

Lock-Based Software Transactional Memory for Real-Time Systems

CATHERINE E. NEMITZ AND JAMES H. ANDERSON



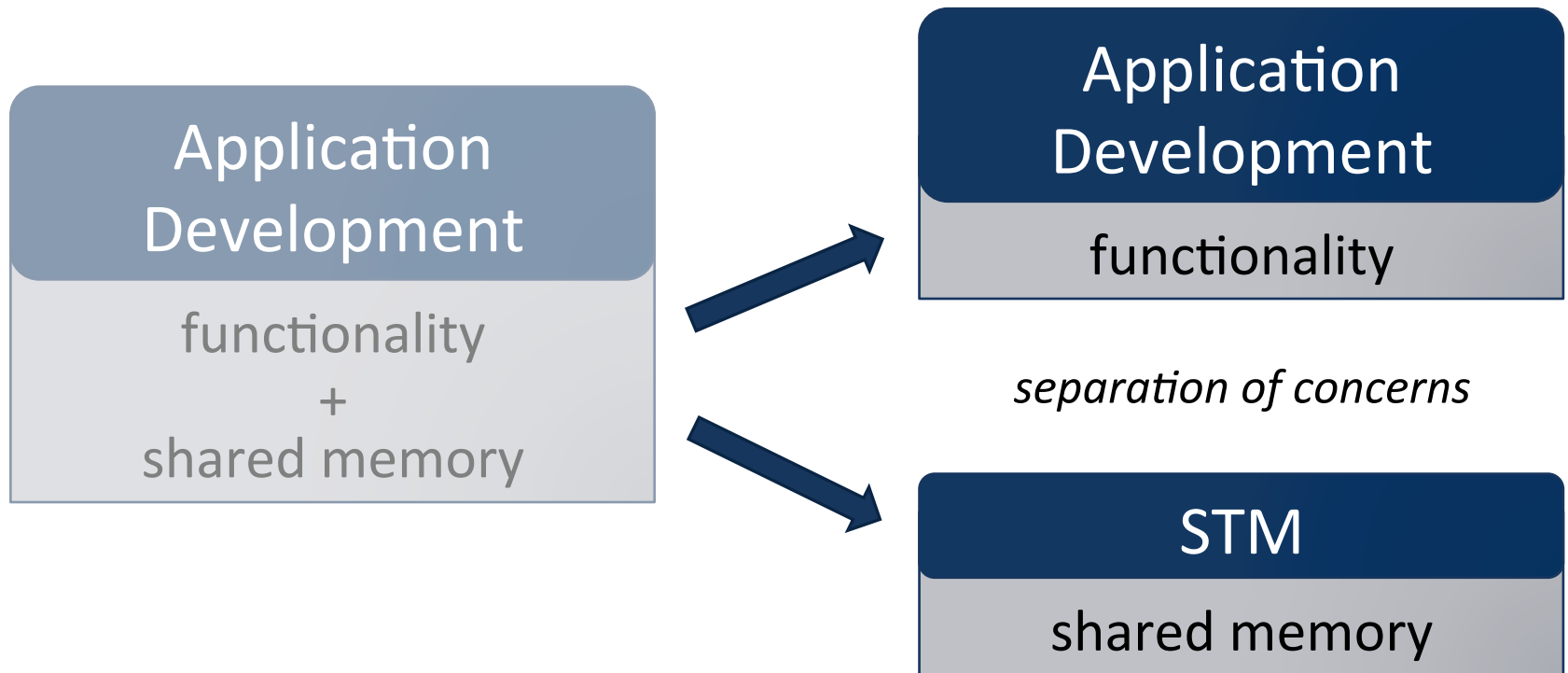
THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL

Software Transactional Memory (STM)

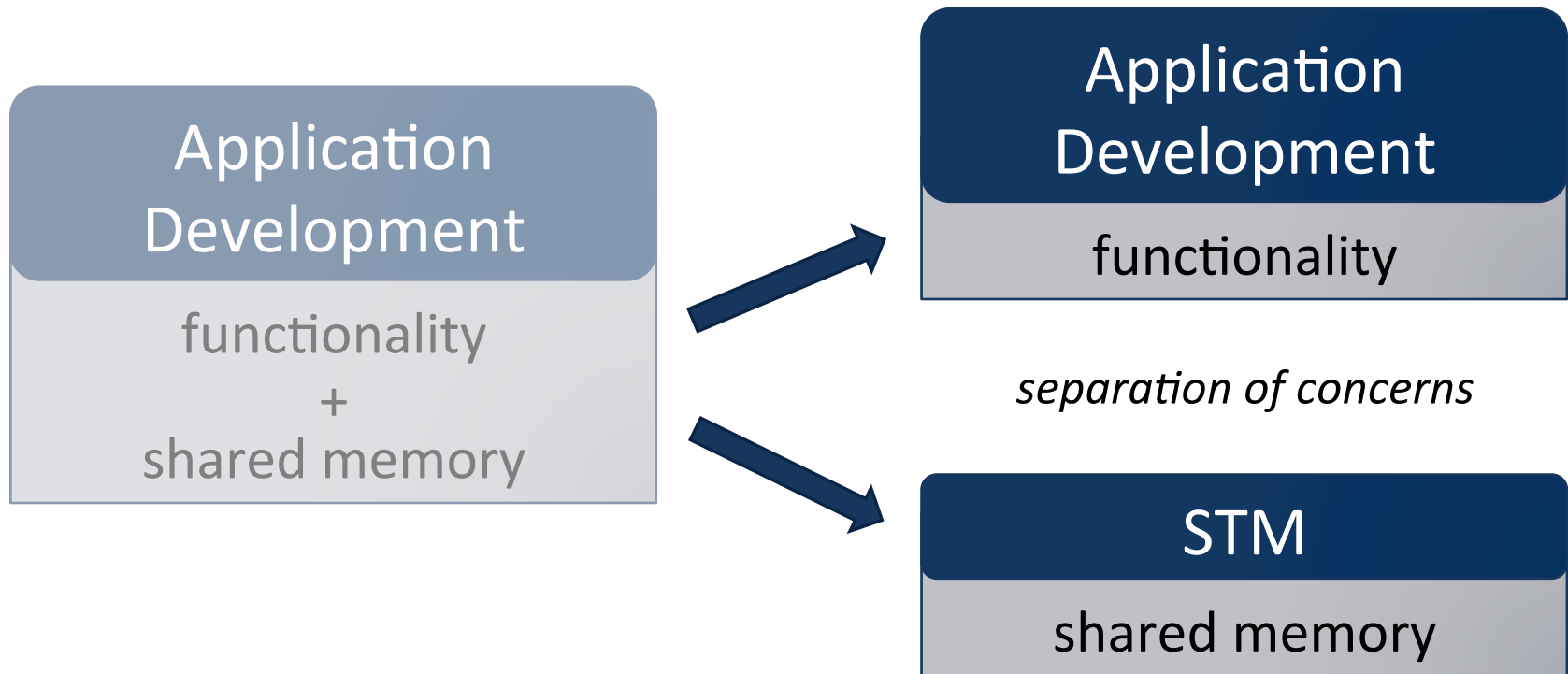
Application
Development

functionality
+
shared memory

Software Transactional Memory (STM)



Software Transactional Memory (STM)



Real-time concern: bound
priority-inversion blocking

A New Model: Lock-Based STM


Application

```
...  
begin <atomic>  
...  
end <atomic>  
...
```

A New Model: Lock-Based STM

Application

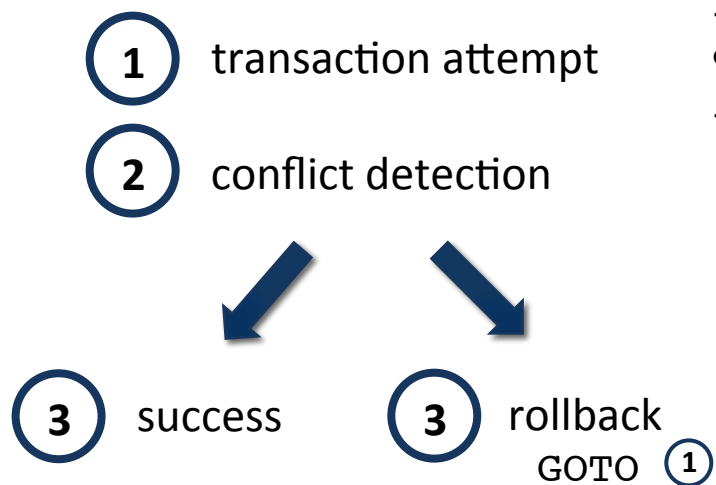
```
...  
begin <atomic>  
...  
end <atomic>  
...
```



Transaction

A New Model: Lock-Based STM

Traditional STM



Application

```
...  
begin <atomic>  
...  
end <atomic>  
...
```

} Transaction

A New Model: Lock-Based STM

Traditional STM

① transaction attempt

② conflict detection

③ success

③ rollback
GOTO ①

Application

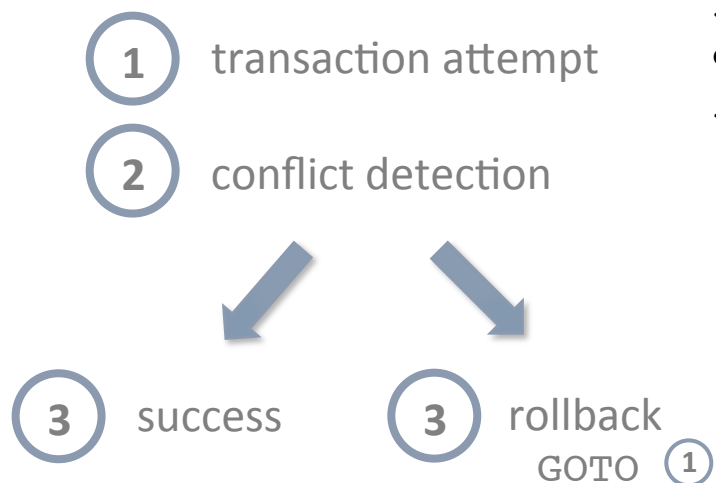
```
...  
begin <atomic>  
...  
end <atomic>  
...
```

} Transaction

Must bound number
of retries.

A New Model: Lock-Based STM

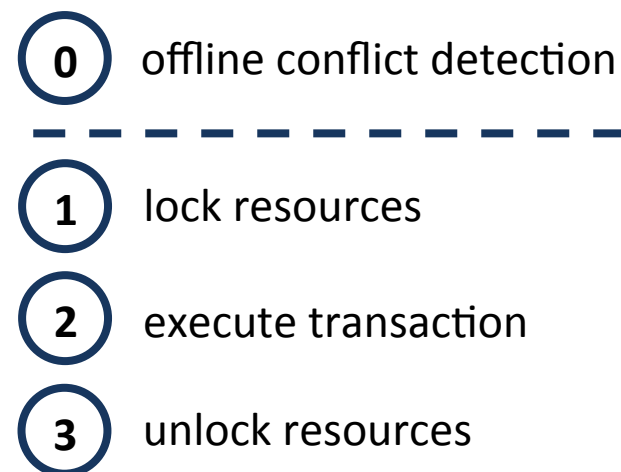
Traditional STM



Application

```
...  
begin <atomic>  
...  
end <atomic>  
...
```

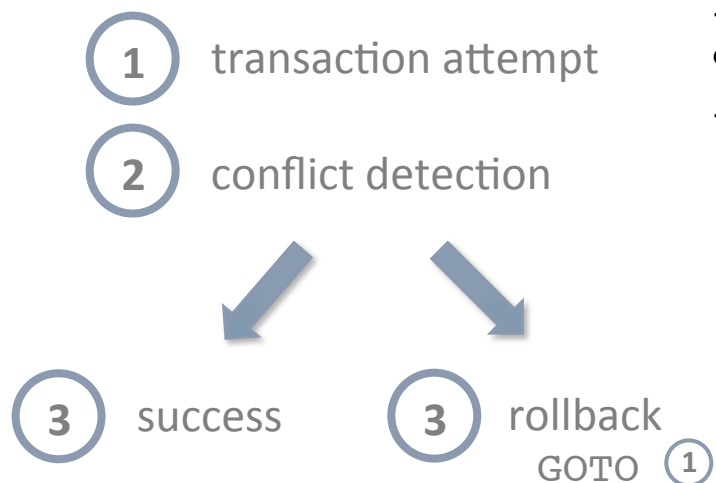
Lock-Based STM



Must bound number
of retries.

A New Model: Lock-Based STM

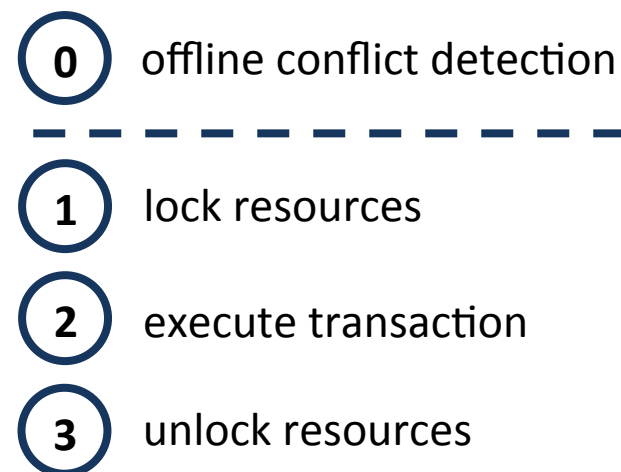
Traditional STM



Application

```
...  
begin <atomic>  
...  
end <atomic>  
...
```

Lock-Based STM



Must bound number of retries.

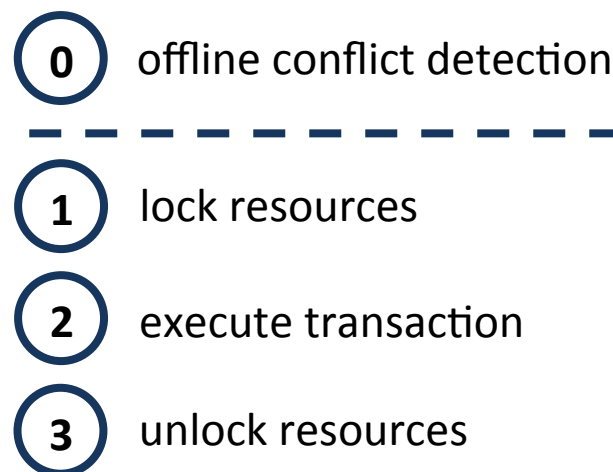
Transactions are guaranteed to commit with no retries.

A New Model: Lock-Based STM

Additional Research Directions

- Determine lock granularity
- Allow additional resource types
- Explore optimizations based on critical-section length or other parameters

Lock-Based STM



Transactions are guaranteed to commit with no retries.