3.2 Unstructured P2P architectures

4 Fundamental models (week 4)

Next to physical models we can come up with abstract models or mathematical models, which we all call *fundamental models*.

Such models include for example an

- interaction model
- failure model
- security model.

We will later talk about models for interaction and also about failure models. We will skip security, even though it is a very important aspect in distributed systems.

4.1 Models of a computation

This material is taken from the book by Vijay Garg.

We consider a distributed system that is a loosely-coupled set of N processes $\{P_1, P_2, \ldots, P_N\}$ and a set of loss-free unidirectional communication channels over which communication takes place by message passing. There is no shared memory.

We discuss three different models of computation.

4.1.1 Interleaving model

The interleaving model assumes that a run of a distributed computation is a global sequence of events on one universal time line.

We can define a chronological order through an operation $\text{next}(G, e)$, which is the next state following state G and executing event e.

Definition 2 *Interleaving model*

A sequence of events $seq = (e_i : 0 \leq i \leq m)$ is a computation of the system in the interleaving model if there exists a sequence of global states $(G_i : 0 \leq i \leq m+1)$ such that G_0 is an initial state and

$$G_{i+1} = \textit{ next } (G_i, e_i), \textit{ for } 0 \leq i \leq m.$$

Please, note that the events in this model form a *total global order*, which means that for each pair of events we can determine which one happened next and which one happened before.

4.1.2 Happened-before model

As we have seen, the interleaving model needs a total order. Leslie Lamport was the first to point out that in a distributed system very often only a partial order is given. This means, for some pairs of events we can determine which one happened before and which one after the other, but there are pairs of events for which such statement cannot be made. Since of course there is a strict chronological order, but it is not known to the participants in the distributed system, we need to define a happened-before relation as it appears in the distributed system.

We assume that each process P_i generates a sequence of states $S_{i,0}e_{i,0}S_{i,1}\dots e_{i,l-1}s_{i,l}$. The initial state of process P_i is $s_{i,0}$ and we can write

$$s_{i,j} \quad \xrightarrow{e_{ij}} \quad s_{ij+1}$$

Definition 3 *Immediately precedes. The relation* immediately precedes $\prec_{im}\subseteq E^2$, *where E is the set of all events $e_{i,j}$ is defined as*

$$\prec_{im}: e \prec_{im} f \iff$$

e immediately precedes f in the sequence of events at P_i. Also: next$(e) = F$, and prev$(f) = e$ whenever $e \prec_{im} f$

We can extend the relation *immediately precedes* by formulating its irreflexive (and reflexive) and transitive closure as

Definition 4 *Locally precedes. The relation* locally precedes $\prec\subseteq E^2$, *as well as $\preceq\subseteq E^2$ are defined as irreflexive (and reflexive) and transitive closure of $\prec_{im}$ where for both E is the set of all events $e_{i,j}$.*

We will still mostly use $\prec_{im}$ as the reflexive and transitive extension of the relation will be included in the happened-before model.

To reason about events in different processes, we need an expression for events in processes. If we write $e.p$ this means that event e occurs in process p.

We still need one more relation to be able to formulate the happened-before model completely. This is the relation

Definition 5 *Remotely precedes. The relation* remotely precedes $\rightsquigarrow\subseteq E^2$, *where E is the set of all events e is defined as*

$$\rightsquigarrow: e \rightsquigarrow f \iff$$

e is the send event of a message at process P_i and f is the receive event of the same message at process P_j. Similarly, states s and t are related by $\rightsquigarrow$ iff a message is sent after state s that is received be the receiver, resulting in state t.

Now we come to the main part in the happened-before model. This is the defition of the happened-before relation. As all relations the happened-before relation can be regarded to be a set of pairs of elements from the sets over which the relation is defined.

Definition 6 *Happened-before. The happened before relation $\rightarrow$ is the smallest relation that satisfies*

1. $(e \prec_{im} f) \vee (e \rightsquigarrow f) \Rightarrow (e \rightarrow f)$, *and*
2. $\exists g : (e \rightarrow g) \wedge (g \rightarrow f) \Rightarrow (e \rightarrow f)$.

A run or a computation is a tuple $(E, \rightarrow)$, where E is the set of all events, $\rightarrow$ is a partial order, such that all events in a process are totally ordered.

Remark: happened-before is a partial order, i.e. two events may not be ordered at all.

Definition 7 *Concurrency. We call two events concurrent if none of them happened before the other, i.e.*

$$\neg(e \rightarrow f) \wedge \neg(f \rightarrow e) \Rightarrow e \parallel f$$

4.1.3 Potential causality model

Our last model of distributed computations is the potential causality model. While happened-before introduced total order within each process, potential causality does not even do this. It is merely about the cause-effect relation, which must be chronological. But not every time relation must indicate a cause-effect relationship.

Therefore, causality is a partial order within one process.

We can state that if event e causes event f then event e potentially causes f. The inverse is not necessarily true.

Note, that happened-before is a potential causality model, i.e. if e causes f then e happened-before f. Not necessarily the reverse.

Definition 8 *Potential Causality. The potential causality relation is the smallest causality relation on the event set that satisfies the following properties:*

- *If an event e potentially causes another event f on the same process, then $e \xrightarrow{p} f$.*
- *If e is the sending of a message and f is the corresponding receive event then $e \xrightarrow{p} f$.*
- *if $e \xrightarrow{p} g$ and $g \xrightarrow{p} f$ then $e \xrightarrow{p} f$.*

Events e and f are *independent* if $\neg(e \xrightarrow{p} f)$ and $\neg(f \xrightarrow{p} e)$.

A potential causality diagram is defined as a partially ordered set $(E, \xrightarrow{p})$, where E is a set of events and $\xrightarrow{p}$ is the potential causality relation.

Given $(E, \xrightarrow{p})$ a happened-before diagram is *consistent* with it if $\xrightarrow{p} \subseteq \rightarrow$.

Model	Basis	Type
Interleaving	Physical time	total order on all events
Happened-before	Logical order	total order on each process
Potential causality	Causality	partial order on each process.

Tabelle 1: Properties of the fundamental models