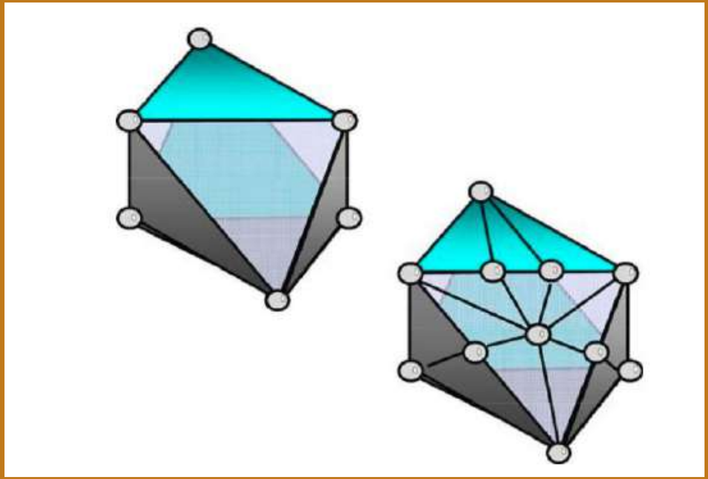




Perspective	Node	Connected to
P1	1	2,3
P2	2	1,3
P3	3	2,5

Subdivisions





Herlihy, Maurice, et al. *Distributed Computing through Combinatorial Topology*. Morgan Kaufmann, 2014.

Not every continuous map A->B has a simplicial approximation.

Verifying a Distributed System with Combinatorial Topology

Thesis

Distributed systems can be formally verified by treating them as (a set of) topological entities that are subject to (valid) subdivisions, analysis of the persistence and consistency of their interconnections (paths), offering a comprehensive set of states of the world

Step 1

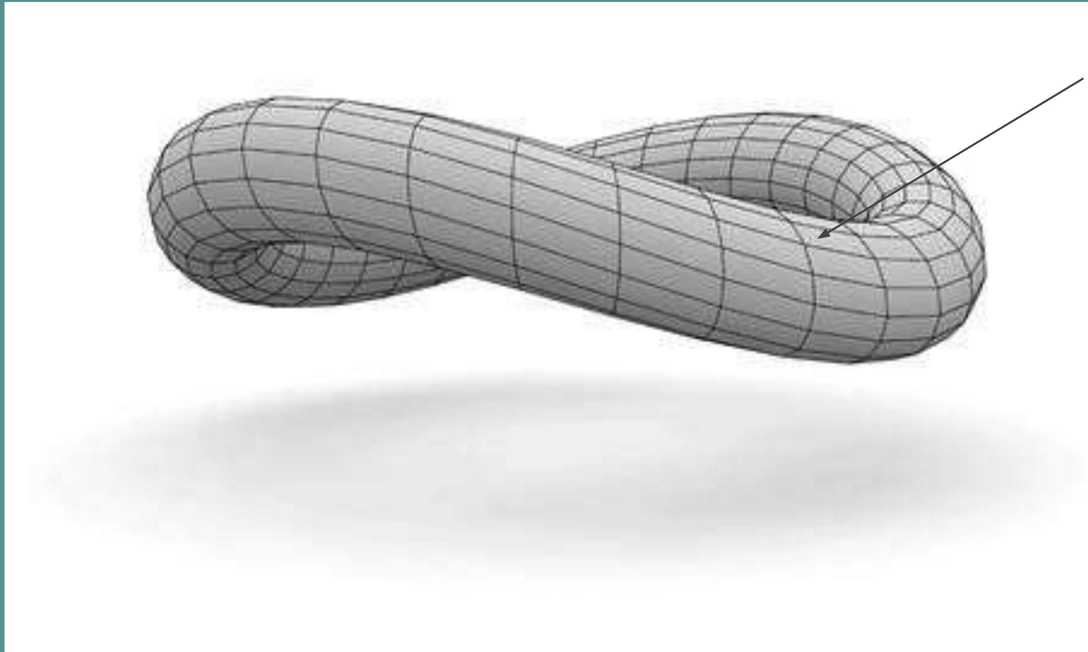
If your system can be described as a graph, it can also be described as a topological object (if the connections are preserved)

Theorem:

A topology on V is compatible with a graph G(V,E) if every induced subgraph of G is connected if and only if its vertex set is topologically connected (too).

Step 2

Describe our systems as a topological object:



Every node is an elemen
 of our system: compute
server, cluster, etc.

Step 3

Prove connectivity -> Verifying the system

Analyze the connections and interactions (in terms of formal Connectivity)

Get all the possible states of the world (use cases; paths)

Once all the connections are topologically correct, we can say that the system is verified.

Resources

1. **Algebraic topology and distributed computing a primer**

https://link.springer.com/chapter/10.1007%2FBFb0015245

2. **The Topology of shared-memory adversaries**

https://dl.acm.org/citation.cfm?doid=1835698.1835724

3. **Distributed Computing Through Combinatorial Topology**

https://www.elsevier.com/books/distributed-computing-through-combinatorial-topology/herlihy/978-0-12-404578-1

Thank you!