International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 4 Issue 3, April 2020 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Deadlock in DBMS

Durgesh Raghuvanshi

B Tech, Department of Computer Science,
IILM Academy of Higher Learning, Greater Noida, Uttar Pradesh, India

ABSTRACT

A deadlock occurs when there is a setoff process waiting for a resource held by the other processes in the same set. This paper describes the deadlock detection and prevention using wait for graph and some deadlock resolution algorithms which resolves the deadlock by selecting victims using different criteria. In a multi-process system, deadlock is a situation, which arises in a shared resource environment where a process indefinitely waits for a resource, which is held by some other process, which in turn waiting for a resource held by some other process. To prevent any deadlock situation in the system, the DBMS aggressively inspects all the operations which transactions are about to execute. DBMS inspects operations and analyzes if they can create a deadlock situation. If it finds that a deadlock situation might occur, then that transaction is never allowed to be executed.

KEYWORDS: *Database management system (DBMS), allocation, planning, super key*

How to cite this paper: Durgesh Raghuvanshi "Deadlock in DBMS" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-4 | Issue-3, April 2020, pp.795-796, URL: www.ijtsrd.com/papers/ijtsrd24064.pdf

IJTSRD24064

INTRODUCTION

The software used to manage and manipulate that structured information is called a DBMS (Database Management System). A database is one component of a DBMS. You can think of a database simply as a list of information.

A fine example is the white pages of the phone book. Each listing in the white pages contains several items of information – name, address and phone number – about each phone subscriber in a particular region (information). All subscriber information share the same form (structure). In database terms, the white pages comprise a table in which each subscriber is represented by a record. Each subscriber record contains three fields: name, address, and phone number. The records are sorted alphabetically by the name field, which is called the key field.

Other examples of databases are membership/customer lists, library catalogs, and web page content. The list is, in fact, infinite. You can model and design a database to store anything which can be represented as structured information.

Entities and relationships in DBMS

Entities are the things in the real world that we will store information about the database. For example, we might choose to store information about employees and the departments they work for. In this case, an employee would be one entity and a department would be another. Relationships are the links between these entities. For example, an employee works for a department. Works-for is the relationship between the employee and department entities.

Relationships come in different degrees. They can be one-to-one, one-to-many (or many-to-one depending on the direction you are looking at it from), or many-to-many. A one-to-one relationship connects exactly two entities. If employees in this organization had a cubicle each, this would be a one-to-one relationship. The works-for relationship is usually a many-to-one relationship in this example. That is, many employees work for a single department, but each employee works for only one department.

Deadlock

- The Deadlock Problem
- System Model
- Deadlock Characterization
- Methods for Handling Deadlocks
- Deadlock Prevention
- Deadlock Avoidance
- Deadlock Detection
- Recovery from Deadlock

The deadlock problem:

A set of blocked processes, each holding a resource and waiting to acquire a resource held by another process in the set.

Example

- The system has 2 tape drives.
- P0 and P1 each hold one tape drive and each needs another one."

Example

- Semaphores A and B, initialized to 1" P0 " " P1" wait (A); wait (B) wait (B); wait (A)

System model

Resource types R1, R2, . . ., Rm CPU cycles, memory space, I/O devices!

- Each resource type Ri has Wi instances.
- Each process utilizes a resource as follows:
- request
- use
- release

Dedlock characterization

Deadlock can arise if four conditions hold simultaneously."

- Mutual exclusion: only one process at a time can use a resource."
- Hold and wait: a process holding at least one resource is waiting to acquire additional resources held by other processes."
- No preemption: a resource can be released only voluntarily by the process holding it after that process has completed its task."
- Circular wait: there exists a set {P0, P1, …, P0} of waiting processes such that P0 is waiting for a resource that is held by P1, P1 is waiting for a resource that is held by " !P2, …, Pn–1 is waiting for a resource that is held by Pn, and P0 is waiting for a resource that is held by P0.

DEADLOCK PREVENTION

Restrain the ways request can be made."

- Mutual Exclusion – not required for sharable resources; must hold for nonsharable resources.
- Hold and Wait – must guarantee that whenever a process requests a resource, it does not hold any other resources." Require process to request and be allocated all its resources before it begins execution, or allow the process to request resources only when the process has no one. Low resource utilization; starvation possible.
- No Preemption – If a process that is holding some resources requests another resource that cannot be immediately allocated to it, then all resources currently being held are released. Preempted resources are added to the list of resources for which the process is waiting. The process will be restarted only when it can regain its old resources, as well as the new ones that it is requesting.
- Circular Wait – impose a total ordering of all resource types, and require that each process requests resources in increasing order of enumeration."

DEADLOCK AVOIDANCE

Requires that the system has some additional a priori information available."

- The simplest and most useful model requires that each process declare the maximum number of resources of each type that it may need.
- The deadlock-avoidance algorithm dynamically examines the resource-allocation state to ensure that there can never be a circular-wait condition.
- Resource-allocation state is defined by the number of available and allocated resources and the maximum demands of the processes."

CONCLUSION

Here we conclude the paper with various features of deadlock in DBMS. This paper basically concludes with the multiprocess system, deadlock is a situation which arises in shared resource environment process indefinitely waits for a resource which is held by some other process which in turn waiting on a resource by some other process. To prevent any deadlock situation aggressively works. in a database, a deadlock is an unwanted situation in which two or more transactions are waiting indefinitely for one another to give up locks.

REFERENCE

[1] Joachim Biskup, Vacherie Informatik Universitat, Germany

[2] http://www.tutorialspoint.com/dbmd/dbms-deadlock.html

[3] Dr. VK Saraswat, member Niti Ayog, India

[4] Organization la francophonie, South Africa