



WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction

Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools

What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export

Create a book
Download as PDF
Printable version

Languages

Deutsch
فارسی

 Edit links

Article **Talk**

Read **Edit** View history


Search 

Snapshot algorithm

From Wikipedia, the free encyclopedia

The **snapshot algorithm** is an [algorithm](#) used in [distributed systems](#) for recording a consistent global state of an [asynchronous](#) system. The algorithm discussed here is also known as the **Chandy–Lamport algorithm**, after [Leslie Lamport](#) and [K. Mani Chandy](#).

History [\[edit\]](#)

According to [Leslie Lamport's website](#) , "The distributed snapshot algorithm described here came about when I visited Chandy, who was then at the [University of Texas in Austin](#). He posed the problem to me over dinner, but we had both had too much wine to think about it right then. The next morning, in the shower, I came up with the solution. When I arrived at Chandy's office, he was waiting for me with the same solution."

It was defined in a paper titled "[Distributed Snapshots: Determining Global States of a Distributed System](#) .

Definition [\[edit\]](#)

The assumptions of the algorithm are as follows:

- There are no failures and all messages arrive intact and only once
- The communication channels are unidirectional and [FIFO](#) ordered
- There is a communication path between any two processes in the system
- Any process may initiate the snapshot algorithm
- The snapshot algorithm does not interfere with the normal execution of the processes
- Each process in the system records its local state and the state of its incoming channels

The algorithm works using marker messages. Each process that wants to initiate a snapshot records its local state and sends a marker on each of its outgoing channels. All the other processes, upon receiving a marker, record their local state, the state of the channel from which the marker just came as empty, and send marker messages on all of their outgoing channels. If a process receives a marker after having recorded its local state, it records the state of the incoming channel from which the marker came as carrying all the messages received since it first recorded its local state.

Some of the assumptions of the algorithm can be facilitated using a more reliable communication protocol such as [TCP/IP](#). The algorithm can be adapted so that there could be multiple snapshots occurring simultaneously.

Algorithm [\[edit\]](#)

The snapshot algorithm works like this:

- The observer process (the process taking a snapshot):
 - Saves its own local state
 - Sends a snapshot request message bearing a snapshot token to all other processes
- A process receiving the snapshot token *for the first time on any message*:
 - Sends the observer process its own saved state
 - Attaches the snapshot token to all subsequent messages (to help propagate the snapshot token)
- Should a process that has already received the snapshot token receive a message that does not bear the snapshot token, this process will forward that message to the observer process. This message was obviously sent before the snapshot "cut off" (as it does not bear a snapshot token and thus must have come from before the snapshot token was sent out) and needs to be included in the snapshot.

From this, the observer builds up a complete snapshot: a saved state for each process and all messages "in the ether" are saved.

Categories: [Distributed algorithms](#)

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

