

Denis Caromel Institut universitaire de France (IUF) INRIA Sophia-Antipolis – CNRS – I3S – Université de Nice

> Luis Mateu DCC – Universidad de Chile

Eric Tanter DCC – Universidad de Chile – Ecole des Mines de Nantes

- 1. SOM Principles
- 2. Scheduling Strategy and API
- 3. Examples
- 4. Implementation and Benchmarks

Denis Caromel









# public class Buffer { List list= new LinkedList(); public void isEmpty() { return list.size()==0; } public void isFull() { return list.size()==10; } public void put(Object o) { list.add(o); } public Object get() { return list.remove(o); } }

### The Full Scheduler for buffer-like classes

















Scher	duler API
scheduling	queue management
void schedule(Request	) Iterator iterator()
void scheduleAll <sup>1</sup>	boolean hasRequest <sup>1</sup>
void scheduleOldest <sup>1</sup>	int requestCount <sup>1</sup>
void scheduleYoungest	1
void scheduleOlderTha	n <sup>2</sup>
void scheduleAllOlder	Than <sup>2</sup>
void scheduleYoungerT	han <sup>2</sup>
void scheduleAllYoung	erThan <sup>2</sup>
<pre>scheduleOldest ();</pre>	scheduleAll(new RequestFil
<pre>scheduleOldest ("get");</pre>	boolean accept(Request r
<pre>scheduleAll("exitRead");</pre>	if return true
scheduleOlderThan("foo","ba	ar"); if return fals
Denis Caromel	···· } );

Fair	<pre>public FairRWScheduler     extends Scheduler {     RWCoordinator c;     // initialized in constructor</pre>
Reader Writer	<pre>public void schedule() {     scheduleAll(new String[]{         "exitRead", "exitWrite",         "getReaders", "isWriting"}),</pre>
SOM	<pre>if(!c.isWriting()) {     if(c.getReaders() &gt; 0)         scheduleOlderThan(</pre>
Denis Caromel	<pre>"enterRead","enterWrite"); else scheduleOldest(); } }</pre>











## SÖM: Sæquentiål Øbjæct Mönitör

- · An alternative to standard, interleaving, monitors
- Key points:

Denis Caromel

- Thread-less scheduler, Thread-less Active Object
- Threads collaborate for *Mutual Scheduling*
- Separation of concerns:
  - Synchronizing + Synchronized code
- Expressive and Efficient:
  - Full access to pending calls
  - Avoids context-switches
- Stateful (object) vs. Pending Function Calls :
  - Reason about data structure state rather than call interleaving
  - Sequentiality: easier to reason about, to maintain, to reuse Ŕ













number of	Java	Condition		SOM	SOM
consumers	monitors	Variables	SOM	Guards	Chords
1	390	1057	796	802	1203
2	510	1088	864	885	1229
4	771	1114	942	948	1265
8	1416	1120	1010	1026	1317
16	2823	1213	1106	1208	1541
32	7317	1375	1604	1593	1958
64	23479	2010	2322	2270	2708
128	80422	3234	3604	3442	4083

# Windows 2000, JDK 1.4.2

# Linux 2.4, JDK 1.4.2

number of	Java	Condition		SOM	SOM
consumers	monitors	Variables	SOM	Guards	Chords
1	1006	1905	1656	1642	1954
2	1225	2018	1708	1690	2029
4	1918	2276 🤇	1891	1839	2148
8	5723	2412	2125	1982	2276
16	16005	2451	2435	2199	2488
32	49767	2659	3156	2766	3123
64	133612	2946	4407	3771	4196
128	358218	3049	6653	5259	5934
nis Caromel		R			

number of	Java	Condition		SOM	SOM	Cond. Vars
consumers	monitors	Variables	SOM	Guards	Chords	JDK1.5
1	531	1279	1199	1157	1425	537
2	732	1234	1225	1196	1518	586
4	1131	1293	1333	1309	1573	556
8	2195	1281	1495	1378	1660	581
16	4312	1276	1851	1549	1916	592
32	9714	1350	2543	1969	2371	645
64	31637	1587	4305	2885	3601	850
128	95391	1762	7331	4414	5683	1062
Denis Carom	el		R			31

# Linux 2.6, JDK 1.5







# An Example: The Bounded Buffer

Ŕ

### Standard object:

### Scheduler:

35

```
class Buffer {
    class Buffer {
        class Buffer class Buffer {
        class Buffer Scheduler {
            extends Scheduler {
            Buffer buf;
            void schedule() {
            if (buf.isEmpty())
            scheduleOldest("put");
            else if (buf.isFull())
            scheduleOldest("get");
            else
            scheduleOldest();
        }
        }
    }
}
```

Denis Caromel