

<http://harmony.apache.org/subcomponents/drlvm/TM.html>

- 1.
- 2.
- 2.1.
- 2.2.
- 2.3.
- 2.4.
- 3.
- 3.1.
- 3.2. VM
- 3.3.
- 4.
- 4.1.
- 4.1.1. Native
- 4.1.2. Java*
- 4.2.
- 4.3.
- 4.3.1. Native
- 4.3.2. Java*
- 4.4.
- 4.5.
- 4.6.
- 4.6.1.
- 4.6.2.
- 4.6.3.
- 5.
- 5.1. Java*
- 5.2.
- 5.3. Stop-the-world
- 5.4.
- 5.5.
- 6.

	Nadya Morozova, Andrey Chernyshev: .	200665

(Dynamic Runtime Layer, DRL)DRL

DRLVM

DRL

- VM
-
-
-

TM.Java*POSIX.J2SE.Java*

JVM

- porting layer
- native layer
- Java*Java*Java* layer

native layerJava* layerporting layer

porting layerportabilitynative layerporting layerJava*Java* layerJava*native layer 1

{{ http://harmony.apache.org/subcomponents/drlvm/images/ThreadingSystem.gif }}
1.

- J2SE API[1|#ref1]JVMTI[2|#ref2]JNI[3|#ref3]
- Apache Porting Layer[4|#ref4]
- Harmony hythread [8|#ref8]
-
- DRLVMjust-in-time, JITmonitor

VM

2

- **VM core**{{java.lang.Thread}}{{thread_externals.h}}
- **garbage collector**GCnative layer
- **porting layer**native layerAPR{{apr_thread_ext}}
- **just-in-time compiler**JITVM Java* layer{{thread_helpers}}

<pre> {{ http://harmony.apache.org/subcomponents/drlvm/images/tm_in_vm.gif }} </pre>
2. VM

porting layerTMApache Porting Layer(APR)TMVM OS APIAPR
APRAPRLinux*Windows* IA-32

VM

nativeJava*

Native

NativeHarmony hythread} }Java* { {waitpark{{join}}{{sleep}}Java*

Native

hythread.h

{{hythread}}[8|#ref8]

-
- (Parking)
-
-
-

hythread_ext.h

{{hythread}}

-
-
-
-
- (Latch)
-
- (Attributes access)
- (Querying state of the thread)
-
-
- (thin)
- (Querying the thread state)
- (Thread attributes access)
-
-

Java*

Java*native layerJava*

Java*Java*kernel classesJNIJVMTIJava*

jthread.h

{{java.lang.Object}}{{java.lang.Thread}} API

-
-
-
-
-
-
- Parking
-

ti_thread.h

JVMTI{{java.lang.management}} API

-
-
-
-
- CPU
- (Peak)
- (Raw)

thread_helpers.h

(stub)JIT

VM{{hythread.hhythread_ext.hjthread.h}} {{ti_thread.h}}{{thread_private.h}}

registeredattaching a thread

- {{hythread_attach()}}native layer
- {{jthread_attach()}}{{java.lang.Thread}}Java* layer

attaching function

- Native threadnative layer(native)
- Java* thread: Java* layer{{java.lang.Thread}}Java*

VM

Native

native layerOS

{{HyThread}}native

{{
<http://harmony.apache.org/subcomponents/drlvm/images/NativeUnattachedThread.gif>
}}

3. Native

native

Suspension		
IDATA	suspend_request	
int16	suspend_disable_count	
hylatch_t	safe_region_event	

hysem_t	resume_event	
hythread_event_callback_proc	safepoint_callback	
Basic manipulation fields		
hythread_group_t	group	Java*GC
hythread_t	next	
hythread_t	prev	
apr_thread_t *	os_handle	OS
void *	private_data	Java* layerJava
Boolean	exit_request	
IDATA	exit_value	
Synchronization support		
hysem_t	park_event	Parking
hysem_t	sleep_event	Sleeping
hylatch_t	join_event	join
hycond_t	current_condition	
State		
IDATA	state	JVMTI
Attributes		
char *	name	
IDATA	priority	
IDATA	daemon	
Others		
IDATA	thread_id	IDlock word record
apr_pool_t *	pool	
apr_threadattr_t *	apr_attrs	APR
hythread_entrypoint_t	start_proc	
void *	start_proc_args	
void *	thread_local_storage	
void *	big_local_storage	

{{thread_private.h}}

Java*

Java*{{JVMThread}}JVMTI 4

<pre> {{ http://harmony.apache.org/subcomponents/drlvm/images/JavaAttached.gif }} </pre>
4. Java*

Java*

JNIEnv *	jenv	Java*JNI
jthread	thread_object	hythread_tjthread
hycond_t	monitor_condition	wait/notify Java*
jthrowable	stop_exception	
jlong	blocked_time	
jlong	waited_time	

JVMTILocalStorage	jvmti_local_storage	JVMTI
jobject	contended_monitor	
jobject	wait_monitor	
jobject *	owned_monitors	
int	owned_monitors_nmb	
apr_pool_t *	pool	APR
jobject	thread_ref	
IDATA	suspend_request	{{java.lang.Thread}}

{{thread_private.h}}
 {{java.lang.Thread}} 3 4

Java*Java*GC 5

{{ http://harmony.apache.org/subcomponents/drlvm/images/thread_groups.gif }}
5.

APR 6

{{ http://harmony.apache.org/subcomponents/drlvm/images/Synchronizer_mutex.gif }}
6.

- APRAPR conditional variableAPR(APR mutex)Apache Portable Runtime
- TM(TM conditional variable)TM(TM mutex){{wait}}APR{{wait}}safe suspension mode
- [Thin](#)Thin monitorinflatable lockJava*
- SemaphorePOSIX
- Java*Java monitor{{java.lang.Object}}
- JVMTI JVMTI raw monitorJVMTI
- parkunpark{{java.util.concurrent}}

APRAPR

Java*DRL——*thin*

thin-fat[6|[#ref6](#)]

- Java*
- fatfat

thinthinJava*thinJava*thinJava*

thinJava*[6]thinTMnative layerJava*Java*Java*TMJava* layer

*lock word*thinfat

0

{{
http://harmony.apache.org/subcomponents/drlvm/images/uninflated_lockword.gif
}}

7. Contention Bit0

- Contention bit0
- Thread ID (15 bits): ID0
- Recursion count: 1
- Reservation bit: [7|#ref7]
- 10 bitsTMJava*

contention bit1thinfat

{{
http://harmony.apache.org/subcomponents/drlvm/images/inflated_lockword.gif
}}

8. Contention Bit1

- Contention bit: 1
- Fat Lock ID (20 bits): fatID
- Reservation bit: [7|#ref7]
- 10 bitsTMJava*

fatIDfat

{{
http://harmony.apache.org/subcomponents/drlvm/images/inflated_thin_monitor.gif
}}

9. ThinFat

{{hythread_thin_monitor_try_enter()}}

{{
http://harmony.apache.org/subcomponents/drlvm/images/Lock_reservation.gif
}}

10. Thin

reservation bitrecursion count1monitor enter-(compare-and-swap, CAS)

lock free""CASlock busyfat

fatIDFatfat

Java*

Java*

- 1. Thread() } native { { HyThread } } { { JVMIThread 1. VM core { { java.lang.Thread.start() } } 1. { { java.lang.Thread.start() } } { { java.lang.VMThreadManager.start() } } Java* layer { { jthread_create() } } 1. jthread_create() } } { { jthread_create_with_function() 1. { { jthread_create_with_function() } } { { hythread_create() } } { { wrapper_proc() } } 1. { { hythread_create() } } APR porting layer { { apr_thread_create() } } { { fork { { { wrapper_proc() } } 1. { { thread_start_proc() } } { { wrapper_proc() } } 1. { { wrapper_proc() } } VM core { { vm_attach() } } { { jvmti_send_thread_start_end_event() } } { { JVMTI_EVENT_THREAD_START } } 1. { { wrapper_proc() } } { { java.lang.Thead.run() } } Java* 1. { { Thread.run() } } { { jvmti_send_thread_start_end_event() } } { { JVMTI_EVENT_THREAD_END } } { { vm_detach } }

<div> {{ http://harmony.apache.org/subcomponents/drlvm/images/ThreadCreation.gif }}</div>
11. Java*

nativeHyThread } } { { JVMIThread { { java.lang.Thread } } { { java.lang.Thread } } native native

native layerAPRsafe suspensionVM"hard"

T1T2T1T2T1 { { hythread_suspend(T2) } } T2

- 1. { { hythread_suspend(T2) } } T2T2 { { hythread_suspend(T2) } }
- a. T2 { { hythread_suspend(T2) } } 12
- a. T2 { { hythread_suspend() } } T2
- 1. T2T1 { { hythread_resume(T2) } }

T2

- 1. T2 { { hythread_safe_point() } } T2T1T1 { { hythread_resume(T2) } } T2
- 1. T2 { { hythread_suspend_ensable() } } { { suspend_disable_count } } T2T1T2
- 1. T2 { { hythread_suspend_disable() } } { { suspend_disable_count } }

Java* 12 GCJava*

<div> {{ http://harmony.apache.org/subcomponents/drlvm/images/safeRegion.gif }}</div>
12.

T2T2T2thin

GCJava*13

<div> {{ http://harmony.apache.org/subcomponents/drlvm/images/safePoint.gif }}</div>
13.

{ { hythread_safe_point() } } { { wait } } { { hythread_resume() } }

Stop-the-world

stop-the-world 14 GCJava*GCJava*

<pre> {{ http://harmony.apache.org/subcomponents/drlvm/images/SuspendAll.gif }} </pre>
14.

{{hythread_suspend_all()}}Java*GCJava*GC

thinthinVM coreT1T2 15

<pre> {{ http://harmony.apache.org/subcomponents/drlvm/images/locking.gif }} </pre>
15.

OT1{{hymutex_lock()}}T1

T2{{hymutex_lock()}}T2T1T1{{hymutex_unlock()}}T2T2T2

Java*VM coreJava*Java* 16

Java*

1. JIT{{hythread_monitor_enter()}}JIT
1. {{hythread_monitor_enter()}}VM core{{vm_object_get_lockword_addr()}}Java*
1. {{thin_monitor_try_lock()}}
1. Java*Java*native{{thin_monitor_try_enter()}}Java*Java*native{{M2nFrame}}
1. {{jthread_monitor_enter()}}JNIJava*

<pre> {{ http://harmony.apache.org/subcomponents/drlvm/images/Monitors.gif }} </pre>
16. Java*

[1] J2SE 1.5.0 specification, <http://java.sun.com/j2se/1.5.0/docs/api/> <ac:structured-macro ac:name="anchor" ac:schema-version="1" ac:macro-id="459531ce-7263-4085-ad05-957f824b01a4"><ac:parameter ac:name="">ref1</ac:parameter></ac:structured-macro>

[2] JVM Tool Interface Specification, <http://java.sun.com/j2se/1.5.0/docs/guide/jvmti/jvmti.html> <ac:structured-macro ac:name="anchor" ac:schema-version="1" ac:macro-id="529737f8-b1a9-4890-8287-71e333534938"><ac:parameter ac:name="">ref2</ac:parameter></ac:structured-macro>


[3] Java* Native Interface Specification, <http://java.sun.com/j2se/1.5.0/docs/guide/jni/spec/jniTOC.html> <ac:structured-macro ac:name="anchor" ac:schema-version="1" ac:macro-id="c5f33b48-8d2b-4fe2-937b-049e6be92ffe"><ac:parameter ac:name="">ref3</ac:parameter></ac:structured-macro>

[4] Apache Portable Runtime project, <http://apr.apache.org/> <ac:structured-macro ac:name="anchor" ac:schema-version="1" ac:macro-id="ca6282e2-ea86-466a-8098-751f04cab4d"><ac:parameter ac:name="">ref4</ac:parameter></ac:structured-macro>

[5] POSIX standard in threading, <http://www.opengroup.org/onlinepubs/009695399/idx/threads.html> <ac:structured-macro ac:name="anchor" ac:schema-version="1" ac:macro-id="812eec82-c8ed-4807-8b26-05ecef4d25d9"><ac:parameter ac:name="">ref5</ac:parameter></ac:structured-macro>

[6] David F. Bacon, Ravi Konuru, Chet Murthy, Mauricio Serrano, Thin locks: featherweight synchronization for Java, <http://portal.acm.org/citation.cfm?id=277734> <ac:structured-macro ac:name="anchor" ac:schema-version="1" ac:macro-id="97d5eb6e-359b-4b9d-a1ae-1eb4a9dfc380"><ac:parameter ac:name="">ref6</ac:parameter></ac:structured-macro>

[7] Kiyokuni Kawachiya Akira Koseki Tamiya Onodera, Lock Reservation: Java Locks Can Mostly Do Without Atomic Operation, <http://portal.acm.org/citation.cfm?id=582433> <ac:structured-macro ac:name="anchor" ac:schema-version="1" ac:macro-id="75c17642-c5f1-44c7-b55b-ec82e3259490"><ac:parameter ac:name="">ref7</ac:parameter></ac:structured-macro>

[8] [HyThread] documentation, http://svn.apache.org/viewvc/incubator/harmony/enhanced/classlib/trunk/doc/vm_doc/html/group__Thread.html  <ac:structured-macro ac:name="anchor" ac:schema-version="1" ac:macro-id="73972c0b-4e74-4c59-b427-3b7a33d37c84"><ac:parameter ac:name="">ref8</ac:parameter></ac:structured-macro>

- Other brands and names are the property of their respective owners.

(Contributors: (asuka@ustc.edu)(yuzhang@ustc.edu.cn))