

توالی پذیری تراکنش : تراکنشها به دو صورت اجرائی شوند یا متوالی ، یکی پس از دیگری یا همروند (موازی)

← مجموعه تراکنشها $\{T_1, T_2, \dots, T_n\}$ به دو صورت قبل اجراست

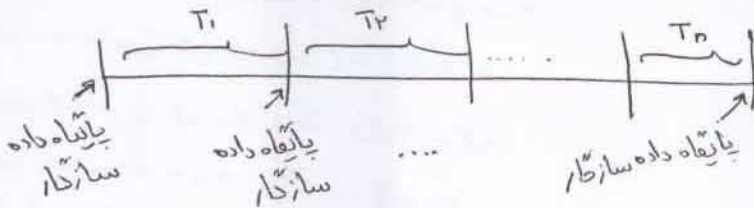
۱- متوالی یا پی در پی ← یکی پس از دیگری

۲- موازی یا همروند ← باهم

معایب و محاسن

① متوالی ← یکی پس از دیگری

اول تراکنش ۱ اجرا می شه (T_1) بعد تراکنش ۲ اجرا می شه (T_2) و در آخر (T_n) چون هر تراکنش یائفاه داده را از حالت سازگار به حالت سازگار می پره اخص متوالی



اشغال این روش ← زمانبر بودن است .

بهتر است ما تراکنشها را به صورت موازی اجرا کنیم منتها اجرای همروند تراکنشها یکسری مشکلات داره که بهش می گیم مشکلات تداخل کنترل شده

② موازی ← یعنی تراکنشها بعضی از اعمال خود را همزمان انجام دهند که از نظر زمانی صرفه جویی

نشه و حق کسی هم ضایع نشود.

مشکل زمان در روش موازی (همروند) حل می شود. حسین موازی

مشکل: ایجاد مسائل تداخل های کنترل نشده

مشکلات - تداخل کنترل نشده:

① نتیجه از دست رفته

② خواندن داده نادرست

③ تحلیل ناسازگار

مشکل نتیجه از دست رفته \neq

مثال: فرض کنید ۲ تراکنش داریم که می خوان همزمان اجرا شوند.

t	T ₁	T ₂
t ₁	R(D ₁) D ₁ = D ₁ + K 1000 + 500	
t ₂		R(D ₁) D ₁ = D ₁ + N 1000 + 700
t ₃	w(D ₁) 1500	
t ₄		w(D ₁) 1700

- t ← زمان
- T₁ ← تراکنش ۱
- T₂ ← تراکنش ۲
- Read ← R
- write ← w
- Data ← D

در زمان t₁ تراکنش T₁ داده D₁ را می خواند بعد D₁ را یک محاسبای روش انجام می ده مثلاً + K می کند و قبل از اینکه این کار رو

شروع کند در زمان t₂ تراکنش T₂ می آید و D₁ رو می خون و به D₁ یک چیز دیگه اضافه می کنه و هنوز write نکرده و اینها فقط در متغیر ها عملی هست و در زمان t₃ تراکنش T₁ write می کنه و در زمان t₄ تراکنش T₂ write می کنه

فرض کنید مقدار D₁ ابتدا برابر ۱۰۰۰ بود و مقدار K = ۵۰۰ ولی هنوز write نکرده و در متغیر محلی

②

هست. در تراکنش T_2 وقتی D_1 خوانده می شود همان ۱۰۰۰ روی صفحه خون چون هنوز تغییر نکرده، مقدار $N = ۷۰۰$ تراکنش T_1 ← اومد و write کرد ۱۵۰۰

$$w(D_1) \Rightarrow ۱۵۰۰$$

بلافاصله تراکنش T_2 می آید و write می کند اینبار ۱۷۰۰ - write می کند

$$w(D_2) \Rightarrow ۱۷۰۰$$

در کل نتیجه می شود ۱۷۰۰ و ۵۰۰ ای که در تراکنش T_1 اضافه شده بود، نادیده گرفته می شود.

نتیجه از دست رفته ← ۵۰۰

وقتی T_2 پیش می آید که تراکنش بلافاصله بعد از اینکه تراکنش دیگری مقداری برای داده ای نوشت، مقدار جدیدی برای آن داده بنویسد.

مشکل خواندن داده ناجور ←

تراکنش T_1 داده ای را در مکان k می خواند و D_1 را با k جمع می کند و بعد write می کند.

سپس در زمان T_2 تراکنش T_2

داده D_1 را می خواند و یکسری

عملیات روش انجام می ده

در زمان T_2 تراکنش T_1

می خواست کاری انجام دهد اما

طرز شد یعنی موفق نبود پس

باید داده های T_1 رو که انجام داده باید عملیاتی که انجام داده

خوشی داشته برگرد به حالت اولیه

t	T_1	T_2
t_1	$R(D_1)$ $D_1 = D_1 + K$ $w(D_1)$	
t_2		$R(D_1)$ $D_1 = D_1 + M$ $w(D_1)$
t_3

مقدار $D_1 = ۱۰۰۰$ و $K = ۵۰۰$ و چاپ کرد ۱۵۰۰ و فرض کنید T_2 اومد خون $D_1 = ۱۵۰۰$ و $M = ۲۰۰$

و چاپ کرد ۱۷۰۰ ولی در آخر باید برگرد به حالت اولیه یعنی ۵۰۰ تا ارزش کم شود چون تراکنش

T_1 هنوز از ثبت نرسیده این رو می گن داده نامچور.

مشکل خواندن

این مشکل زمانی پیش می آید که تراکنشی مثل T_2 داده های موقت به هک نام شده ی تراکنش دیگری مثل T_1 را بخواند. حال در صورت طرد تراکنش T_1 نتیجه تراکنش T_2 غلط است.

مشکل تحلیل ناسازگار ←

تراکنش T_2 در زمان t_1 یک مجموعی رو می خواهد حساب کند و داده ی D_1 رو خونده و با sum جمع کرده ، در زمان t_2 تراکنش T_1 مقدار D_2 رو خونده و D_2 رو با k جمع کرده و write می کنه در زمان t_3 تراکنش T_2 اومده مقدار D_2 رو خونده و مقدار D_2 رو با sum جمع کرده در زمان t_4 تراکنش T_2 اومده D_2 رو هم خونده . در زمان t_5 تراکنش T_1 اومده D_2 رو خونده و D_2 رو با R جمع کرده و D_2 رو write کرده

t	T_1	T_2
t_1		Sum = 0 $R(D_1)$ sum = sum + D_1
t_2	$R(D_2)$ $D_2 = D_2 + k$ $w(D_2)$	
t_3		$R(D_2)$ sum = sum + D_2
t_4		$R(D_2)$ sum = sum + D_2
t_5	$R(D_2)$ $D_2 = D_2 + R$ $w(D_2)$	

مثال یک بانک آخر شب بی فواد جمع پولهای تو حساب ها رو حساب کنه ، مثلاً شده ۱۰ بعد از حسابی این حالا یک نفر اومد از حسابش برداشت کرد یا به مقداری به حسابش واريز کرد اون مقدار جدید که واريز يا برداشت شده در این جمع حساب نشده یعنی جمع ما با داده ما ناسازگار است.

مشکل تحلیل ناسازگار وقتی بروز می کند که تراکشی مثلاً T_1 دو داده D_1 و D_2 را به هنگام سازی کند. تغییرات بعد و تراکشی دیگری مثل T_2 یکی از این داده ها را قبل از به هنگام سازی و دیگری را بعد از به هنگام سازی بخواوند و استفاده کند، نتیجه غلط خواهد شد.

نتیجه. همروند اجرا شدن تراکشی خوب است از نظر زمانی ولی باید کنترل شود که این تداخلها و مشکلاتی که ذکر شد ایجاد نشود.

توالی پذیری \Leftarrow اگر تعدادی تراکشی داشته باشیم که به صورت همروند اجرا شوند ولی نتیجه کار مساوی حالتی است که این تراکشی ها به صورت متوالی انجام می شوند. در این صورت گوئیم این تراکشی ها توالی پذیرند.

انواع مختلف توالی پذیری \Leftarrow

تعریف: فرض کنید n تراکشی T_1, T_2, \dots, T_n داریم، یک طرح اجرای (S) از این n تراکشی دنبالای از اعمال است که

- ۱- عمل هر تراکشی دقیقاً یکبار در S باشد.
- ۲- ترتیب عمل ها در S همان ترتیب نسبی آنها در تراکشی باشد.

$$T_1 = \{ R(D_1), D_1 := D_1 + k, w(D_1), R(D_2), D_2 := D_2 + M, w(D_2) \}$$

$$T_2 = \{ R(D_1), R(D_2), D_1 := D_1 + D_2, w(D_1) \}$$

طرحهای اجرای مختلفی می توان داشت. باشیم.

این یک طرح اجرا \Leftarrow

S_1	t	T_1	T_2
	t_1	$R(D_1)$ $D_1 := D_1 + k$ $w(D_1)$	
		$R(D_2)$ $D_2 := D_2 + M$ $w(D_2)$	
	t_2		$R(D_1)$ $R(D_2)$ $D_1 := D_1 + D_2$ $w(D_1)$

ϵ	τ_1	τ_2
z_1	$R(D_1)$ $D_1 = D_1 + k$ $w(D_1)$	
z_2		$R(D_1)$ $R(D_2)$ $D_1 = D_1 + D_2$ $w(D_1)$
z_3	$R(D_2)$ $D_2 = D_2 + M$ $w(D_2)$	

یک طرح اجرایی دید ←

S_1 ← متوالی

S_2 ← اگر تصمیم S_1 با S_2 برابر شد می گن این طرح اجرا متوالی پذیر هست .
 طرح S_2 همروند هست .

نکته ← طرح اجرایی n تراکشن می تواند همروند باشد یا متوالی باشد، اگر متوالی باشد کار اینست ولی پایباده داده را از حالت سازگار به حالت سازگار می برد .
 طرح اجرایی همروند کار است ولی ممکن است پایباده داده سازگار نشود .

← چه طرح اجرایی مناسب است ؟

تعریف ← دو طرح اجرایی S_1 و S_2 از n تراکشن به 3 شکل می تواند معادل باشند

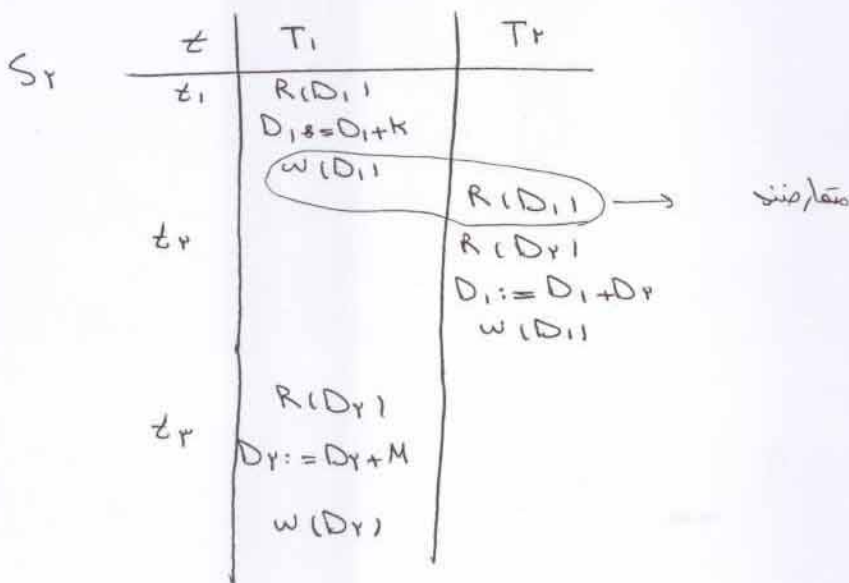
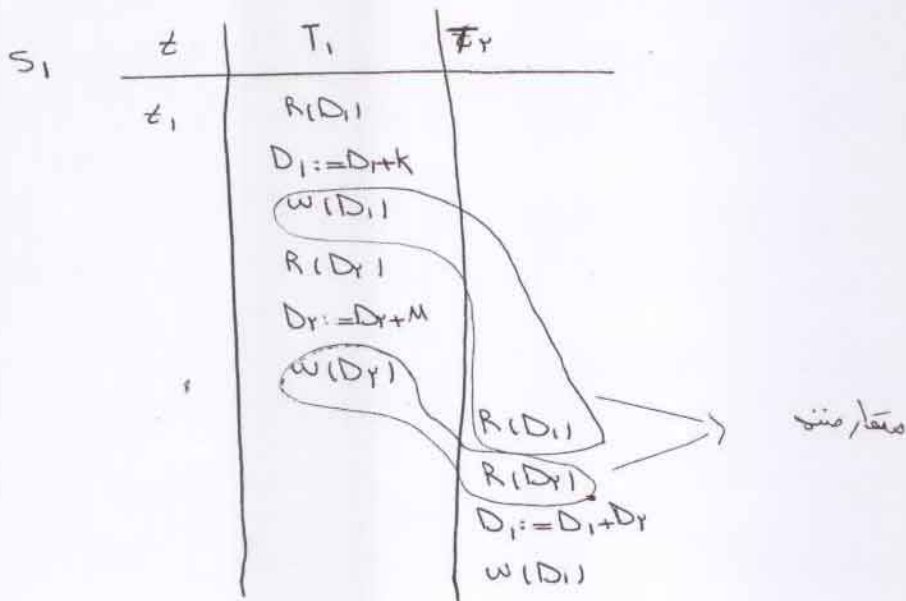
۱- معادل نتیجه ای

۲- معادل تقاضای

۳- معادل منابع

تعریف ← دو طرح اجرایی S_1 و S_2 از n تراکشن معادل نتیجه ای هستند اگر نتیجه ای اجرای این دو طرح روی یک مجموعه داده ای یکسان شود .
 معادل نتیجه ای زیاد جالب نیست .

تعریف ۴: دستورات I_1 و I_2 از نوع دستورات خواندن و نوشتن یا هم متعارضند اگر هم دو دستوری یک داده عمل کند. (۲) دو دستور در ترانژیکشن های مختلف باشند. (۳) حداقل یکی از دستورات دستور write «نوشتن» باشد. (۴) دستور متعارض دیگری بین آنها نباشد.



تعریف: دو طرح اجرا معادل تقاضای هستند اگر ترکیب دستورات متقاضی در دو طرح یکسان باشند.

جلسه بعدی: تعریف دو طرح معادل تقاضای

سوال: متقاضی یعنی یک write در T_1 و یک Read در T_2 و هر یکی حتماً باید write شود یعنی ممکن است T_1 و T_2 هر دو write هم بشوند. یا یک write و یک read روی یک داده

