Understanding the common UNABLE TO LOCK ROW issue with bulk data job

November 01, 2015

Every time a record is inserted or updated, Salesforce must lock the target records that are selected for each lookup field; this practice ensures that, when when the data is committed to the database, its integrity is maintained.

Learn how to avoid the "Unable to lock row" error while uploading large records in Salesforce. Below, are some sample scenarios to help you better understand the issue and how to resolve it:

Sample scenario :

Here, the "Order Detail" object is the main object on which we want to perform a DML operation and "Order__c" is the parent of "Order Detail" object.

A. Process:

---

1. User can make an upsert/insert call on "Order Detail" object, lets take an example of upsert.

2. Wherein external ID is "Order_Detail_Id__c"

B. What's happening at the back end?

---

1. When we do an Upsert call on "Order Detail" object, it updates "Order__c" object as well.

2. Order__c is parent of "Order Detail."

3. "Order__c" can have Triggers/Workflows as associated with it. These triggers/workflows process as event of "after/before update/insert."

4. When we load bulk "order Detail" it updates parent object records as well resulting in lock on child and parent both. This lock get released in micro seconds though.

*Due to large number of child records, two child simultaneously update same parent which in turn generates "Row Lock,". In a layman language we can say that it is a kind of Dead lock.

Example:

Record	ParentRecord (Order__c)	Result
A	X	Succeed
B	X	Succeed
C	Y	Succeed
D	Z	Succeed
E	X	Locked

*Suppose Record is "Order Detail" and object is ParentRecord is Order__c

Here, A, B and E have same parent X, A & B locks X, C locks Y, and D locks Z. Because the process is parallel and happens within micro second, E also tried to lock X however it is already locked by A & B.

C. Why X (Parent Record) is taking time to release?

If X has a Trigger or Workflow and because we are Upserting A, it will update X (We'll discuss why it updates X later), so the Trigger also fires and lock takes some time to get released. At the same time there is another request made by E which causes an error. It'll never generate an error if it's in sorted mode as parent record. Like mentioned below:

Record	ParentRecord(Order__c)
A	X
B	X
E	X
C	Y
D	Z

D. Why A,B and E are updating X (Parent record)?

Parent might have Roll-Up Summary field for child "Order Detail," so whenever you insert or update the child Roll-up Summary will fire, which calls the parent trigger.
[** There might be other components of child as well like child object's trigger or its workflows which can play a role in calling the parent's trigger]

E. Solutions to resolve this issue:

---

1. You can either reduce the batch size and try again or create separate smaller files to be imported if this issue keep occurring.

2. "Unable to lock row error" will occur when concurrent Users are trying to access the same record or parent record. To resolve this error User needs to change the "Concurrency Mode" of processing batches. It should be changed from "Parallel" to "Serial" mode. If you go with serial mode it will be resolved.

3. [Most concrete]Sorting main records as per parent records, or you can order by your records as per parent records. In this scenario, you can take this example with SOQL query, which sorts your main records [Select id, name, so on.. fields from Order_Detail__c order by Order__c]. It processes records for a particular parent record. Actually it divides in chunks and processes main records (which has 1 parent) at a time.

It will be in this format:

Record	ParentRecord (Order__c)	Result
A	X	Succeed
B	X	Succeed
E	X	Succeed
C	Y	Succeed
D	Z	Succeed