MYSQL order by排序与索引关系总结

这里主要讨论一下InnoDB B-Tree索引的使用,不提设计,只管使用。B-Tree索引主要作用于WHERE和ORDER BY子句。这里讨论的均在MySQL-Server-5.1.42测试

CREATE TABLE `friends` (`ID` int(10) UNSIGNED NOT NULL AUTO_INCREMENT, `uid`bigint(20) UNSIGNED NOT NULL DEFAULT '0', `fuid` bigint(20) UNSIGNED NOT NULL DEFAULT'0', `fname` varchar(50) NOT NULL DEFAULT '', `fpicture` varchar(150) NOT NULL DEFAULT'', `fsex` tinyint(1) NOT NULL DEFAULT '0', `status` tinyint(1) NOT NULL DEFAULT '0', PRIMARY KEY (`ID`))
ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8; ALTER TABLE`friends` ADD INDEX uid_fuid (uid, fuid);

1.如果索引了多列,要遵守最左前缀法则。所谓最左前列,指的是查询从索引的最左前列开始,并且不跳过索引中的列。

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra	1
1	SIMPLE	friends	ref	u1d_fu1d	u1d_fu1d	1 8	const	251	1	
	in set (0.01 : explain sele		d, fname	from friends who	ere fuid =	187261112	9; 2	(ui	id, fuid)
id	select_type	table	type	possible_keys	key k	ey_len r	ef rows	i	Extra	
1	SIMPLE	friends	(ALL)	NULL	NULL N	ULL N	ULL 4117	318	using who	ere
	+	ct uid, fui	d,fname	from friends who	ere uid = :	1936840739 key_len	-+ 0	1872		Ext
1.54	serecc_cype	friends	ref	uid_fuid	uid_fuid	-+	const.co		1	

第2条语句,从索引的第二列开始查找,使用索引失败,导致MySQL采用ALL访问策略,即全表查询.在开发中,应该尽量避免全表查询。

2.当MySQL一旦估计检查的行数可能会"太多",范围查找优化将不会被使用。

id	select_type	table	type	possible_keys	l key	key_le	n r	ef	rows	Extra
1	SIMPLE	friends	range	u1d_fu1d	uid_fuic	1 8	1 1	WLL	251	Using when
	in set (0.01 selection)	1000000	d, fname	from friends who	ere uid > 2	1676103;	2	(uid	l, fuic	d)
sql	explain selec	ct uid,fui	+	from friends who				(uid	l, fuic	
sql		ct uid,fui	+	from friends who		1676103; ry_len r		(uid	-	d)

第2条语句使用了全表查询,它与第1条语句唯一的区别在于需要检查的行数远远多于第1条语句。在应用中,可能不会碰到这么大的查询,但是应该避免这样的查询出现: select uid from users where registered < 1295001384

3.索引列不应该作为表达式的一部分,即也不能在索引列上使用函数

id	select_type	table	type	possible_keys	key	l key.	len	ref	rows	Extra
1	SIMPLE	friends	ref	u1d_fu1d	u1d_f	u1d 8	i	const	251	i
row	in set (0.00	sec)	,	,						,
ysql	> explain sele	ct uid, fui	d, fname	from friends who	ere abs	(u1d) = 19	368407	739; 2	(ui	id, fuid)
1d	select_type	table	type	possible_keys	key	key_len	ref	rows	- 1	Extra
1	SIMPLE	friends	ALL	NULL	NULL	NULL	NULL	4117	818	Using where
LOM	in set (0.01	sec)				+	+			
ysq1:	> explain sele	ct uid,fui	d, fname	from friends who	ere uid	/1 = 19368	340739;	3	(uid	d, fuid)
id	select_type	table	type	possible_keys	key	key_len	ref	rows		Extra
	SIMPLE	friends	(ALL)	NULL	NULL	NULL	NULL	4117	040	Using where

第2和3条语句都有使用表达式,索引派不上用场。

4.尽量借用覆盖索引,减少select * from ...语句使用

id select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1 SIMPLE	friends	ref	uid_fuid	uid_fuid	8	const	251	Using index
i row in set (0.00 mysql> explain sele		d,(fsex)	from friends whe	re uid = 193	6840739;	2		7
id select_type	+	+	poss1ble_keys	+	key_len	ref	rows	Extra
1 SIMPLE	friends	ref	uid_fuid	uid_fuid	8	const	251	11
1 row in set (0.00 mysql> explain seld t d select_type	ct wid, fui	+	om friends where	+	340739; key_len	ref 3	rows	Extra
1 SIMPLE	friends	ref	uld_fuld	uid_fuid	8	const	251	+

第1句Extra中使用了Using index表示使用了覆盖索引。第3句也使用了覆盖索引,虽然ID不在索引uid_fuid索引列中,但是InnoDB二次索引(second index)叶子页的值就是PK值,不同于MyISAM。Extra部分的Using index表示应用了索引,不要跟type中的index混淆。第2句没有使用覆盖索引,因为fsex不在索引中。

5.ORDER BY子句,尽量使用Index方式排序,避免使用FileSort方式排序 MySQL支持二种方式的排序,FileSort和Index,后者效率高,它指MySQL 扫描索引本身完成排序。FileSort方式效率较低。ORDER BY满足以下情况,会使用Index方式排序:

a)ORDER BY 语句使用索引最左前列。参见第1句

- b)使用Where子句与Order BY子句条件列组合满足索引最左前列。参见第2句.
- 以下情况,会使用FileSort方式的查询
- a)检查的行数过多,且没有使用覆盖索引。第3句,虽然跟第2句一样,order by使用了索引最左前列uid,但依然使用了filesort方式排序,因为status并不在索引中,所以没办法只扫描索引。
- b)使用了不同的索引,MySQL每回只采用一个索引.第4句,order by出现二个索引,分别是uid_fuid和聚集索引(pk)
- c)对索引列同时使用了ASC和DESC。通过where语句将order by中索引列转为常量,则除外。第5句,和第6句在order by子句中,都出现了ASC和DESC排序,但是第5句却使用了filesort方式排序,是因为第6句where uid取出排序需要的数据,MySQL将其转为常量,它的ref列为const。
- d)where语句与order by语句,使用了不同的索引。参见第7句。
- e)where语句或者ORDER BY语句中索引列使用了表达式,包括函数表达式。参见第8,9句
- f)where 语句与ORDER BY语句组合满足最左前缀,但where语句中使用了条件查询。查见第10句,虽然where与order by构成了索引最左有缀的条件,但是where子句中使用的是条件查询。
- g)order by子句中加入了非索引列,且非索引列不在where子句中。
- h)order by或者它与where组合没有满足索引最左前列。参见第11句和12句,where与order by组合,不满足索引最左前列. (uid, fsex)跳过了fuid i)当使用left join,使用右边的表字段排序。参见第13句,尽管user.uid是pk,依然会使用filesort排序。

1 row 1	select_type		1		*******		h-						1		
row 1		+	+		+	ys			key_	+		raws	Extra		
		friend	s 1no	iex	NULL		u1d_t	fuld_fsex	1 17		NULL	416960	0 Using	1ndex	
<pre>/sql></pre>	In set (0.00								200						
+-			+		s where uid =					+			2		
1d	select_type	table	Typ	96	possible_key	5	key		key_le	in r	ef	rows			
1	SIMPLE	friend	s ref	1	u1d_fu1d_fse	×	u1d_fi	id_fsex	8	10	const	1	using w	mere; Us	ing index [
row f	n set (0.00	sec)													
					friends orde								3	+	
fd	select_type	table	typ	se	possible_key	5	key	key_len	ref	rov	rs	Extra	_	i	
1	SIMPLE	friend	S ALL	- 1				NULL	NULL	416	8600	us1ng	filesort	į	
row f	In set (0.00	sec)												600	
<pre>/sql></pre>	explain sele	ct uld f	rom fri	end	s order by uf	d, I	0;						4		
1d	select_type	table	typ	e e	possible_ke	ys I	key		key_1	len	ref	rows	Extr		
1	SIMPLE	friend	s 1nd	iex	NULL	1	u1d_t	fuld_fsex	1 17	1	NULL	416860	0 Using	index;	using/files
row 1	n set (0.00	sec)	+		+	+			+	+-					
					s order by ut	d AS	c, fut	d DESC;					_		
	select_type				possible_ke	ys i	key		key_1	en I	ref	rows	5 Extra		
1	SIMPLE	friend	s I find	iex	NULL		u1d_f	fu1d_fsex	1 17		NULL	416860	o usin	index;	using files
+-	In set (0.00	+	+		+	+			+	+					
			tatus f	rom	friends when	e uf	d = 68	387 order	by utd	ASC.	fuld t	MESC:	0		
												rows	Extra		
11	SIMPLE	friend	s ref	-	possible_key uid_fuid_fse	× 1	uid fi	dd fsex	8		onst		using w	ere I	
	In set (0.00	+	+												
				e con	friends when		d - 60	197 order	but You				77		
+-		+	+	+	possible_key								u		
		+							key_le	+		rows			
1	SIMPLE	friend	s rer		uid_fuid_fse	×	und_ti	Ind_rsex		1.9	const	1	using w	sere; us	ing filesor
		sec)													
		ect uid,			n friends whe			1 = 6887				Q			
ysql>		ect uld,			possible_ke	ys I	key	1 = 6887	n ref	17	OWS	8	·a		
/sq1>		ect uid,		pe	possible_ke	ys	key	1 = 6887	n ref	17	OWS	8 Extr	a ng where;	ustng/f	ilesort i
/sql>	select_type	ect uid,s table friend	İty	pe	possible_ke	ys	key	1 = 6887	n ref	17	OWS	8 extr	a ng where;	Using/f	flesort'i
/sql>	select_type SIMPLE in set (0.00	table friend	i ty	/pe	possible_ke	ys	key	1 = 6887 key_le	n ref	r	ows 168600	8 xtr	a ng where;	Using ^r ,f	ilesort'
/sql> 1d 1 row /sql>	select_type SIMPLE in set (0.00	table friend sec)	ty ds AL	pe L from	possible_ke	ys i	key NULL	1 = 6887 key_le NULL 6887 orde	n ref	L 4	ows 168600	8 xtr		using/f	ilesorti
/sql> 1d 1 row /sql> 1d	select_type SIMPLE in set (0.00 explain sele	ect uid,s table friend sec) ect uid,s table	ty ds AL	pe L from	possible_ke	ys I	key NULL i1d = key	1 = 6887 key_le NULL 6887 orde	n ref NUL er by ab key_	L 4	ows 168600	9	Extra		ilesort
/sql> 1d 1 row /sql> 1d 1	select_type SIMPLE in set (0.00 explain sele select_type	ect uid,: table frienc sec) ect uid,: table frienc	ty ds AL	pe L from	possible_ke NULL n friends whe possible_ke	ys I	key NULL i1d = key	1 = 6887 key_le NULL 6887 orde	n ref NUL er by ab key_	L 4	0ws 168600); ref	9	Extra		
/sql> 1d 1 row /sql> 1d 1 row	select_type SIMPLE in set (0.00 explain sele select_type SIMPLE in set (0.00 explain sele	ect uid,: table frienc sec) ct uid,: table frienc sec) sct uid,:	ty ds AL	from	possible_ke NULL In friends whe possible_ke uid_fuid_fs	ys re u ys ex	key NULL id = key uld_ id >	1 = 6887 key_1e NULL 6887 orde fu1d_fsex	n ref	s(u1d	ows 168600); ref	9	Extra		
/sql> 1d 1 row /sql> 1d 1 row	select_type SIMPLE in set (0.00 explain sele select_type SIMPLE in set (0.00 explain sele	table frience sec) sct uid,s table frience sec) sct uid,s	ty ds AL	from	possible_ke NULL friends whe possible_ke uid_fuid_fs	re u	key NULL id = key uld_ id >	1 = 6887 key_le NULL 6887 orde fuld_fsex	NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL	s(u1d	ows 168600); ref const	9	Extra Using	where;	
/sql> 1 row /sql> 1 row /sql> 1 row /sql>	select_type SIMPLE in set (0.00 explain sele select_type SIMPLE in set (0.00 explain sele select_type	table frience sec) tot uid,s table frience sec) tot uid,s table trience sec)	tyds AL	from	possible_ke NULL friends whe possible_ke uid_fuid_fs friends whe possible_k	ys i	key NULL i1d = key u1d i1d >	1 = 6887 key_le NULL 6887 orde fuld_fsex	n ref	l r L 4 les (u1d	ows 168600 ref const	9 0w	Extra Using	where:	using/files
/sql> 1 1 1 1 1 1 1 1 1 1	select_type SIMPLE in set (0.00 explain sele select_type SIMPLE in set (0.00 explain sele select_type	table friencesec) sect uid,s table friencesec) sect uid,s table friencesec) sect uid,s	tyds AL	from	possible_ke NULL friends whe possible_ke uid_fuid_fs	ys i	key NULL i1d = key u1d i1d >	1 = 6887 key_le NULL 6887 orde fuld_fsex	n ref	l r L 4 les (u1d	ows 168600 ref const	9 0w	Extra Using	where:	using/files
/sql> 1d 1 row /sql> 1d 1 row /sql> 1d 1 row /sql>	select_type SIMPLE in set (0.00 explain sele select_type SIMPLE in set (0.00 explain sele select_type SIMPLE in set (0.00	table friend sec) ct uid,: table friend sec) ct uid,: table friend sec) ct uid,:	ty ty status ty status ty status ty status ty status ty status	from pe from pe from pe	possible_ke NULL friends whe possible_ke uid_fuid_fs friends whe possible_k uid_fuid_f	ys 	key NULL Id = key u1d Id > key u1d	1 = 6887 key_le NULL 6887 orde fuld_fsex fuld_fsex	en ref	s (u1d	ows 168600); ref const u1d; ref	9 0w	Extra Using	where:	using/files
	select_type SIMPLE in set (0.00 explain sele	table friend sect uid,: table friend sect uid,: table friend sect uid,: table friend sect uid,:	ty t	pe L from pe from pe inge	possible_ke NULL friends whe possible_ke uid_fuid_fs friends whe possible_k uid_fuid_f	re u re u eys sex	key NULL id = key uld id > key uld id >	1 = 6887 key_le NULL 6887 orde fuld_fsex 193684073	r by ab key l 8 s order key x 8	by f	ows 168600); ref const uld; ref	9 0w	Extra Using	where:	using/files
/sql> 1d 1 row /sql> 1d 1 row /sql> 1d 1 row /sql> 1d 1 row	select_type SIMPLE In set (0.00 explain sele select_type SIMPLE In set (0.00 explain sele select_type SIMPLE In set (0.00 explain sele select_type select_type select_type select_type	table friencesec) table friencesec) table friencesec) table friencesec) table friencesec) table	ty t	pe L from pe from pe from pe	possible_ke NULL n friends whe possible_ke uid_fuid_fs n friends whe possible_k uid_fuid_f	ys ex ex eys sex	key NULL III = key uId > I key uid = key	1 = 6887 key_1e NULL 6887 orde fu1d_fsex 193684073 fu1d_fsex fu1d_fsex	r by ab key_ 8 8 order key key_	by f	ows 168600 ref const uld; ref NULL sex; ref	9 10 1 row	Extra	where;	using files
/sql> 1d 1 row /sql> 1d 1 row /sql> 1d 1 row /sql> 1d 1	select_type SIMPLE in set (0.00 explain sele	table friencesec) table friencesec) table friencesec) table friencesec) table friencesec) table	ty t	pe L from pe from pe from pe	possible_ke NULL friends whe possible_ke uid_fuid_fs friends whe possible_ke uid_fuid_f	re u eys isex	key NULL id = key uld id > key uld id > key uld id = key uld	1 = 6887 key_1e NULL 6887 orde fu1d_fsex 193684073 fu1d_fsex fu1d_fsex	r by ab key key key key key key	by f	ows 168600); ref const u1d; ref NULI	9 10 10 1 25	Extra	where;	using/files
/sql> 1d 1 row /sql> 1d 1 row	select_type SIMPLE in set (0.00 explain sele select_type SIMPLE in set (0.00 explain sele select_type SIMPLE in set (0.00 explain sele select_type SIMPLE in set (0.00	table friend sec) ct uid,:	ty t	from pe in	possible_ke NULL In friends whe possible_ke uid_fuid_fs In friends whe possible_ke uid_fuid_f	ys ys ys ex ex	key NULL id = key uld id > key i uld key uld uld uld =	1 = 6887 key_le NULL 6887 orde fuld_fsex fuld_fsex fuld_fsex fuld_fsex	r by ab key key key key key key	by f	ows 168600); ref const u1d; ref NULI	9 10 10 1 25	Extra	where;	using files
/sql> 1d 1 row /sql> 1d 1 row	select_type SIMPLE in set (0.00 explain sele select_type SIMPLE in set (0.00 explain sele select_type SIMPLE in set (0.00 explain sele select_type SIMPLE in set (0.00	table friend sec) ct uid,:	ty t	from pe in	possible_ke NULL n friends whe possible_ke uid_fuid_fs n friends whe possible_k uid_fuid_f	ys ys ys ex ex	key NULL id = key uld id > key i uld key uld uld uld =	1 = 6887 key_le NULL 6887 orde fuld_fsex fuld_fsex fuld_fsex fuld_fsex	r by ab key key key key key key	by f	ows 168600); ref const u1d; ref NULI	9 10 10 1 25	Extra	where;	using files
/sql> 1d 1 row /sql> 1d row /sql> 1d row /sql> 1d row /sql> 1d row /sql>	select_type SIMPLE in set (0.00 explain sele select_type SIMPLE in set (0.00 explain sele select_type SIMPLE in set (0.00 explain sele select_type SIMPLE in set (0.00	table friend friend friend friend friend friend sec) table friend sec) table friend sec) ct uid,:	ty t	from the fro	possible_ke NULL In friends whe possible_ke uid_fuid_fs In friends whe possible_ke uid_fuid_f	re u yys ex eys sex fe u d,fs	key NULL Id = key u1d Id > key u1d Id = key u1d key u1d key uid key key	1 = 6887 key_le NULL 6887 orde fu1d_fsex fu1d_fsex fu1d_fsex fu1d_fsex	r by ab key_ key_ key_ key_	by f	ows 168600 ref const u1d; ref NULL sex; ref const	9 10 10 1 25	Extra	where; where; where;	using files
/sql> 1d 1 row /sql> 1d row /sql> 1d row /sql> 1d row /sql> 1d row /sql>	select_type SIMPLE In set (0.00 explain sele select_type simple select_type select_type select_type select_type select_type select_type	table friend friend friend friend friend friend sec) table friend sec) table friend sec) ct uid,:	ty t	from the period of the period	possible_ke NULL friends whe possible_ke uid_fuid_fs friends whe possible_k uid_fuid_f friends whe possible_ke uid_fuid_fs s order by uid possible_ke NULL	re u yys ex eys sex fe u d,fs	key NULL Id = key u1d Id > key u1d Id = key u1d key u1d key uid key key	1 = 6887 key_le NULL 6887 orde fuld_fsex fuld_fsex fuld_fsex fuld_fsex	r by ab key_ key_ key_ key_	by f	ows 168600 ref const u1d; ref NULL sex; ref const	9 ox 10 rows	Extra Using Extra Using	where; where; where;	using files
	select_type SIMPLE in set (0.00 explain sele select_type	table friend	tyds AL status tyds re status tyds ra status tyds ra status tyds ra status	from the fro	possible_ke NULL friends whe possible_ke uid_fuid_fs friends whe possible_k uid_fuid_f friends whe possible_ke uid_fuid_fs s order by uid possible_ke NULL	ore universe	key NULL Id = key u1d Id > key u1d key und key und key und	1 = 6887 key_le NULL NULL 6887 orde fuld_fsex 193684073 fuld_fsex fuld_fsex	r by ab key_ key_ B	by f	ows 168600 ref const uld; ref NULL	9 rows 10 rows 12 rows 41686	Extra Using Extra Using Extra Using	where; where; where; a g index;	using files using files using files
	select_type SIMPLE in set (0.00 explain sele select_type	table friend	ty ds AL status ty ds re status ty ds tento ty ds te	from the fro	possible_ke NULL In friends whe possible_ke uid_fuid_fs In friends whe possible_ke uid_fuid_fs In friends whe possible_ke uid_fuid_fs s order by uid possible_ke NULL con friends as	ore universe	key NULL Id = key u1d Id > key u1d key u1d key u1d lest; key u1d	1 = 6887 key_le NULL 6887 orde fuld_fsex 193684073 fuld_fsex fuld_fsex fuld_fsex fuld_fsex	r by ab key_ key_ B	by f	ows 168600 ref const uld; ref NULL ref NULL	9 rows 12 rows 41686	Extra Using Extra Using Extra Using	where; where; where; a g index;	using files using files using files
ysql> 1d 1 1 1 1 1 1 1 1 1 1	select_type SIMPLE in set (0.00 explain sele select_type	table friend	ty status	from the fro	possible_ke NULL In friends whe possible_ke uid_fuid_fs I possible_k uid_fuid_f I friends whe possible_ke uid_fuid_fs s order by uid possible_ke NULL	ore upys ex eys sex d,fs	key NULL Id = key u1d > id > key i u1d > key u1d = key	1 = 6887 key_le NULL 6887 orde fuld_fsex 193684073 fuld_fsex fuld_fsex fuld_fsex key_le	r by ab key_ key_ B	by f	ows 168600 ref const uld; ref NULL ref const ref const ref	9 rows 10 rows 112 rows 41686	Extra Using Extra Using Extra Using Extra Using	where; where; where; i = 1936i	using files using files using files

6.慎用left join语句,避免创建临时表 使用left join语句的时候,避免出现创 建临时表。尽量不要用left join,分而治之。非要使用的时候,要询问自 己是不是真要必须要使用。

1d	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE SIMPLE	f u		uid_fuid_fsex PRIMARY	uid_fuid_fsex PRIMARY	8	const	251	using temporary; using filesort

2 rows in set (0.00 sec)

7.高选择性索引列。 尽量使用高选择性的过引来过滤数据。高选择性指 Cardinality/#T越接近1,选择性越高,其中Cardinality指表中索引列不重 复值(行)的总数。PK和唯一索引,具有最高的选择性,即1。推荐可选性 达到20%以上。

```
sysql> explain select ID, wid, fuid from friends where wid = 7884 and fuid = 222294335 and ID = 5801383;

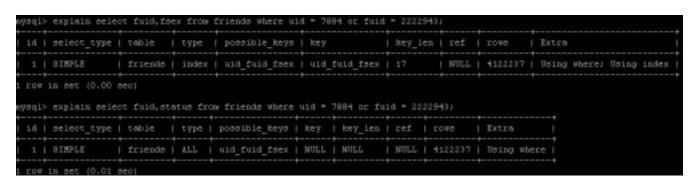
| id | select_type | table | type | possible_keys | key | key_len | ref | rows | Extra |

| 1 | SIMPLE | friends | const | PRIMARY, wid_fuid_fsex | PRIMARY | 4 | const | 1 | |

1 row in set (0.00 sec)
```

这里有二个索引可供使用,而MySQL选择PRIMARY,是因为它具有更高的选择性。

8.谨防where子句中的OR。where语句使用or,且没有使用覆盖索引,会进行全表扫描。应该尽量避免这样OR语句。尽量使用UNION代替OR



第1句虽然使用了索引,但是查行时间依然不可以恭维, mysql要检查的行很多,但是返回的行却很少.Extra中的using where表示需要通过where子句扔弃不需要的数据行。

9.LIMIT与覆盖索引 limit子句,使用覆盖索引时比没有使用覆盖索引会快很多

转自: http://my.oschina.net/longniao/blog/110384?fromerr=dKzaJdu1