PARALLEL SIMULATION OF A.T.M. SWITCHES

A.S.McGOUGH I.MITRANI

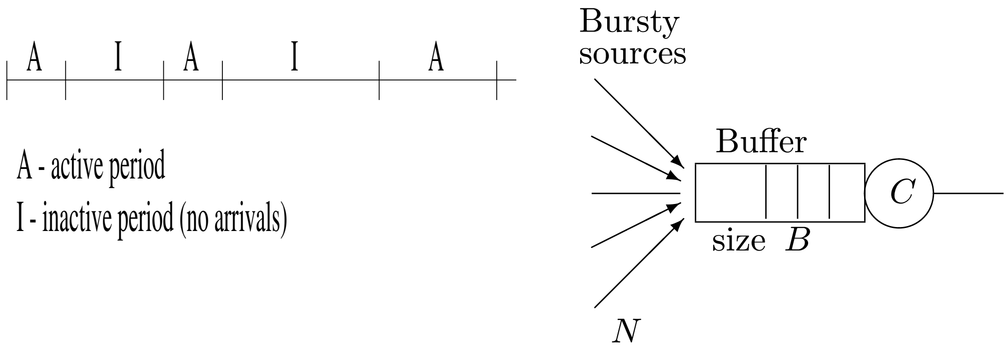
UNIVERSITY OF NEWCASTLE UPON TYNE

26/4/1997

1: What is an ATM switch ?

An ATM switch is a "statistical multiplexor" switch which forms part of an ATM network.

- Packets of data, called "cells", arrive from several input streams.
- Cells are combined in the order of arrival.
- Cells are output onto a single departure stream.



Assumptions for the simulation.

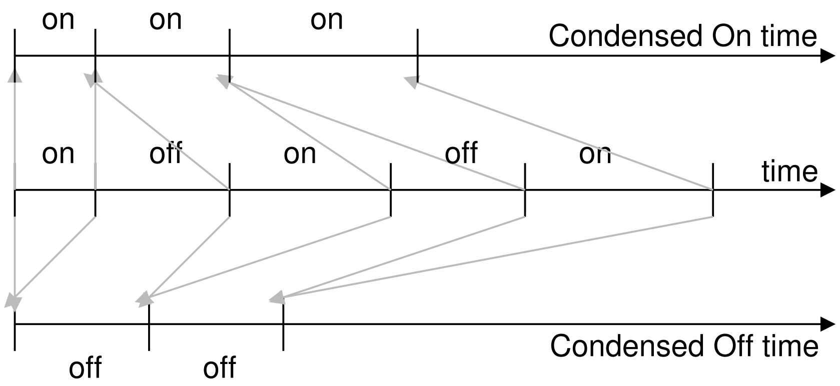
- Cells are of fixed length and arrive from N input streams.
- The sources have a Bursty nature, with periods of activity and inactivity.
- The switch can handle C cells per unit of time.
- There is a fixed amount of buffer space in the switch for B cells.
- When the buffer is full any cell that arrives will be lost.

2: Outline of method

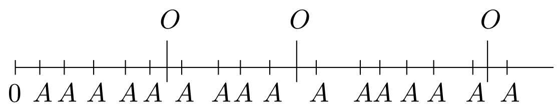
The program iterates the following stages until M cells have been simulated.

- Generate arrivals
  - Arrival assumptions
    - Any distribution can be simulated.
    - But with bursty "on" / "off" property.

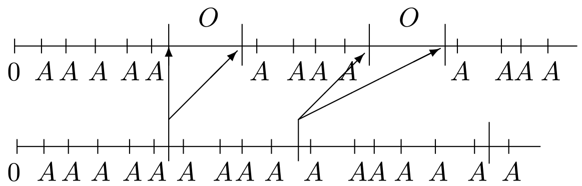
  - Generate B arrivals for each input stream, taking into account the active and inactive periods. (on and off periods)
    - Generate on / off periods in advance.
      - On periods are condensed time line with off periods removed.
      - Off periods are condensed time line with on periods removed.



• Generate arrival instances on the condensed "on" time line using parallel prefix algorithm.



• Expand the on time line to become the whole time line.



• Merge arrivals
  • Want to merge the first B cells, w.r.t. time, from the N input sources.
  • Use a balanced merge method where each process handles E cells.
  • Process i will merge from cell iE to (i+1)E -1.
  • Therefore if E = B/P, then the first B cells will be merged over the P processes.
  • All remaining cells are left for the next iteration of the program.

- Calculate cell losses
  - Mark cells that are clearly accepted or lost.
  - Mark all other cells as unsure & calculate the number of cells that need to be lost in order for their acceptance.
  - Iterate over unsure cells to determine their acceptance.

- Calculate departures
  - Use parallel prefix method to compute departure times of accepted cells.

- Replace lost cells
  - Use a serial version of the algorithm to top the iteration back up to B cells if any cells have been lost in this iteration.

3: Parallel Prefix Algorithm

- Given n objects

  $a_1, a_2, \ldots\ a_n$

- And an associative operation $\oplus$.

- Need to compute the prefix sums

  $a_1\ ,\ a_1 \oplus a_2\ ,\ a_1 \oplus a_2 \oplus a_3\ ,\ a_1 \oplus a_2 \oplus a_3 \oplus a_4\ ,\ \ldots\ ,\ a_1 \oplus a_2 \oplus \ldots \oplus a_n$

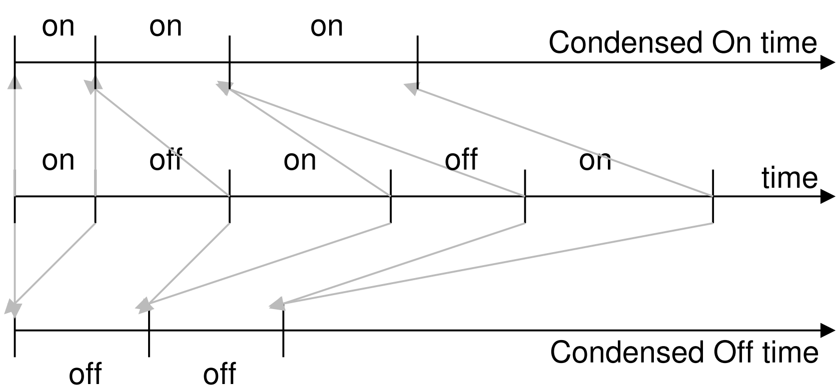
- This can be efficiently solved using the parallel prefix algorithm.
  - Time requirement approximately of the order O(n/P).
  - Thus almost linear speed-up.

- Eg for computing the arrivals.
  - $U_{n+1} = U_n + a_{n+1}$.
  - This is a prefix sum with associative operation of normal addition.

4: Generate arrivals

- Arrival assumptions
  - Any distribution can be simulated.
  - But with bursty "on" / "off" property.

- Generate on / off periods in advance.
  - On periods are condensed time line with off periods removed.
  - Off periods are condensed time line with on periods removed.



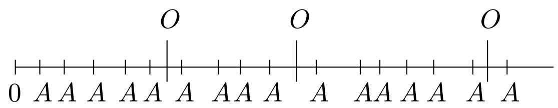
- Can use parallel prefix method to calculate these times.

$Q_{i,j+1} = Q_{i,j} + q_{i,j+1}$ (on times)

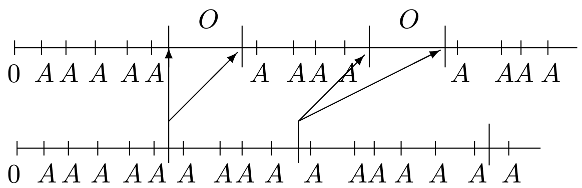
$R_{i,j+1} = R_{i,j} + r_{i,j+1}$ (off times)

- Generate the next B arrivals.
  - Generate arrival instances on the condensed on time line using parallel prefix algorithm.

$$U_{i,n+1} = U_{i,n} + a_{i,n+1}$$



- Expand the on time line to become the whole time line.



  - Achieved by solving the relation

$$\sum_{j=1}^{k-1} q_{i,j} < U_{i,n} \leq \sum_{j=1}^{k} q_{i,j} \ ,$$

  for each cell in the block.
  - And then adding k-1 off periods to the condensed on time.

$$A_{i,n} = U_{i,n} + \sum_{j=1}^{k-1} r_{i,j}$$

  - This expansion can be done in parallel as each cell is independent.

- Calculating number of on / off periods to compute.
  - Can be predicted from theory. [see paper]
  - Values doubled for use in program.
  - Results within 5% of theoretical values.

5: Merge arrivals

- Want to merge the first B cells, w.r.t. time, from the N input sources.
  - Use a balanced merge method where each process handles E cells.
  - Process i will merge from cell iE to (i+1)E -1.
  - Therefore if E = B/P, then the first B cells will be merged over the P processes.

- Balanced partitions are achieved by iterating the following steps for each boundary.
  - Take an estimate e for the value that will give iE cells before it.
  - Use a binary search algorithm to compute the number of cells in each stream less than e and sum to give total number less than e.
  - Use this total to reassess the estimate for the partitioning value.
    - too many ➡ reduce estimate.
    - too few ➡ increase estimate.

  - With good choice of estimates it is possible to half the search space for the correct value of e at each iteration.

- Once boundary's are found.
  - Each process can merge its cells independently of the others.
  - Thus the merges can be performed in parallel.
  - Each process uses a standard serial merging algorithm.

6: Calculating loss and accepts

- A record of the departure times for the previous B cells needs to be kept for this stage labelling them from 1-B, 2-B, to 0.
- The current B cells will be labelled as 1, 2, to B.

- Ascertain the states of each cell:

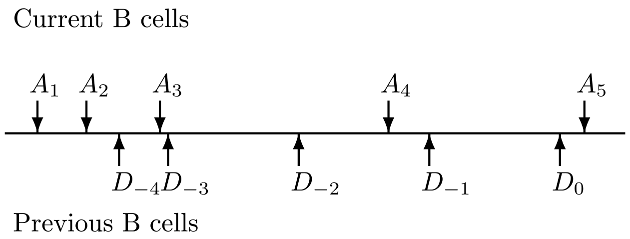
- Cell i is accepted if
  - The cell i-B (from departure list) has departed when cell i arrives.

- Cell i is lost if
  - The cell i-B has not departed when cell i arrives and
  - All cells, before i-B, from the departure block have departure times after cell i arrives.

- Cell i is unsure if
  - The cell i-B has not departed when cell i arrives and
  - There is a cell i-B-j in the departure list that departs before cell i arrives.

  - Thus if at least j cells are lost from the current arrival block before cell i then cell i can be accepted.

Eg. for case of buffer size 5.



- Cells 1 and 2 will be lost.
- Cell 3 is unsure, but will be accepted if 2 cells are lost before it.
- Cell 4 is unsure, but will be accepted if 1 cell is lost before it.
- Cell 5 is accepted.

- Each process is thus allocated B/P cells to ascertain their state as above.
    - These can be performed independently.
    - And thus in parallel.

- Cell acceptance for the remaining unsure cells. Iterate the following stages:

  - Each process calculates the number of cells lost in it's block.
  - Each process calculates the maximum number of cells that can be lost in it's block (assumes all unsure cells are lost).

  - Each process can now compute a maximum and minimum for the number of cells lost in the buffer before it's own block.
  - If for an unsure cell, $j <$ minimum, then the cell can now be accepted. and marked as such.
  - If for an unsure cell, $j >$ maximum, then the cell is now lost and marked as such.

  - This will be repeated until all processes have maximum = minimum.
  - Will take no more than P iterations.
  - If cell loss is low will only take a few iterations.
  - Each process can work in parallel.

7: Departure times

- Computation of departure times for remaining cells is performed using the parallel prefix method using the matrix product in the (max, +) algebra given by Greenberg et all (1991) as the associative operation.

- Serial version of the above algorithm used to top the buffer back up to B cells.

- Assumed cell loss is low enough that parallelising the top up would be more time consuming.

8: Conclusion and results

- Speed-up obtained from the algorithm is almost linear O(M/P).
- Even in cases where the losses are 1% of sent cells.
- Can be used to model many arrival stream properties.
- Confidence intervals for timings below are good - within 1 second.
- Some results:

Table 1: Simulation times for a 6 input ATM switch
Buffer size B, no of processors P
s = sequential version L = average cell loss

B	500	1000	5000	10000	50000
P					
s	151.01	150.17	151.73	151.16	152.63
1	193.36	189.86	194.53	202.58	228.76
2	131.55	125.72	122.52	124.14	138.51
4	84.13	75.14	66.53	65.24	73.25
6	66.58	57.86	49.58	46.86	51.65
8	56.99	46.46	37.55	35.51	39.21
10	52.82	41.03	31.52	29.60	33.20
12	51.37	40.91	29.05	27.49	30.03
14	67.10	54.23	32.08	30.04	31.76
L	10000	7600	2100	1100	100