بسم الله الرحمن الرحيم

Database Programming

برمجة قواعد البيانات

عدد الساعات: ٢ نظري+٢عملي

الرمز: ٣١٤ حسب

المتطلبات: ٢٢٣ حسب (مبادئ قواعد البيانات)

أستاذات/المادة: م. لندا عمر البدري

م. نجلاء حسن

Lecture 11

- تابع:تقنيات التحكم التزامني

Concurrency control Technique

Dealing with Deadlock (الجمود)

- تقنيات استعادة قواعد البيانات

Database Recovery Techniques

الجمود
Dealing with Deadlock

الجمود deadlock

- تحدث هذه الحالة عندما تكون كل معاملة T ضمن مجموعة من أثنين او أكثر من المعاملات ، تنتظر عنصر بيانات تم قفله أو تأمينه من قبل معاملة اخري T’ في نفس المجموعة و T’ في صف الانتظار من أجل انتظار معاملة اخري ان تفتح تأمين عنصر بيانات آخر .
- يحدث هذا الشرط عندما تكون معاملتين ، تنتظر أحداهما الأخرى لفتح قفل البيانات .

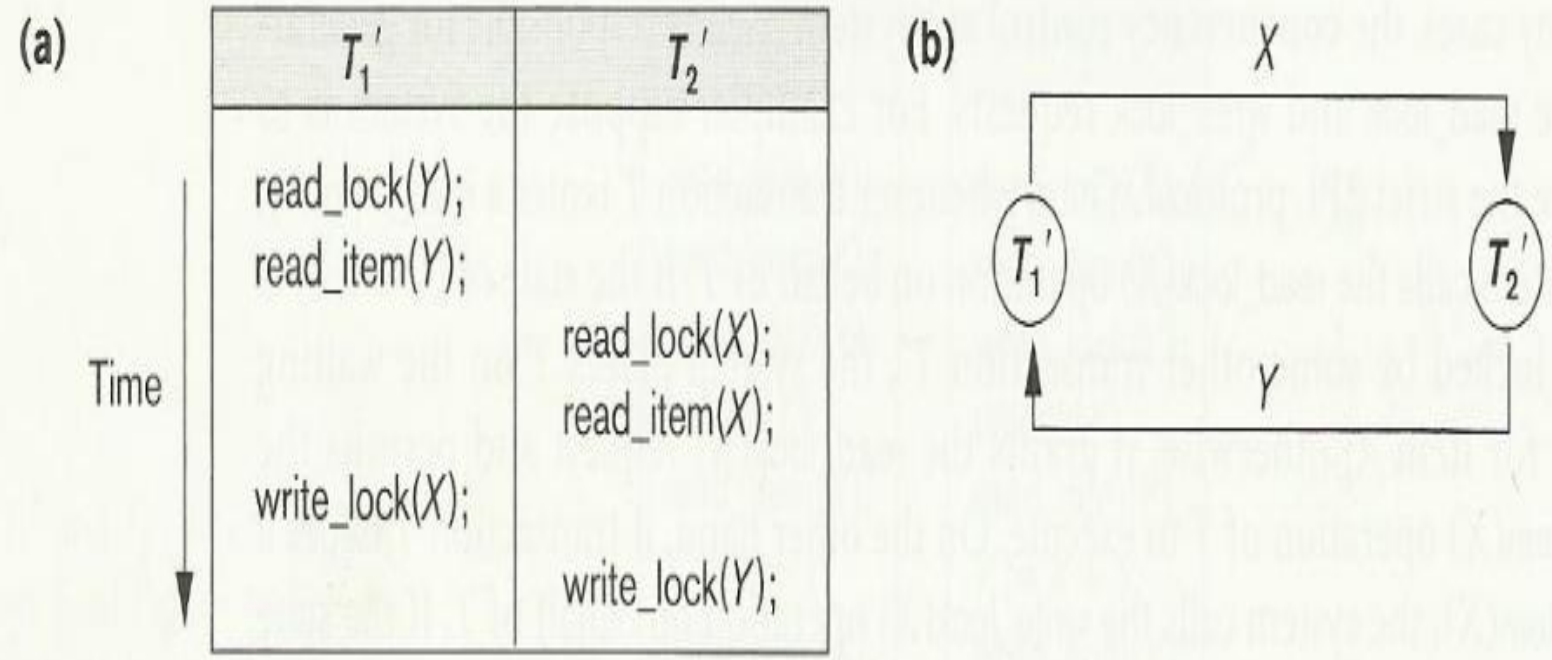

Figure 18.5

Illustrating the deadlock problem. (a) A partial schedule of T_1' and T_2' that is in a state of deadlock. (b) A wait-for graph for the partial schedule in (a).

بروتوكولات تجنب الجمود
Deadlock prevention protocols

- تستخدم هذه البروتوكولات لتجنب حالة الجمود deadlock.
- أحد هذه البروتوكولات والذي يستخدم مع تأمين المرحلتين يقوم على فكرة أن تقوم كل معاملة بتأمين جميع عناصر البيانات التي ستحتاجها منذ البداية ، وإذا لم تتمكن من إيجاد أيً من هذه العناصر فإنها لن تغلق أي عنصر بيانات ، وبالتأكيد هذا يقلل من تزامن المعاملات .

الطبعة الزمنية Time stamp(TS)

- توجد تقنية أخرى لتجنب الجمود تفترض ان يتم إتخاذ قرار ما في حالة التعامل مع معاملات يمكن أن يحدث لها deadlock (هل يتم إيقاف المعاملة وجعلها تنتظر أو يتم إجهاضها) .
- هذه التقنية تستخدم مفهوم الطبعة الزمنية Time stamp(TS) .
- الطبعة الزمنية Time stamp(TS): هي معرف فريد (طبعة زمنية) يخصص لكل معاملة وهذا المعرف يعتمد على ترتيب بداية كل معاملة ..
- للطبعة الزمنية صفتين:التفرديةوالتسلسل.
- مثلاً : إذا كانت T1 تبدا التنفيذ قبل T2 فهذا يعني أن TS(T1)<TS(T2) .(المعاملة الأقدم لها قيمة TS أقل).

توجد طريقتين لتجنب الdeadlock بناءً على مفهوم Timestamp

- Wait-die : اذا كانت (TS (Ti)<TS (Tj) وهذا يعني أن Ti أقدم من Tj ففي هذه الحالة يُسمح ل Ti بالإنتظار ، وإلا (Ti أحدث من Tj) وعندها يتم إجهاض Ti (Ti dies) ويتم إعادتها مرة أخرى لاحقاً ومنحها نفس قيمة ال TS .
- Wound-wait : اذا كانت (TS (Ti)<TS (Tj) وهذا يعني أن Ti أقدم من Tj ففي هذه الحالة يتم إجهاض Tj.

(Ti wounds Tj) وتعاد مرة اخرى لاحقاً مع إعطائها نفس قيمة ال TS ، وإلا (Ti أحدث من Tj) وفي هذه الحالة يُسمح ل Ti بالإنتظار .

- في كلتا الطريقتين يتم اجهاض المعاملة الأحدث والتي تتسبب في حالة الdeadlock.وبذلك لا تدخل المعاملتين في «دائرة انتظار»


تقنيات استعادة قواعد البيانات
Database Recovery Techniques
© TemplatesWise.com

مفهوم الإستعادة Recovery concept

- نقصد باستعادة قواعد البيانات من فشل المعاملات أن يتم اعادتها الي حالة التوافق والثبات التي كانت عليها قبل تنفيذ المعاملات الفاشلة .
- حتى يتمكن النظام من القيام بذلك يجب أن يحتفظ بمعلومات عن التغيرات التي يتم اجراءها على عناصر قاعدة البيانات من قبل المعاملات .
- هذه المعلومات تخزن في سجل النظام System Log .

يمكن تلخيص الاستراتيجية المستخدمة لاستعادة قاعدة البيانات بالأتي :

- إذا حدث ضرر شامل لجزء كبير من قاعدة البيانات نتيجة لفشل كبير (مثل تعطل قرص التخزين) ، ففي هذه الحالة يمكن استعادة قاعدة البيانات عن طريق نسخة احتياطية مخزنة في مكان أخر (نسخة أرشيفية) ، ثم محاولة إعادة قاعدة البيانات الى حالتها التي كانت عليها قبل حدوث آخر معاملة أدت الى فشل قاعدة البيانات .
- أذا كانت قاعدة البيانات غير متضررة بشكل كبير ولكن وصلت الى حالة عدم تناغم فالطريقة المتبعة في هذه الحالة لاستعادة قاعدة البيانات ، هو التراجع عن التغيرات التي كانت السبب في حدوث حالة عدم التناغم عن طريق التراجع عن بعض العمليات . وفي هذه الحالة لا نحتاج الى نسخة ارشيفية ولكن يمكن أن نحتاج الى الرجوع الي سجل النظام.

يمكن التمييز بين نوعيين من التقنيات المستخدمة للاستعادة (من non catastrophic failure)

١. تقنية التحديث المُرْجَأ (المؤجل)deferred update technique :

- لا يتم تحديث قاعدة البيانات المخزنة في قرص التخزين الى أن تصل المعاملة الى نقطة التزام commit point ، وعندها فقط يتم تسجيل التحديث في قاعدة البيانات .
- قبل الوصول الي commit point يتم حفظ التحديثات التي تقوم بها المعاملة في وسائط تخزين مؤقتة buffer.
- في هذه الأثناء فأن تحديثات المعاملة يتم حفظها اولاً في السجل (Log) ، قبل تسجيلها على قاعدة البيانات .
- إذا فشلت المعاملة قبل الوصول الي commit point ، لا يكون لها تأثير على قاعدة البيانات ، وبالتالي لا نكون بحاجة الى إجراء UNDO .
- في هذه الحالة يكون من المهم إجراء عملية Redo على العمليات المخزنة في السجل Log ، لأن تأثيرها لم يتم تسجيله على قاعدة البيانات .
- لذلك يطلق على هذه التقنية أيضاً أسم No-Undo /Redo algorithm .

٢. تقنية التحديث الفوري immediate update technique :

- يتم تحديث قاعدة البيانات عن طريق بعض عمليات معاملة قبل وصولها الى save point .
- تسجيل هذه العمليات في السجل "log" قبل تطبيقها على قاعدة البيانات يجعل من الممكن استعادة قاعدة البيانات فيما بعد .
- اذا حدث فشل لاحد المعاملات التي أثرت على قاعدة البيانات قبل الوصول الى save point ، فسيتم التراجع عن هذه العمليات بمعنى تجري عملية rollback على المعاملة .
- بشكل عام فإن هذه التقنية تتطلب اجراء UNDO ، و REDO من اجل استعادة البيانات .
- هي التقنية الأكثر أستخداماً .
- هناك بعض خوارزميات الإستعادة التي تستخدم تقنية التحديث الفوري تتطلب إجراء عملية UNDO فقط لذلك تعرف باسم UNDO/NO-REDO algorithm .

استخدام الذاكرة المؤقتة
Caching(Buffering) of disk block

- عملية استعادة قواعد البيانات تتداخل بشكل كبير مع مهام نظام التشغيل (operating system functions) ، وخاصةً فيما يتعلق بالتخزين المؤقت في الذاكرة الرئيسية .
- عادةً يتم تخزين بعض صفحات قرص التخزين (disk pages) والتي تحتوي على عناصر قاعدة البيانات التي سيتم تحديثها بشكل مؤقت (cached) في الذاكرة المؤقتة main memory buffer ثم يتم تحديثها قبل أن يتم نقلها مرة أخرى الى قرص التخزين الدائم .
- سابقاً كانت هذه العملية تعتبر من مهام نظام التشغيل ، أما الان ونظراً لأهمية استعادة قواعد البيانات فان هذه العملية يقوم بها ال DBMS عن طريق استدعاء Low-Level operating routine

- من المناسب اعتبار أن عملية الاستعادة تتم على اجزاء من قاعدة البيانات تعرف بأسم صفحات قرص التخزين (Database disk pages “Block”) ، .
- عادة ما تترك بعض اجزاء الذاكرة المؤقتة (يطلق عليه إسم DBMS cache) تحت سيطرة وتحكم ال DBMS من أجل أغراض الإستعادة .
- يترك دليل (directory) من ال cache لتحديد أيً من عناصر البيانات يتم تخزينه حالياً في الذاكرة cache .
- اذا طلب الDBMS اجراء عملية ما على عنصر فعندها سيتم التأكد من ال cache directory من وجود العنصر ضمن العناصر الموجودة في ال cache .

- إذا لم يكن العنصر موجود سيتم نقل العنصر الي ال cache عن طريق اخذ نسخة من صفحة القرص (block) التي تحتوي على العنصر . هذا يتطلب احلال (replace) العنصر الجديد ببعض العناصر الموجودة اصلاً في ال cache اذا لم تتوفر مساحة تخزينية خالية .
- تستخدم لعملية الإحلال هذه بعض الاستراتيجيات المتبعة من قبل نظم التشغيل مثل Last Recently Used(LRU) ، أو First – In –First –Out (FIFO) .
- كل buffer في ال Cache له مايعرف باسم ال dirty-bit مرتبط بأي محتوى من محتويات الcache يدل على ما اذا كانت محتويات الbuffer قد اجري عليها تعديل أم لا .
- في البداية تكون قيمة ال dirty-bit تساوي 0 ، ويتم تغيرها الى 1 عند اجراء اول عملية تحديث على محتويات ال buffer

- إذا حدثت عملية إحلال (Replaced or flushed) لمحتويات الbuffer من ال Cache فان هذه المحتويات سيتم كتابتها مرة اخري في قرص التخزين الدائم ، وذلك فقط إذا كانت قيمة ال dirty-bit تساوي 1 .

طرق احلال محتويات ال buffer بعد تعديلها ونقلها الى الdisk:

flushing a modified buffer back to disk

١. تحديث في نفس الموقع In- place update : يتم نقل محتويات الbuffer الي نفس المكان السابق بمعني ان يتم الكتابة فوق البيانات القديمة (overwriting) وبذلك تحتفظ قاعدة البيانات بنسخة واحدة فقط من ال block .

٢. التظليل Shadowing : يتم كتابة محتويات الbuffer المحدثة في مكان مختلف عن المكان الاصلي للبيانات ، وبذلك تكون قاعدة البيانات محتفظة بأكثر من نسخة من عناصر قاعدة البيانات .

- بشكل عام يطلق على النسخة الأصلية من عنصر البيانات اسم (before image(BFIM)) ، ويطلق على النسخة المُحدثّة اسم (after image (AFIM))

Transaction rollback تراجع المعاملات

(b)

		A	B	C	D
		30	15	40	20
	[start_transaction, T_3]				
	[read_item, T_3, C]				
*	[write_item, T_3, B, 15, 12]		12		
	[start_transaction, T_2]				
	[read_item, T_2, B]				
**	[write_item, T_2, B, 12, 18]		18		
	[start_transaction, T_1]				
	[read_item, T_1, A]				
	[read_item, T_1, D]				
	[write_item, T_1, D, 20, 25]				25
	[read_item, T_2, D]				
**	[write_item, T_2, D, 25, 26]				26
	[read_item, T_3, A]				

* T_3 is rolled back because it did not reach its commit point.

** T_2 is rolled back because it reads the value of item B written by T_3.

تراجع المعاملات Transaction rollback

- كما سبق وان ذكرنا فان اي عنصر بيانات تم تغييره (الكتابة عليه) من قبل معاملة فاشلة فانه يحدث له تراجع الى حالته الأولى BFIM .
- يجب تجنب حدوث عملية التراجع المتتالي cascading rollback ، حيث أنها تستهلك الكثير من الوقت .
- يتم تخزين فقط عمليات write_item في سجل المعاملات لانها مهمة عند اجراء التراجع Rollback كما سبق وان راينا في المثال السابق .
- يمكن تجنب حدوث التراجع المتتالي cascading rollback باستخدام طرق استعادة تعتمد علي guarantee cascade less or strict schedules

THE END