Symbolic Deadlock Analysis for Concurrent Libraries and their Clients

Jyotirmoy V. Deshmukh¹ E. Allen Emerson¹ Sriram Sankaranarayanan² {deshmukh, emerson}@cs.utexas.edu, srirams@colorado.edu

¹University of Texas at Austin

²University of Colorado at Boulder

Automated Software Engineering 2009

Symbolic Deadlockability Analysis

A DEN A DEN A DEN A DEN ELLE

Deshmukh

Want your own MarkWail? Tell us about it. Messages per Month (Swipe to refine by date) Stor Actions 1 to 10 of about 126087 Biological Actions I to 10 of about 126087 <th>MARK</th> <th></th> <th></th> <th></th> <th>Search 7,324 lists for: deadlock</th>	MARK				Search 7,324 lists for: deadlock
Messages per Month (Swipe to refine by date) Sort by Relevance Actions 1 to 10 of about 126087 3500 300	🖶 Home		2		Want your own MarkMail? Tell us about it.
 Re: [PATCH 2/4] ext3: Fix possible deadlock between ext3 tru]ext3: Fix possible deadlock between ext3_truncate() and ext3 tru]ext3: Fix possible deadlock between ext3_truncate() and ext3 tru]ext3: Fix possible deadlock between ext3_truncate() and ext3 get_blocks() elow transaction start (and it can lead to a real deadlock with ext3_get_blocks() allocating new blocks fro (inode)); +/* + Drop truncate mutex to avoid deadlock with ext3_get_blocks() allocating new blocks fromA deadlock with ext3_get_blocks() allocating new blocks fromA and get_blocks() allocating new blocks fromA deadlock with ext3_get_blocks() allocating new blocks fromA deadlock muteteansA deadlock muteteans	Messages per Month		(Swipe to refine l	oy date)	Sort by Relevance Actions 1 to 10 of about 126087
3000 2500 2500 2500 2000 2500 2000 2500 1500 2500 1500 2500 1500 2500 1500 2500 1000 2500 1000 2500 1000 2500 1000 2500 1000 2500 1000 2500 1000 2500 1000 2500 1000 2500 1000 2500 1000 2500 1000 2500 1000 2500 1000 2500 1000 2500 1000 2500 1000 2500 1000 2500 10000 2500 10000 2500 10000 2500 10000 2500 10000 2500 10000 2500 10000 2500 10000 2500	3500				Re: [PATCH 2/4] ext3: Fix possible deadlock between
2500	3000				ext3 tru
2000 1500 1500 1500 10000 1500 100000 100000 1000	2500] ext3: Fix possible deadlock between ext3_truncate() and
1500 +/* + To producate mutex to avoid deadlock with 1000 +/* + To producate mutex to avoid deadlock with 's2	2000				deadlock with ext3_get_blocks() allocating new blocks fro (inode));
1000 ext3 get plocks nanide + At this month,, t can lead to a real 500 iss_get plocks nanide + At this month,, t can lead to a real 's2 's2 'y2 's2 What List? Who Sent It? View more Who Sent It? iss_get plocks nanide + At this month,, t can lead to a real Mart List? View more What List? Who Sent It? iss_get plocks nanide + At this month,, t can lead to a real metaeans broken_builds 18,521 Hudson Builds 8,972 org.netbeans.api-change 6,996 iss_get plocks 18 iss_get plocks 1, and the window only after the owning thread is Idwr-user] Java-level deadlock 1 Jdwr-user] Java-level deadlock 1 get plocks 1, and the window only after the owning thread is 1 iss_get plocks 1, and the window only after the owning thread is 1 iss_get plocks 1, and the window only after the owning thread is 1 iss_get plocks 1, and the window only after the owning thread is 1 iss_get plocks 1, and 1	1500		1.1		+ /* + * Drop truncate_mutex to avoid deadlock with
500 Aug 17, 2009 - Jan Kara - org.kernel.vger.linux-ext4 - Ø '92 '92 '92 '92 '92 '92 '93 What List? View more Who Sent It? u.melvger.linux-kernel 18,521 Hudson Builds 8,972 matbeans.broken_builds 10,177 builds 9,972 corg.netbeans.api-change 6.996 iss@own.netbeans.org com.mysql.lists.commits 3,040 uetbeans.openide.issues 2,171 FreeBsD bugmaster 1,109 staresel.ogenide-hackers 1,470 Andrew Morton 1,052	1000		a harded		deadlock with ext3 get blocks() allocating new blocks from
Image: Stress of the stres	500		and the ball of the second second		Aug 17, 2009 - Jan Kara - org.kernel.vger.linux-ext4 - 🖉 📀
What List? View more Whot Sent It? View more owning thread is metuger.linux-kernel 18,521 Hudson Builds 8,972 Mar 13, 2009 - Kirill K. Smirnov - org.winehq.wine-patches - Ø Mar 13, 2009 - Kirill K. Smirnov - org.winehq.wine-patches - Ø Image: Smirnov - org.winehq.wine-patches - Ø Mar 13, 2009 - Kirill K. Smirnov - org.winehq.wine-patches - Ø Image: Smi	'92			····	mciavi: fix deadlock + /* To avoid deadlock deal with the window only after the owning thread + /* To avoid deadlock deal with the window only after the
metuger.linux-kernel 18,521 Hudson Builds 8,972 netbeans.broken_builds 10,177 hud@deadlock.netbea 8,362 netbeans.broken_builds 6,976 iss@www.netbeans.org 4,599 netbeans.opichanges 6,996 iss@www.netbeans.org 4,599 netbeans.opichanges 2,171 FreeBSD bugmaster 1,109 netbesd.freebsd-current 1,893 iss@openide.netbean 1,109 netbesd.freebsd-current 1,470 Andrew Morton 1,052	what List? V	iew more	who sent it? V	iew more	owning thread is Mar 13, 2009, Kirill K. Smirney, are winede wine patches.
netbeans.broken_builds 10,17 hud@deadlock.netbea 8,362 jdwr-user1.java-level deadlock org.netbeans.api-changes 6,996 iss@www.netbeans.org 4,599 Java-level deadlock ====================================	rnel.vger.linux-kernel	18,521	Hudson Builds	8,972	Wai 15, 2005 - Kini K. Shintov - org. Wineriq. Wine-patches - 8
org.meubranis.api-changes 6,954 iss@vww.ineuceans.org 4,954 java-level deadlock: ====================================	netpeans.proken_pullds	10,177	nud@deadlock.netbea	8,362	[dwr-user] Java-level deadlock
Continuing and association of the second	com myeal lists commits	2,040	hugz @apacho.org	4,599	Java-level deadlock: ====================================
Interebsd/freebsd-current 1,893 iss…@openide.netbean 1,104 Storestions of the store	otheone onenide issues	2 1 7 1	FreeBSD humaster	1 1 1 9	deadlock. Heap PSYoungGen total 218752K, used 213 Java-level
ustaresal.pasal-hackers 1.470 Andrew Morton 1.052 Маг 25 2009 - Иван Трофимов - net java dev dwr.users - 🖉 🕒	reebsd freebsd-current	1.893	iss. @openide netbean	1.104	deadlock: ====================================
	stgresql.pgsql-hackers	1,470	Andrew Morton	1,052	оооо-ехес-20 Mar 25, 2009 - Иван Трофимов - net.java.dev.dwr.users - 🖉 💿

MARK				Search 7,324 lists for: deadlock
😢 Home		2		Want your own MarkMail? Tell us about it.
Messages per Month		(Swipe to refine	by date)	Sort by Relevance Actions 1 to 10 of about 126087
3500				Re: [PATCH 2/4] ext3: Fix possible deadlock between
3000		1		ext3 tru
2500] ext3: Fix possible deadlock between ext3_truncate() and
2000				deadlock with ext3_get_blocks() allocating new blocks fro (inode));
1500				+ /* + * Drop truncate_mutex to avoid deadlock with
1000				deadlock with ext3 get blocks() allocating new blocks from
500		and the second		Aug 17, 2009 - Jan Kara - org.kernel.vger.linux-ext4 - 🖉 📀
'92	الافصيافي		'09	mciavi: fix deadlock + /* To avoid deadlock deal with the window only after the owning
What List? Vi	iew more	Who Sent It?	View more	owning thread is
rnel.vger.linux-kernel	18,521	Hudson Builds	8,972	Mar 13, 2009 - Kirill K. Smirnov - org.winehq.wine-patches - 🖉 🛛 📀
netbeans.broken_builds	10,177	hud@deadlock.netbea	8,362	[dwr-user] Java-level deadlock
org.netbeans.api-changes	6,996	iss@www.netbeans.org	4,599	Java-level deadlock: ====================================
com.mysql.lists.commits	3,040	bugz@apache.org	1,587	8085-exec-20 at java.lang.Thread.run(Thread.java:619) Found 1
etbeans.openide.issues	2,171	FreeBSD bugmaster	1,109	deadlock: ====================================
reebsd.freebsd-current	1,893	iss@openide.netbean	1,104	8085-exec-20
stgresql.pgsql-hackers	1,470	Andrew Morton	1,052	Mar 25, 2009 - Иван Трофимов - net.java.dev.dwr.users - 🖉 📀

MAREMoil		4		Search 7,324 lists for: deadlock			
😢 Home		3		Want your own MarkMail? Tell us about it.			
Messages per Month		(Swipe to refine	by date)	Sort by Relevance Actions 1 to 10 of about 126087			
3500		1	De	adlocks increasingly important			
2500			In the] ext3: Fix possible deadlock between ext3_truncate() and ext3_get_blocks() elow transaction start (and it can lead to a real			
2000				deadlock with ext3_get_blocks() allocating new blocks fro (inode));			
1500		(M)		ext3_get_blocks_handle + * At this moment,t can lead to a real			
500	-	المتعاد الملي المريد المريد المريد		deadlock with ext3_get_blocks() allocating new blocks from Aug 17, 2009 - Jan Kara - org.kernel.vger.linux-ext4 - ⊘			
92				mciavi: fix deadlock + /* To avoid deadlock deal with the window only after the owning			
What List? View	w more	Who Sent It?	View more	owning thread is			
rnel.vger.linux-kernel	18,521	Hudson Builds	8,972	Mar 13, 2009 - Kirill K. Smirnov - org.winehq.wine-patches - 🖉 🛛 📀			
netbeans.broken_builds	10,177	hud@deadlock.netbea	. 8,362	[dwr-user] Java-level deadlock			
org.netbeans.api-changes	6,996	iss@www.netbeans.org	4,599	Java-level deadlock: ====================================			
com.mysql.lists.commits	3,040	bugz@apache.org	1,587	8085-exec-20 at java.lang.Thread.run(Thread.java:619) Found 1			
etbeans.openide.issues	2,171	FreeBSD bugmaster	1,109	deadlock: Heap PS roungGen total 218/52K, used 213 Java-level deadlock: ====================================			
reebsd.freebsd-current	1,893	iss@openide.netbean	1,104	8085-exec-20			
stgresql.pgsql-hackers	1,470	Andrew Morton	1,052	Mar 25, 2009 - Иван Трофимов - net.java.dev.dwr.users - 🖉 📀			

Want your own MarkMail? Tell us about it. Messages per Month (Swipe to refine by date) Sort by Relevance Actions 1 to 10 of about 126087 Decadlocks increasingly important <t< th=""><th>MAREMoil</th><th>/</th><th></th><th></th><th>Search 7,324 lists for: deadlock</th></t<>	MAREMoil	/			Search 7,324 lists for: deadlock
Messages per Month (Swipe to refine by date) Sort by Relevance Actions 1 to 10 of about 126087 3000 000 <t< td=""><td>😢 Home</td><td></td><td>2</td><td></td><td>Want your own MarkMail? Tell us about it.</td></t<>	😢 Home		2		Want your own MarkMail? Tell us about it.
3500 Deadlocks increasingly important 3500	Messages per Month		(Swipe to refine	e by date)	Sort by Relevance Actions 1 to 10 of about 126087
3000 ext3 tru 2500 pet3 Fix possible deadlock between ext3 trucate() and ext3 get blocks() allocating new blocks for (inode)); t* + * Drop trucate mutex to avoid deadlock with ext3 get blocks() allocating new blocks for (inode)); t* + * Drop trucate mutex to avoid deadlock with ext3 get blocks() allocating new blocks for (inode)); t* + * Drop trucate mutex to avoid deadlock with ext3 get blocks() allocating new blocks for (inode)); t* + * Drop trucate mutex to avoid deadlock with ext3 get blocks() allocating new blocks for (inode)); t* + * Drop trucate mutex to avoid deadlock with ext3 get blocks() allocating new blocks for (anole); t* + * Drop trucate mutex to avoid deadlock with ext3 get blocks() allocating new blocks for (anole); t* + * Drop trucate mutex to avoid deadlock with ext3 get blocks() allocating new blocks for (anole); t* - * To avoid deadlock deal with the window only after the owning thread i for the window only after the owning thread i for the window only after the owning thread i for the for the window only after the owning thread i for the window only after t	3500			De	adlocks increasingly important
2000 ex13 get blocks get blocks glockal mead to a real deallock with ex13 get blocks get blocks glockal mead to a real deallock with ex13 get blocks get blocks glockal mead to a real deallock with ex13 get blocks get blocks glockal mead to a real deallock with ex13 get blocks glockal mead to a real deallock deal with the window only after the owning thread to a real tex17. To avoid deallock deal with the window only after the owning thread to a real tex12. Set blocks glockal mead to a real tex17. To avoid deallock deal with the window only after the owning thread to a real tex12. Set blocks glockal mead to a real tex17. To avoid deallock deal with the window only after the owning thread to a real tex18. Set blocks glockal mead to a real tex17. To avoid deallock deal with the window only after the owning thread to a real t	2500		/	the stars	ext3_tru] ext3: Fix possible deadlock between ext3_truncate() and
1500 +/* + 2 Torp tuncate_mitex to avoid deadlock with 1500 +/* + 2 Torp tuncate_mitex to avoid deadlock with 1500 +/* + 2 Torp tuncate_mitex to avoid deadlock with 1500 +/* + 2 Torp tuncate_mitex to avoid deadlock with 1500 +/* + 2 Torp tuncate_mitex to avoid deadlock with 1500 +/* + 2 Torp tuncate_mitex to avoid deadlock with somentt can lead to a real deadlock with ext3 get_blocks() allocating new blocks from 1500 +/* To avoid deadlock deal with the window only after the owning thread +/* To avoid deadlock deal with the window only after the owning thread is 4. metbeans.broken_builds 10,177 nud@deadlock.netbea8,362 Idwr.user] Java-level deadlock Iss_@www.netbeans.org 4.599 Idwr_user_laya-level deadlock = ************************************	2000			LU-	ext3_get_blocks() elow transaction start (and it can lead to a real deadlock with ext3 get blocks() allocating new blocks fro (inode));
1000 Sob_grid production and the sector a relation of the sector a relatio	1500				+ /* + * Drop truncate_mutex to avoid deadlock with
Soo Aug 17, 2009 - Jan Kara - org. kernel.vger.inux-ext4 - P '92 '09 What List? View more What List? Who Sent It? View more Who Sent It? metbeans.broken_builds 10,177 nude.gedeadlock.netbea	1000	_			deadlock with ext3_get_blocks() allocating new blocks from
'y2 mclavi. fix deadlock mclavi. fix deadlock mclavi. fix deadlock What List? View more Who Sent It? View more "/ To avoid deadlock deal with the window only after the owning thread+"/ To avoid deadlock deal with the window only after the owning thread+"/ To avoid deadlock deal with the window only after the owning thread+"/ To avoid deadlock deal with the window only after the owning thread+"/ To avoid deadlock deal with the window only after the owning thread+"// To avoid deadlock deal with the window only after the owning thread+"// To avoid deadlock deal with the window only after the owning thread+"// To avoid deadlock deal with the window only after the owning thread+"// To avoid deadlock deal with the window only after the owning thread+"// To avoid deadlock deal with the window only after the owning thread+"// To avoid deadlock deal with the window only after the owning thread+"// To avoid deadlock deal with the window only after the owning thread+"// To avoid deadlock deal with the window only after the owning thread+"// To avoid deadlock deal with the window only after the owning thread+"// To avoid deadlock deal with the window only after the owning thread+"// To avoid deadlock deal with the window only after the owning thread+"// To avoid deadlock deal with the window only after the owning thread+"// To avoid deadlock deal with the window only after the owning thread+// To avoid deadlock deal with the window only after the owning thread+// To avoid deadlock deal with the window only after the owning thread+// To avoid deadlock deal with the window only after the owning thread+// To avoid deadlock deal with the window only after the owning thread+/// To avoid deadlock deal with the window only after the owning thread+///// To avoid deadlock deal with the window only	500		and the second second		Aug 17, 2009 - Jan Kara - org.kernel.vger.linux-ext4 - 27
What List? View more Who Sent It? View more thread+? To avoid deadlock deal with the window only after the owning thread is rnelvger.linux-kernel 18,521 Hudson Builds 8,972 Mar 13,2009 kiell K. Smirnoy - org. winehq.wine-patches - P Image: Smirnoy - org. winehq.wine-patches - P	'92		hhhhhhhh.		mciavi: fix deadlock + /* To avoid deadlock deal with the window only after the owning
rnelvger.linux-kernel 18,521 Hudson Builds 8,972 Mar 13, 2009 - Kell K. Smirnov - org. winehq.wine-patches - Ø netbeans.broken_builds 10,177 hud@deadlock.netbea 8,362 Idwr-user] Java-level deadlock org.netbeans.api-changes 6,996 iss@www.netbeans.org 4,599 Ingr-level deadlock org.netbeans.api-changes 6,996 iss@www.netbeans.org 1,597 Nod8-stoc20-mingranting.Thread.run(Thread.java.619) Found 1	What List? V	/iew more	Who Sent It?	View more	thread + /* To avoid deadlock deal with the window only after the owning thread is
netbeans.broken_builds 10,177 hud@deadlock.netbea 8,362 returns ap-changes 6,996 iss@www.netbeans.org 4,599 brg-level deadlock ====================================	rnel.vger.linux-kernel	18,521	Hudson Builds	8,972	Mar 13, 2009 - Kirill K. Smirnov - org.winehq.wine-patches - Ø
org.netbeans.api-changes 6,996 iss@www.netbeans.org 4,599 lava-level deadlock ====================================	netbeans.broken_builds	10,177	hud@deadlock.netbea	. 8,362	(dwr-user) Java-level deadlock
com.mysgl.lists.commits 3,040 bugz@apache.org 1,587 8085-exec 20 at java lang. I hread.run(Thread.java:619) Found 1	org.netbeans.api-changes	6,996	iss@www.netbeans.org	4,599	lava-level deadlock: ====================================
	com.mysql.lists.commits	3,040	bugz@apache.org	1,587	8085-exec-20 at java.lang. Thread.run(Thread.java:619) Found 1
etbeans.openide.issues 2,171 FreeBSD bugmaster 1,109 deadlock. Heap PSY oungGen total 218/52K, used 213 Java-level	etbeans.openide.issues	2,171	FreeBSD bugmaster	1,109	deadlock: Heap PSYoungGen total 218/52K, used 213 Java-level
reebsd.freebsd-current 1,893 iss@openide.netbean 1,104 8085-exec.20	reebsd.freebsd-current	1,893	iss@openide.netbean	1,104	8085-exec-20
stgresql.pgsql-hackers 1,470 Andrew Morton 1,052 Маг 25, 2009 - Иван Трофимов - net.java.dev.dwr.users - 🖉 🥑	stgresql.pgsql-hackers	1,470	Andrew Morton	1,052	Mar 25, 2009 - Иван Трофимов - net.java.dev.dwr.users - 🖉 📀

Deshmukh



Symbolic Deadlockability Analysis

Concurrent Software is Modular

- Concurrent Library: methods concurrently invokable.
- Multi-threaded Client: each thread invokes library methods.
- "Whole-program approach" too expensive.

소리 에 소문에 이 제 문어 소문에 드릴 것

 Predict concurrent method invocations potentially leading to deadlock. [Williams et. al, ECOOP '05]

Symbolic Deadlockability Analysis

- Predict concurrent method invocations potentially leading to deadlock. [Williams et. al, ECOOP '05]
- Aliasing information for improved accuracy.

- Predict concurrent method invocations potentially leading to deadlock. [Williams et. al, ECOOP '05]
- Aliasing information for improved accuracy.
- Interface Contracts on methods to ensure deadlock-freedom.

《曰》《圖》《曰》《曰》 되는

- Predict concurrent method invocations potentially leading to deadlock. [Williams et. al, ECOOP '05]
- Aliasing information for improved accuracy.
- Interface Contracts on methods to ensure deadlock-freedom.
- Use interface contracts when analyzing Client code.

소리 에 소문에 이 것 같아. 소문 이 모님의

Outline



- Problem Size Reduction
- Symbolic Computation



Outline

Deadlockability Analysis

- 2 Problem Size Reduction
- 3 Symbolic Computation

4 Results

・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・

java.awt.EventQueue



イロト 不得 トイヨト イヨト 正言 ろくの

Lock-Order Graphs





	—		
Deshmukh	Symbolic Deadlockability Analysis		8 / 2

Aliasing Pattern leading to Deadlock?

Deshmukh



Such weird aliasing comes from...

```
EventQueue nextQueue;
void push (EventQueue eq) {
    ...
    nextQueue = eq;
    ...
}
```

Sequence of method calls

イロト 不得 トイヨト イヨト 正言 ろくの

Deadlockability Analysis

Deadlock-causing Aliasing Pattern

Aliasing Pattern between *lg*(**postEventPrivate**), *lg*(**wakeup**)

 $\alpha = isAliased(ob1, ob2.nextQueue) \land$

isAliased(ob2, ob2.nextQueue)

Symbolic Deadlockability Analysis

イロト 不得 トイヨト イヨト 正言 ろくの

Interface Contract



For postEventPrivate, wakeup

 \neg isAliased(ob1, ob2.nextQueue) \lor

-isAliased(ob2, ob1.nextQueue)

Call-site $\models \mathcal{I} \Rightarrow \texttt{postEventPrivate} \parallel \texttt{wakeup}$ is deadlock-free.

Approach: View from 10,000 feet

Compute:

- Lock-graphs for library methods (static analysis)
- DL-causing patterns for combinations of 2 or more methods.
- Derive Interface Contracts.

Outline





Problem Size Reduction
Lock-graph Size Reduction
Smorter Enumeration

Smarter Enumeration

Symbolic Computation



비금 사람 사람 수

Prune Lock-graphs:

Remove nodes that cannot be part of cycle



• Terminal nodes that may alias only to other terminal nodes.

Symbolic Deadlockability Analysis

비금 사람 사람 수

Prune Lock-graphs:

Remove nodes that cannot be part of cycle



• Terminal nodes that may alias only to other terminal nodes.

Prune Lock-graphs:

Remove nodes that cannot be part of cycle



- Terminal nodes that may alias only to other terminal nodes.
- Initial nodes that may alias only to other initial nodes.

(日) (周) (日) (日) (日) (日)

Smarter Enumeration by Subsumption



Symbolic Deadlockability Analysis

16/27

・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・

Smarter Enumeration by Subsumption



Deadlock-causing Aliasing Pattern (α_1)

isAliased(\mathbf{b}, \mathbf{y}) \land

isAliased(c, x)

 Δc	nr	nu		
0.0		II.U.	IN	
_				

Smarter Enumeration by Subsumption



Deshmukh



< 🗇 🕨

★ 문 ▶ ★ 문 ▶ 문 님

Deadlock-causing Aliasing Pattern (α_2)

isAliased $(\mathbf{b}, \mathbf{y}) \land$ isAliased $(\mathbf{c}, \mathbf{x}) \land$ isAliased (\mathbf{a}, \mathbf{x})

Symbolic Deadlockab	ility Analysis
---------------------	----------------

Subsumption

- α_2 subsumes $\alpha_1 : \alpha_2$ has more aliasing.
- DL with lesser aliasing \Rightarrow DL with more aliasing.
- Only enumerate "minimally" unsafe patterns.
- Disregard subsuming patterns.

Explicit Enumeration



Symbolic Deadlockability Analysis

Outline



2 Problem Size Reduction



Symbolic Computation



Aliasing Pattern Enumeration with SMT

Theorem

Enumerating all deadlock-causing aliasing patterns is NP-complete.

Symbolic Computation

- Encode Lock-Order Graphs as Inequality Constraints.
- Encode Aliasing as Equality Constraints.
- Transform Cycle Detection in a Graph to SAT of a Constraint.
- Use SMT solvers to check SAT.

Symbolic Computation

Symbolic Encoding





21/27

Symbolic Computation

Symbolic Encoding



Symbolic Encoding



Symbolic Deadlockability Analysis

Symbolic Encoding



$\mathsf{Cycle} \equiv \mathsf{UNSAT} \equiv \mathsf{deadlock!}$



Symbolic Deadlockability Analysis

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

Symbolic Algorithm



Outline

- Deadlockability Analysis
- 2 Problem Size Reduction
- 3 Symbolic Computation



< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

Experimental Results

Library Name		LOC	Time Taken (secs)	False + ves	Potential Deadlocks
ftpproxy	(ftp proxy)	1.0K	13.0	-	-
JavaFTP	(ftp client)	2.6K	9.0	-	-
cache4j	(object cache)	2.6K	15.0	-	-
netty	(network app f/w)	11.0K	14.0	-	-
apache-log4j	(logging service)	33.3K	130.1	1	1
oddjob	(job scheduler)	41.3K	250.0	-	-
hsqldb	(database engine)	157.6K	806.8	3	3
javax 1.6 sdk		534.3K	629.0	6	2
java 1.6 sdk		551.8K	1011.6	14	12
		> 1.3M	< 2880	24	18

Symbolic	Deadlockability	y Analysis
----------	-----------------	------------

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

Results

Vindication

Most deadlocks identified correspond to real, live bug reports by developers!

Library Name	Method names	Bug Report
java.awt (EventQueue)	postEventPrivate, wakeup	Sun Bug DB ids: 4913324, 6424157, 6542185.
java.awt (Container)	removeAll, addPropertyChangeListener	OS-dir mail archive.
java.util (LogManager) (Logger)	addLogger getLogger	Sun Bug DB id: 6487638.
javax.swing (JComponent)	setFont paintChildren	Bug in Jajuk player
hsqldb (Session)	isAutoCommit close	OS-dir mail archive
(Session)	close	

With Interface Contracts, we get ...

- better specification of (deadlock-free) thread-safe behavior,
- useful documentation for client developers,
- plug-in for statically analyzing existing client code, and,
- compositional flavor in reasoning about deadlocks.

《曰》《圖》《曰》《曰》 되는

Thank You!

Deshmukh

Symbolic Deadlockability Analysis

27 / 27

Lock-order Graphs

Definition

Access Expression (a.e.): ob or sequence of nested fields of ob.

Definition (Lock-order Graph G(V, E) for method m)

 $(v_1, v_2) \in E \Leftrightarrow$:

- *v*₁ aliased to some a.e. *x*,
- v₂ aliased to some a.e. y,
- Path lock (x) $\rightarrow \ldots \rightarrow \text{lock}$ (y) in cfg(m)

イロト 不得 トイヨト イヨト 正言 ろくの

Computing Lock-order Graphs

Summary = State after each program statement

- which locks currently held (Is)
- Iock-order graph (Ig)
- root nodes (rs), and,
- aliasing information,

Computing Lock-order Graphs

- Standard interprocedural summary-based forward static analysis.
- lock (x) = add x to ls, $\forall y \in ls$ add (y, x) to lg.
- unlock (x) = remove x from ls.
- Branch merge = union of summaries.
- Invocation of m = concatenate lg(m) to current lg.

Deadlockability Analysis

Given library $\mathcal{L} = \{C_1, \ldots, C_m\}$

Methods m_1, \ldots, m_k spread across classes C_1, \ldots, C_m .

Compute for all m_1, \ldots, m_k

Lock-order graphs $lg(m_1), \ldots, lg(m_k)$.

Check for each pair m_i , m_i

Is there any aliasing pattern s.t. $lg(m_i) \cup lg(m_j)$ has cycles?

Compute

 \mathcal{D} : set of all *deadlock-causing aliasing patterns*.

_				
		0.00		
			11 1 1 1	
_	~~			

・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・

So far ...

Model Checking

- Generate global state graph.
- Explore all possible interleavings.

But...

May not scale after abstraction and partial order reduction.

So far ...

Static Analysis

- Lock-acquisition order graph (*lg*) for each thread.
- Conservatively merge *lg* for concurrent threads.
- Cycle in merged graph \Rightarrow possible deadlock.

But...

Too many false positives if analysis coarse, unscalable otherwise.

Deadlock-causing Aliasing Pattern Enumeration

Definition (Subsumption)

 α_2 subsumes $\alpha_1 \ (\alpha_1 \subseteq \alpha_2)$ iff $\forall (u, v) : (u, v) \in \alpha_1 \Rightarrow (u, v) \in \alpha_2$.

Lemma (Given $\alpha_1 \subseteq \alpha_2$)

 α_1 is deadlock-causing $\Rightarrow \alpha_2$ is deadlock-causing.

Definition (Minimally Unsafe)

 α minimally unsafe iff for any (u, v), $\alpha - (u, v)$ is safe.

We only need to consider minimally unsafe patterns.

_					
	De	hm	ni il	٢h	
	65		IUI	NЦ	

< □ > < □ > < 亘 > < 亘 > < 亘 > < 亘 > < □ > <

Deadlock-causing Aliasing Pattern Enumeration

Subsumption

- α_2 subsumes $\alpha_1 \Rightarrow \alpha_2$ has more aliasing than α_1 .
- $\alpha_1 \subseteq \alpha_2$: α_1 is deadlock-causing $\Rightarrow \alpha_2$ is deadlock-causing.
- α minimally unsafe if removing any aliasing makes it safe.

We only need to enumerate minimally unsafe patterns!

소리 에 소문에 이 제 문어 소문에 드릴 것

Encoding Lock-Graph G(V, E)

• $x(v_i)$: topological rank of $v_i \in V$. • $\Psi(G) = \bigwedge_{(v_i, v_j) \in E} (x(v_i) < x(v_j)).$

Encoding Aliasing Pattern α

$$\Psi(\alpha) = \bigwedge_{(\mathbf{v}_i, \mathbf{v}_j) \in \alpha} (\mathbf{x}(\mathbf{v}_i) = \mathbf{x}(\mathbf{v}_j))$$

Reduction to SAT

 $\alpha \triangleright G$ has a cycle iff $\Psi(\alpha, G) = \Psi(G) \land \Psi(\alpha)$ is unsatisfiable.

		-		
	es		шк	
_			i u i i	

イロト 不得 トイヨト イヨト 正言 ろくの

A few more (sound) filters...

Prune ...

- locks corresponding to final fields.
- private fields not accessed outside constructor/finalizer.
- immutable constants.
- private objects that cannot escape scope of methods.

Joint Lock-Order Graph without Aliasing



Derive Interface Contracts

Definition (Interface Contract)

Compute \mathcal{D} : all deadlock-causing aliasing patterns.

$$\mathcal{I}(m_i, m_j): \bigwedge_{\alpha \in \mathcal{D}} \bigvee_{(\boldsymbol{e}_i, \boldsymbol{e}_j) \in \alpha} \neg \mathsf{isAliased}(\boldsymbol{e}_i, \boldsymbol{e}_j).$$

Call-site of m_i, m_j satisfies $\mathcal{I} \Rightarrow m_j \parallel m_j$ is deadlock-free.

_			
	00	hm	zh
	1-1-1		