

#	Source		Pattern	
	Original	Modified	Original	Modified
7	<pre> <del>-synchronized void enqueue(final long seqno,</del>     final boolean lastPacketInBlock,     final long offsetInBlock) { <del>- if (running) {</del>     final Packet p = new Packet(...);     LOG.debug(...);      ackQueue.addLast(p);     notifyAll(); <del>}</del> <del>}</del> </pre>	<pre> +void enqueue(final long seqno,     final boolean lastPacketInBlock,     final long offsetInBlock) {      final Packet p = new Packet(...);     LOG.debug(...); + synchronized (this) +     if (running) { +         ackQueue.addLast(p); +         notifyAll(); +     } + } </pre>	<pre> synchronized (obj) {     statements1     statements2 } </pre>	<pre> statements1 synchronized (obj) {     statements2 } </pre>
8	<pre> synchronized (buffers) {     if (...) { <del>-     if (spillWriter != null) {</del> <del>-         spillWriter.close();</del> <del>-     }</del>         isFinished = true;     } } </pre>	<pre> synchronized (buffers) {     if (...) {         isFinished = true;     } } +if (spillWriter != null) { + spillWriter.close(); +} </pre>	<pre> synchronized (obj) {     statements1     statements2 } </pre>	<pre> synchronized (obj) {     statements2 } statements1 </pre>
9	<pre> <del>-Membership membership = null;</del> public boolean hasMembers() {     if (membership == null) setupMembership();     return membership.hasMembers(); } synchronized void setupMembership() {     if ( membership == null ) {         membership = new Membership(             super.getLocalMember(true));     } } </pre>	<pre> +volatile Membership membership = null; public boolean hasMembers() {     if (membership == null) setupMembership();     return membership.hasMembers(); } synchronized void setupMembership() {     if ( membership == null ) {         membership = new Membership(             super.getLocalMember(true));     } } </pre>	<pre> T foo; </pre>	<pre> volatile T foo; </pre>
10	<pre> <del>-private volatile int requestCount;</del> <del>-private volatile int errorCount;</del> <del>- requestCount++;</del> </pre>	<pre> +private final AtomicInteger requestCount = + new AtomicInteger(0); +private final AtomicInteger errorCount = + new AtomicInteger(0); + requestCount.incrementAndGet(); </pre>	<pre> volatile T foo; </pre>	<pre> TT foo; </pre>
11	<pre> <del>-private volatile long genstamp;</del> <del>-public synchronized long nextStamp() {</del> <del>-     this.genstamp++;</del> <del>-     return this.genstamp;</del> <del>}</del> </pre>	<pre> +private AtomicLong genstamp = + new AtomicLong(); +public long nextStamp() { +     return genstamp.incrementAndGet(); + } </pre>		