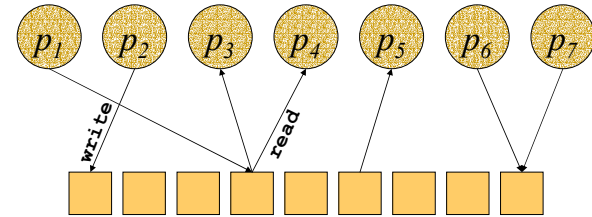


Lower Bounds for Adaptive Collect and Related Objects

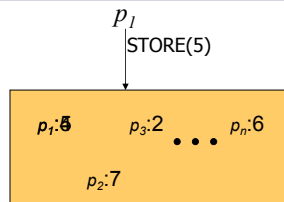
Hagit Attiya, Technion
 Faith Ellen Fich, University of Toronto
 Yaniv Kaplan, Technion

Asynchronous Shared-Memory Systems



- All registers are multi-reader
- Either single(owner)-writer or multi-writer.

The Collect Problem



A COLLECT returns (at least) the latest preceding store.

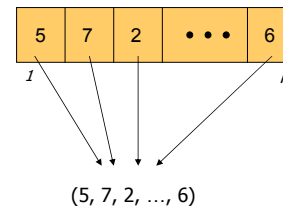
- Building block to solve
 - Long-Lived Weak Test&Set
 - Renaming
 - Atomic Snapshots

```

test&set (code for pi):
STORE (L)
v = COLLECT ()
if ∃j! = i , v[j]! = L
    STORE (L)
    return fail
return win

reset:
STORE (⊥)
    
```

The Collect Problem

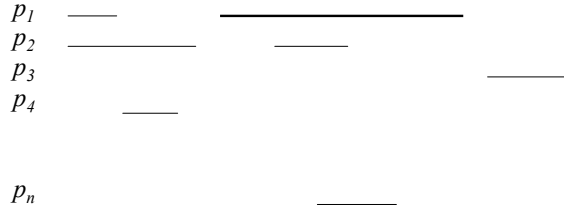


- There is a simple array implementation
 - Using n single-writer registers
- What happens when only few processes participate?

Total Contention

The number of processes taking a step **before** or **during** the operation.

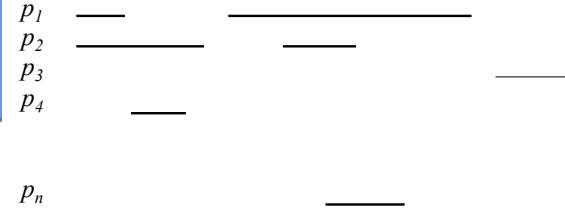
[Attiya, Fouren, 1998]



Total Contention

The number of processes taking a step **before** or **during** the operation.

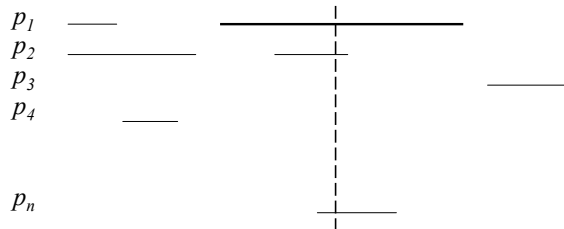
[Attiya, Fouren, 1998]



Point Contention

Max number of processes taking steps **simultaneously** during the operation.

[Attiya, Fouren, 1999]



Point Contention

Max number of processes taking steps **simultaneously** during the operation.

[Attiya, Fouren, 1999]

Clearly, point contention
 \leq interval contention
 \leq total contention

Adaptive Algorithms

An algorithm is f -adaptive to total contention if executing any operation takes $f(k)$ steps
 k is the total contention of the operation

Same for interval or point contention

Is Concurrent-Access Necessary?

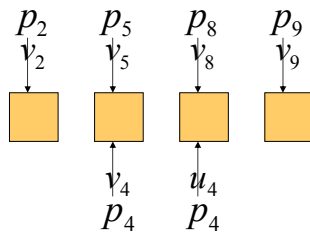
- Current adaptive collect algorithms require lots of multi-writer registers
- Indeed, a constant number of multi-writer registers does not suffice for long-lived collect
 [AfeK, Boxer and Touitou, 2000]
- Same for one-shot collect
 [Aguilera, Englert and Gafni, 2003]

But...

- Proofs are complicated
- Bounds are weak

Covering

[Burns and Lynch, 1993]

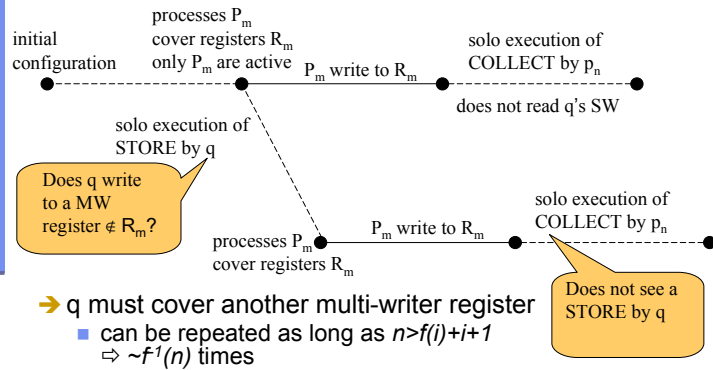


Store Operations Must be Seen

initial configuration processes P_m cover registers R_m solo execution of COLLECT by p_n
 only P_m are active p_m write to R_m

- » p_n reads at most $f(m)$ single-writer registers
- » If $n > f(m) + m + 1$, p_n does not read the single-writer register of some process $q \notin P_m$
- » COLLECT does not see a STORE of q

Store Operations Must be Seen



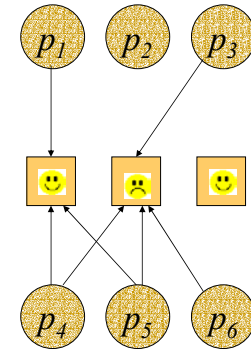
July 2004

Lower Bounds for Adaptive Collect and Related Objects / Attiya, Fich and Kaplan

13

Access Restrictions

- Restrict the competition on a register by c
 - At most c processes cover the register simultaneously
- Exclusive-write registers prohibit competition ($c=1$)
 - Different processes may cover at different times



July 2004

Lower Bounds for Adaptive Collect and Related Objects / Attiya, Fich and Kaplan

14

Bounds on Concurrent-Write Registers

With an unlimited number of c -write registers

- An f -adaptive collect algorithm that adapts to point contention requires $\Omega(f^1(n/c))$ multi-writer registers
- An f -adaptive collect algorithm that adapts to total contention requires a number of multi-writer registers that grows with n .

July 2004

Lower Bounds for Adaptive Collect and Related Objects / Attiya, Fich and Kaplan

15

Contention-Free Step Complexity of Collect

- Collect has $\Omega((n-1)/c)$ solo-execution length, when using only c -write registers
 - Must access $\Omega((n-1)/c)$ different registers (latency)
- ⇒ Exclusive-write registers do not reduce the contention-free step complexity of collect

July 2004

Lower Bounds for Adaptive Collect and Related Objects / Attiya, Fich and Kaplan

16

Contention-Free Step Complexity of Weak Test&Set

- There is a weak test&set implementation with $O(\log(n))$ contention-free step complexity, using exclusive-write registers
 - $\Omega(n)$ lower bound w/ single-writer registers
- The solo-execution length of weak test&set is $\Omega(\log(n)/(\log(c) + \log\log(n)))$, when using only c-write registers
 - Holds for **obstruction free** implementations.
[Herlihy, Luchangco and Moir, 2003]

Extensions...

- All results for collect extend to **nondeterministic** implementations of **sensitive** objects using **historyless** objects
[Fich, Herlihy and Shavit, 1998]
- Sensitive objects ensure that operations must be “seen”.
- Related (but not identical) to **perturbable** objects
[Jaynti, Tan and Toueg, 2000]
 - Increment, fetch&add, b-valued c&s ($b \geq n$), ...

Open Problems

- Renaming, and weak test&set.
 - Our space lower bound techniques do not apply.
 - Exclusive-write registers help for weak test&set but not for collect
 - ⇒ A possible gap?
- Our proofs bounds the number of multi-writer registers **actually used** by the algorithm
 - Matches the upper bound of [Attiya, Fouren, Gafni, 2001]
- A lot more multi-writer registers are **allocated**.
 - What is the lower bound?