Global Snapshots

K. Rustan M. Leino

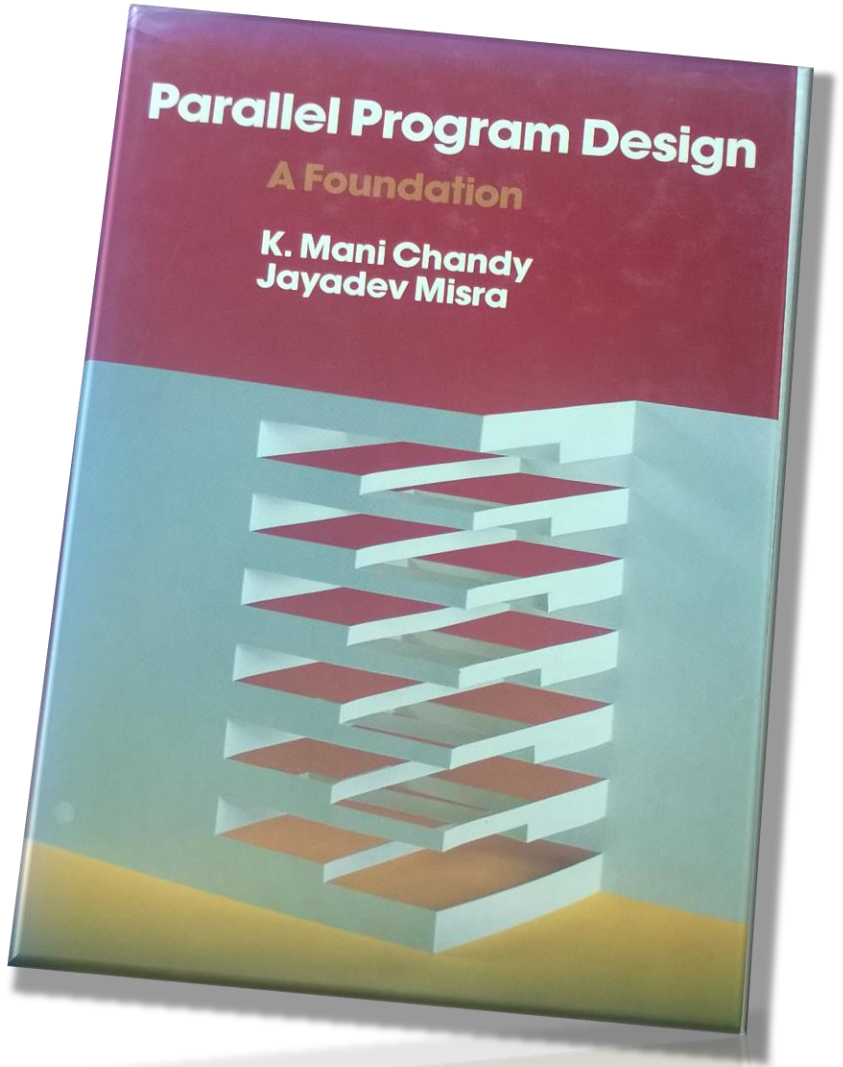
Principal Researcher
Research in Software Engineering (RiSE), Microsoft Research, Redmond

Visiting Professor
Department of Computing, Imperial College London

Dr. TLA+ Series, Microsoft Research, Redmond, WA, USA , 23 September 2016

Credits

A useful source: Chapter 10 of *Parallel Program Design: A Foundation*, by K. Mani Chandy and Jayadev Misra (Addison Wesley, 1988)



Recording the state of a computation

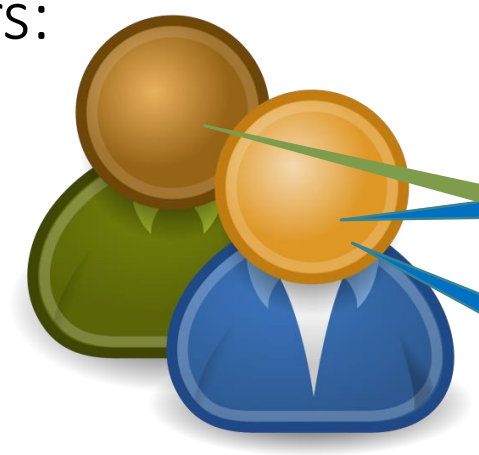
Taking a snapshot

Useful for

- Checkpointing
- Detecting when a condition, known to be stable, has been reached

Example: Bank

Customers:



Requests:

Transfer $40 from A to C

Transfer $20 from B to C

Transfer $70 from C to A

Conceptual history:

A	B	C
$100	$100	$100
$60	$100	$140
$60	$80	$160
$130	$80	$90

Accounts:

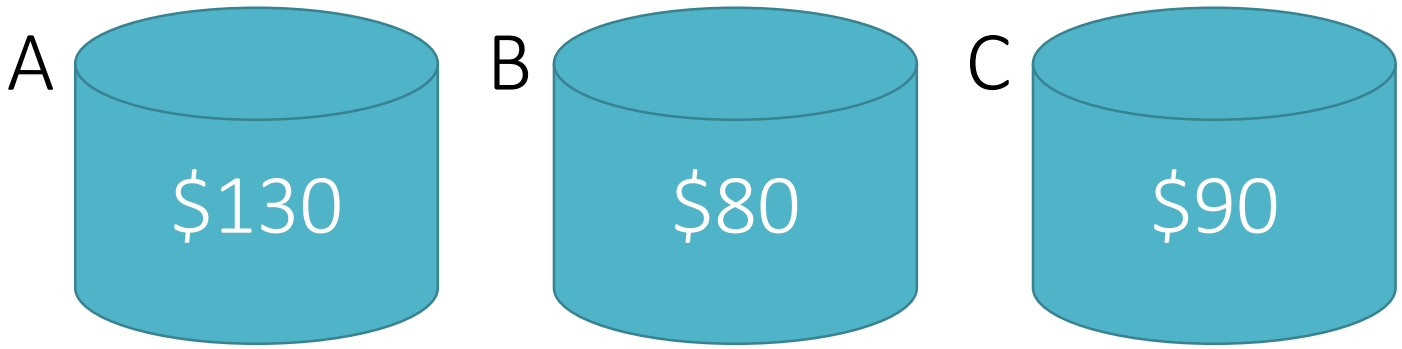
A B C

Transfer $40 from A to C

Transfer $20 from B to C

Transfer $70 from C to A

A	B	C
$100	$100	$100
$60	$100	$140
$60	$80	$160
$130	$80	$90



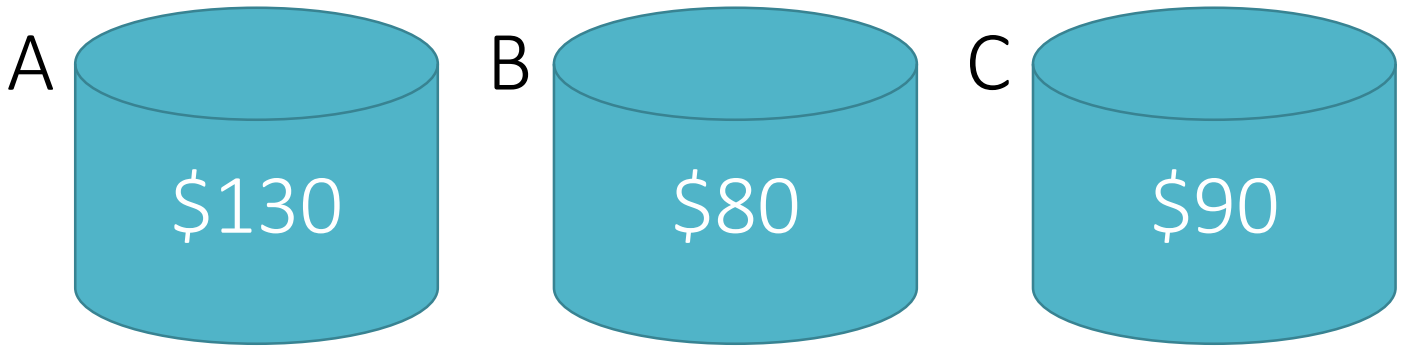
Snapshot:

Transfer $40 from A to C

Transfer $20 from B to C

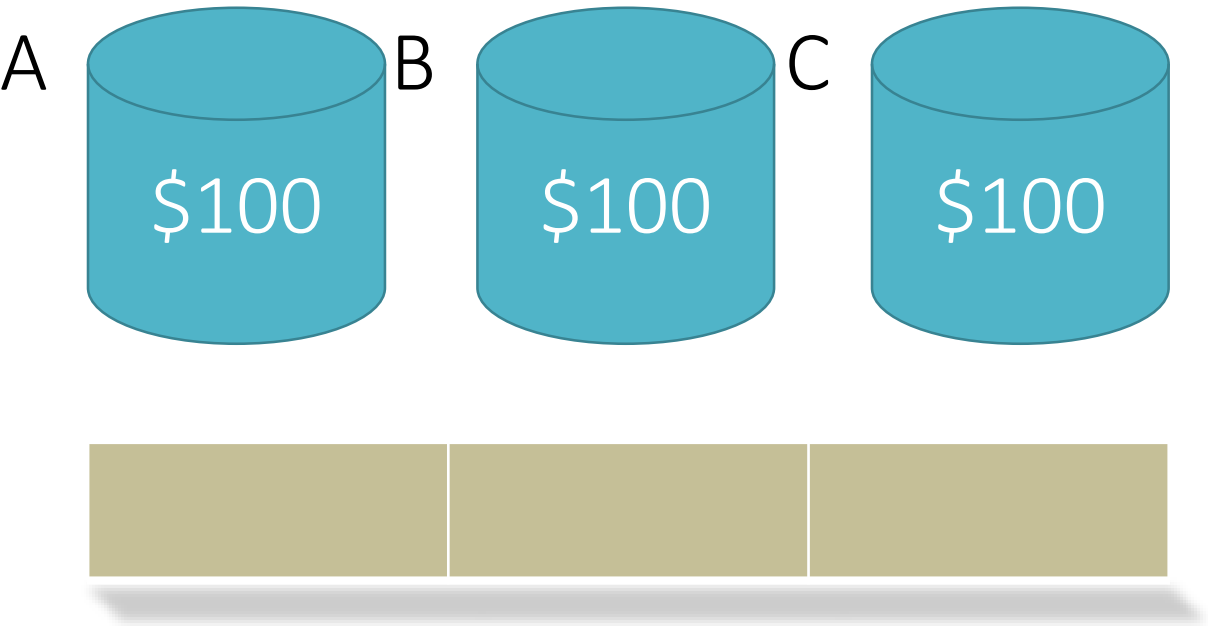
Transfer $70 from C to A

A	B	C
$100	$100	$100
$60	$100	$140
$60	$80	$160
$130	$80	$90



Snapshot:

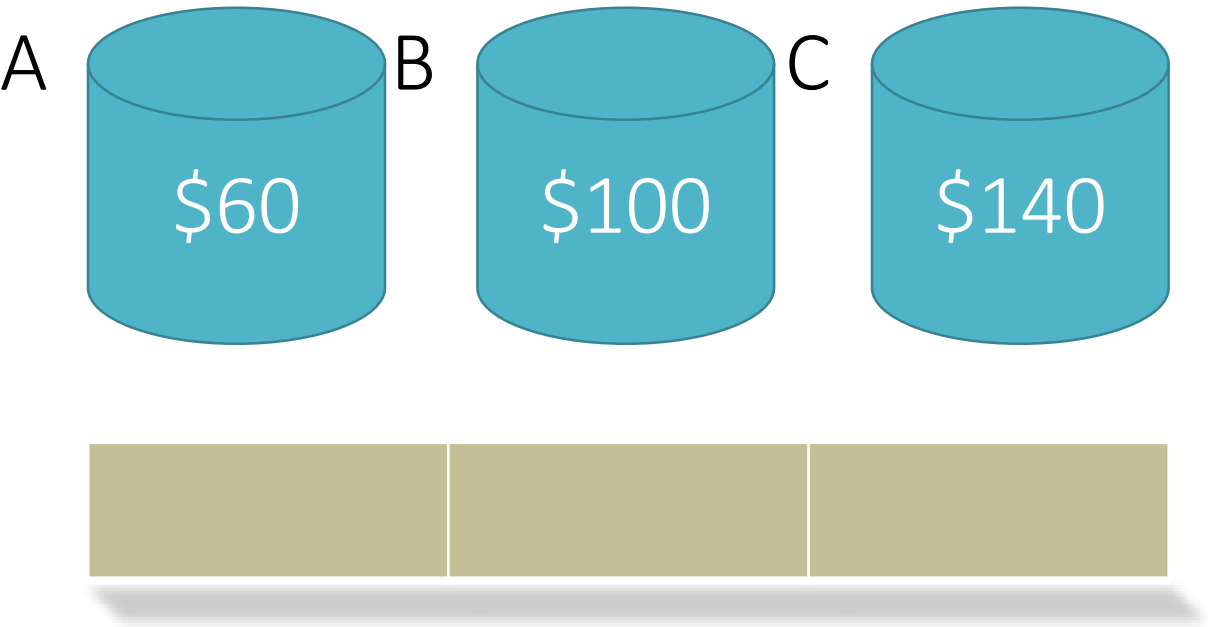
Example computation



A	B	C
$100	$100	$100

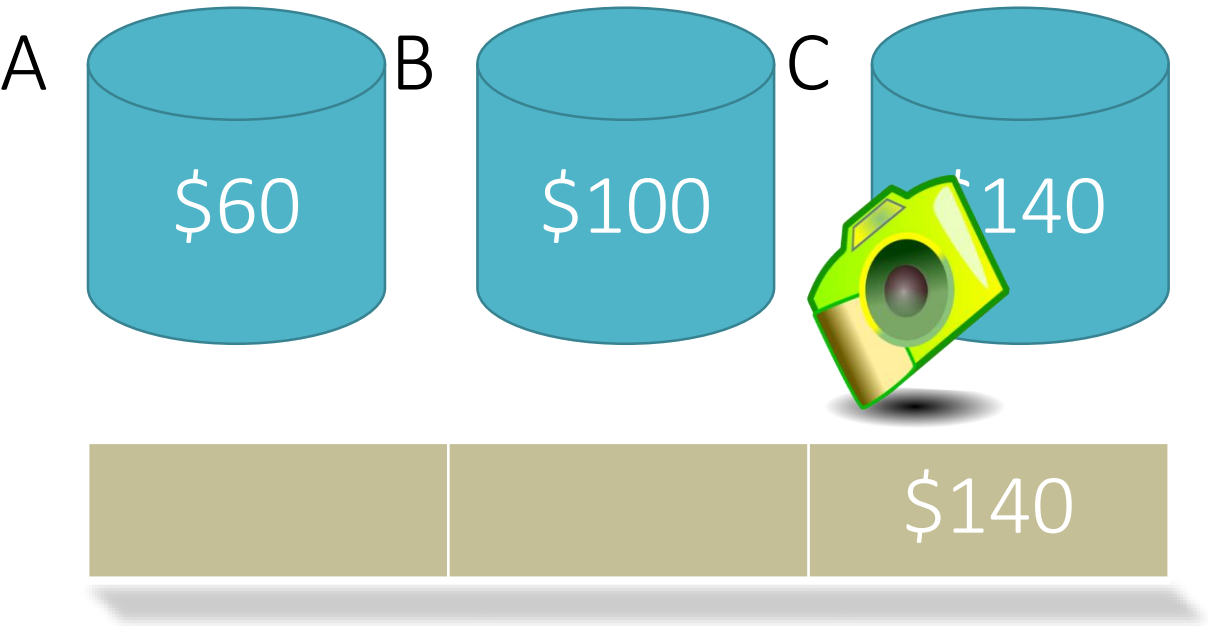
A	B	C
$100	$100	$100
$60	$100	$140

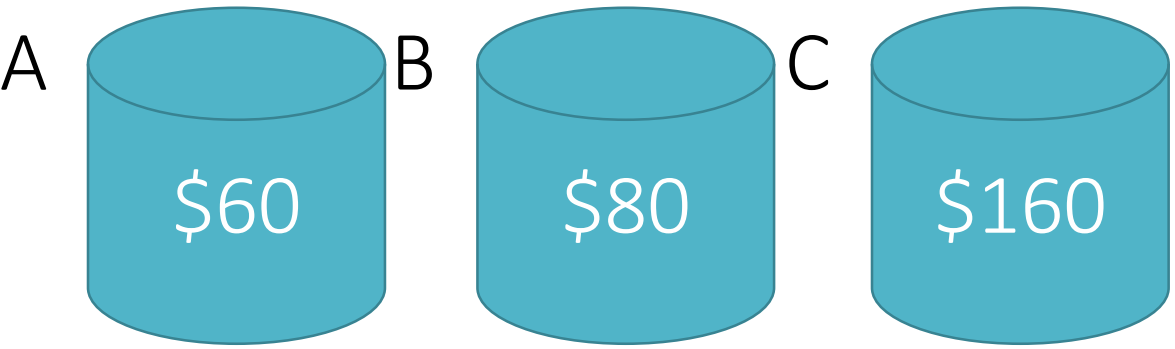
Transfer $40 from A to C



A	B	C
$100	$100	$100
$60	$100	$140

Transfer $40 from A to C

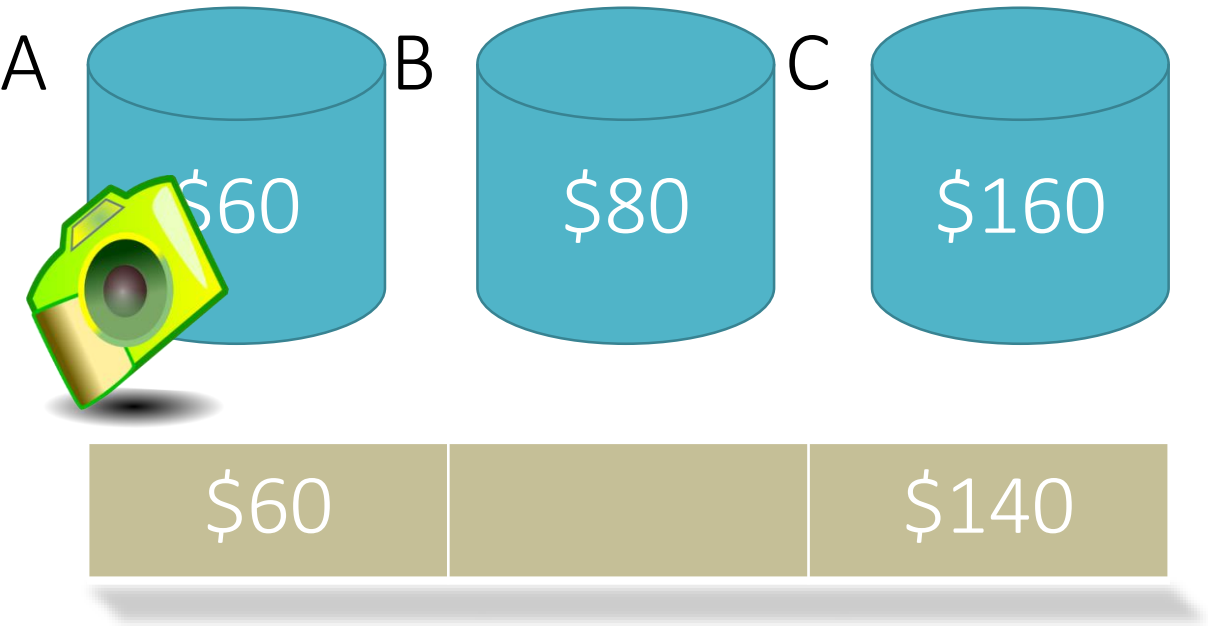




Transfer $40 from A to C

Transfer $20 from B to C

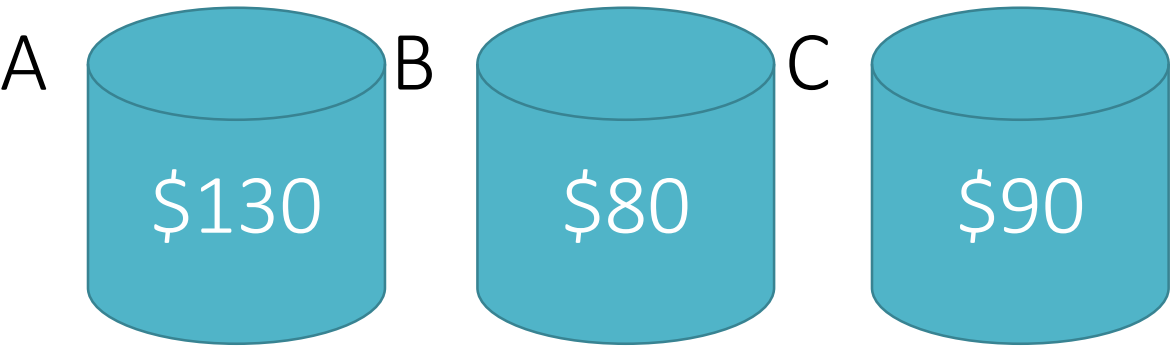
A	B	C
$100	$100	$100
$60	$100	$140
$60	$80	$160



A	B	C
$100	$100	$100
$60	$100	$140
$60	$80	$160

Transfer $40 from A to C

Transfer $20 from B to C

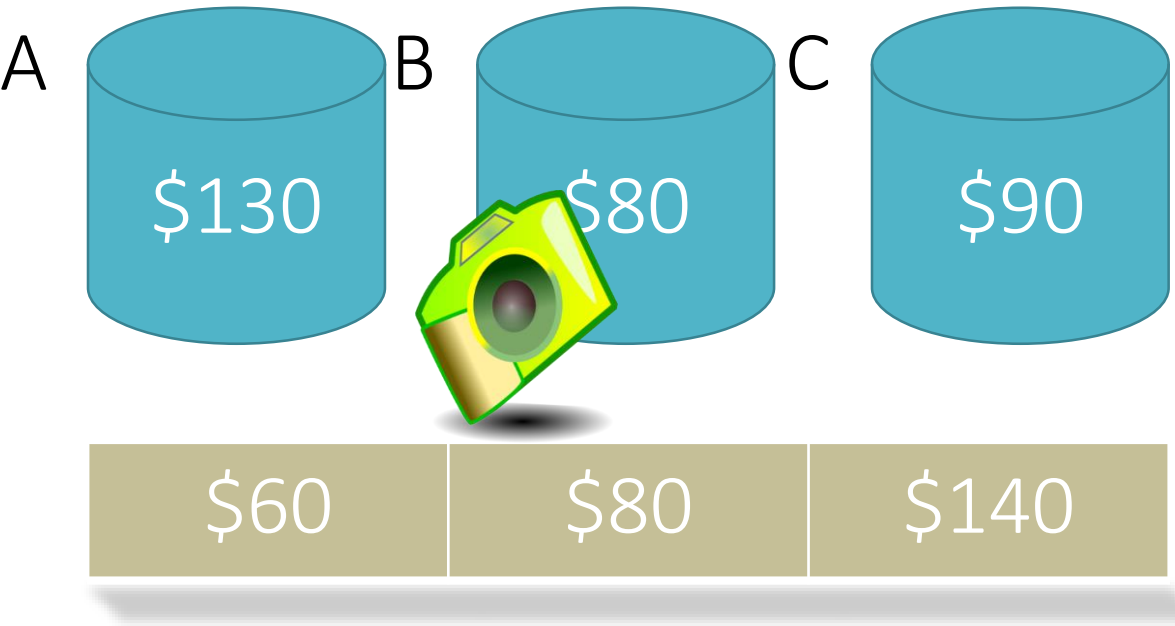


A	B	C
$100	$100	$100
$60	$100	$140
$60	$80	$160
$130	$80	$90

Transfer $40 from A to C

Transfer $20 from B to C

Transfer $70 from C to A



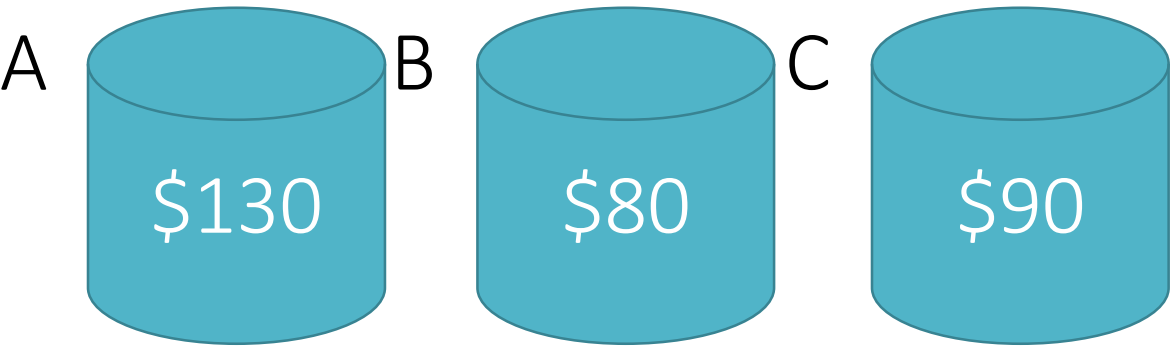
A	B	C
$100	$100	$100
$60	$100	$140
$60	$80	$160
$130	$80	$90

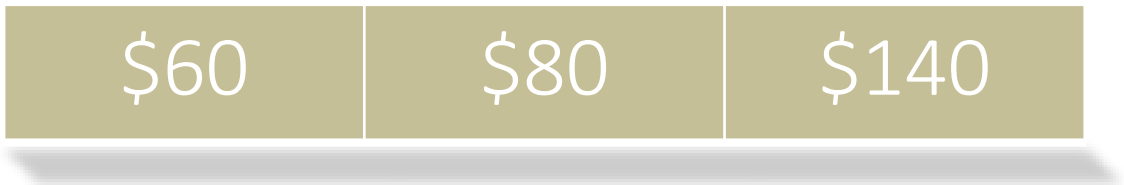
Transfer $40 from A to C

Transfer $20 from B to C

Transfer $70 from C to A

This snapshot seems bad.
Recording a snapshot requires some care.





	A	B	C
	$100	$100	$100
Transfer $40 from A to C	$60	$100	$140
Transfer $20 from B to C	$60	$80	$160
Transfer $70 from C to A	$130	$80	$90

The snapshot we get may have never occurred in the computation

$130	$100	$70	$100

A	B	C	D
$100	$100	$100	$100
$60	$100	$140	$100
$60	$80	$140	$120
$130	$80	$70	$120
$130	$90	$70	$110

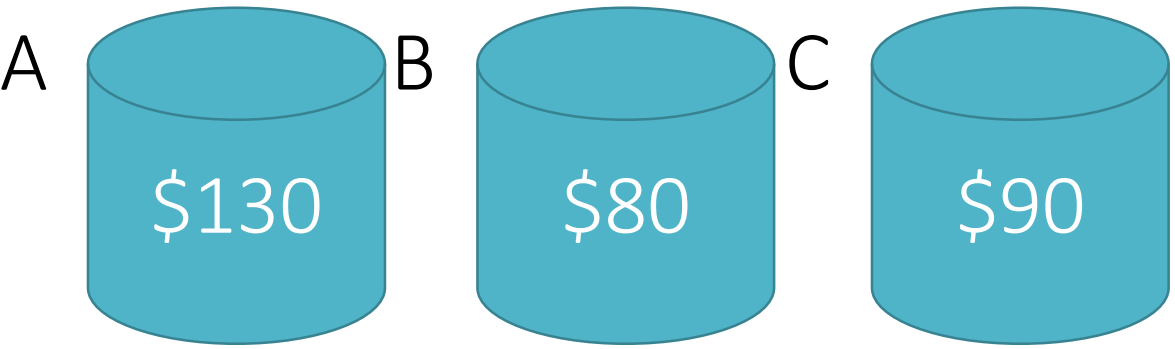
Transfer $40 from A to C

Transfer $20 from B to D

Transfer $70 from C to A

Transfer $10 from D to B

Would we be happy if we got this snapshot?



$130	$100	$70

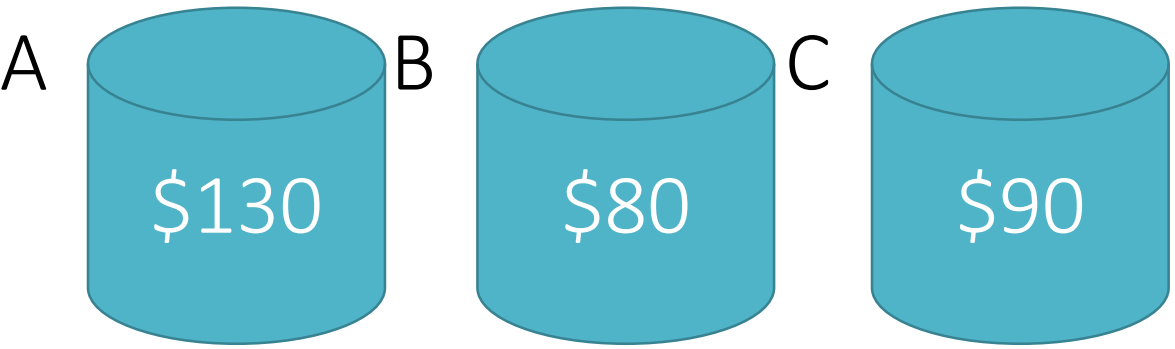
Transfer $40 from A to C

Transfer $20 from B to C

Transfer $70 from C to A

A	B	C
$100	$100	$100
$60	$100	$140
$60	$80	$160
$130	$80	$90

Would we be happy if we got this snapshot?



$130	$100	$70

Transfer $40 from A to C

Transfer $70 from C to A

Transfer $20 from B to C

A	B	C
$100	$100	$100
$60	$100	$140
$130	$100	$70
$130	$80	$90

Specification

For any computation with program events E:

there exist program events U, V $\subseteq$ E, such that

Specification: Termination

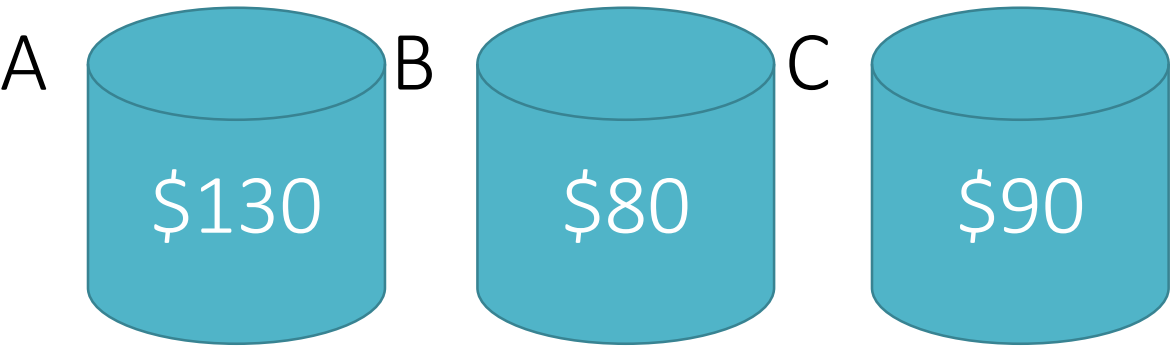
Eventually all variables have been recorded

Rule R

When an event in the underlying computation is executed, either

- All variables mentioned in the event have been recorded, or
- All variables mentioned in the event have not yet been recorded

Does this recording satisfy Rule R?



$60	$80	$140

Transfer $40 from A to C

Transfer $20 from B to C

Transfer $70 from C to A

A	B	C
$100	$100	$100
$60	$100	$140
$60	$80	$160
$130	$80	$90

Does this recording satisfy Rule R?

$130	$100	$70	$100

A	B	C	D
$100	$100	$100	$100
$60	$100	$140	$100
$60	$80	$140	$120
$130	$80	$70	$120
$130	$90	$70	$110

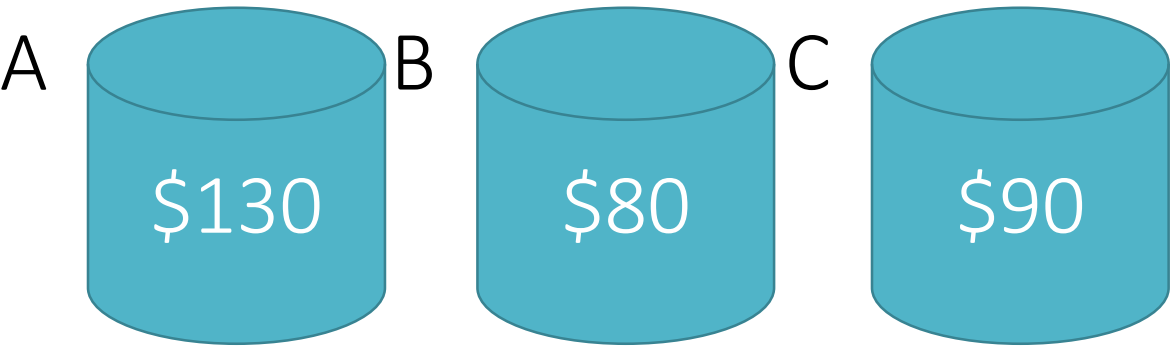
Transfer $40 from A to C

Transfer $20 from B to D

Transfer $70 from C to A

Transfer $10 from D to B

Does this recording satisfy Rule R?



$130	$100	$70

Transfer $40 from A to C

Transfer $20 from B to C

Transfer $70 from C to A

A	B	C
$100	$100	$100
$60	$100	$140
$60	$80	$160
$130	$80	$90

Proving correctness

Define

```
X_partial  =  if X_done then X_recorded else X
```

Invariant:

There exist U, V $\subseteq$ E such that

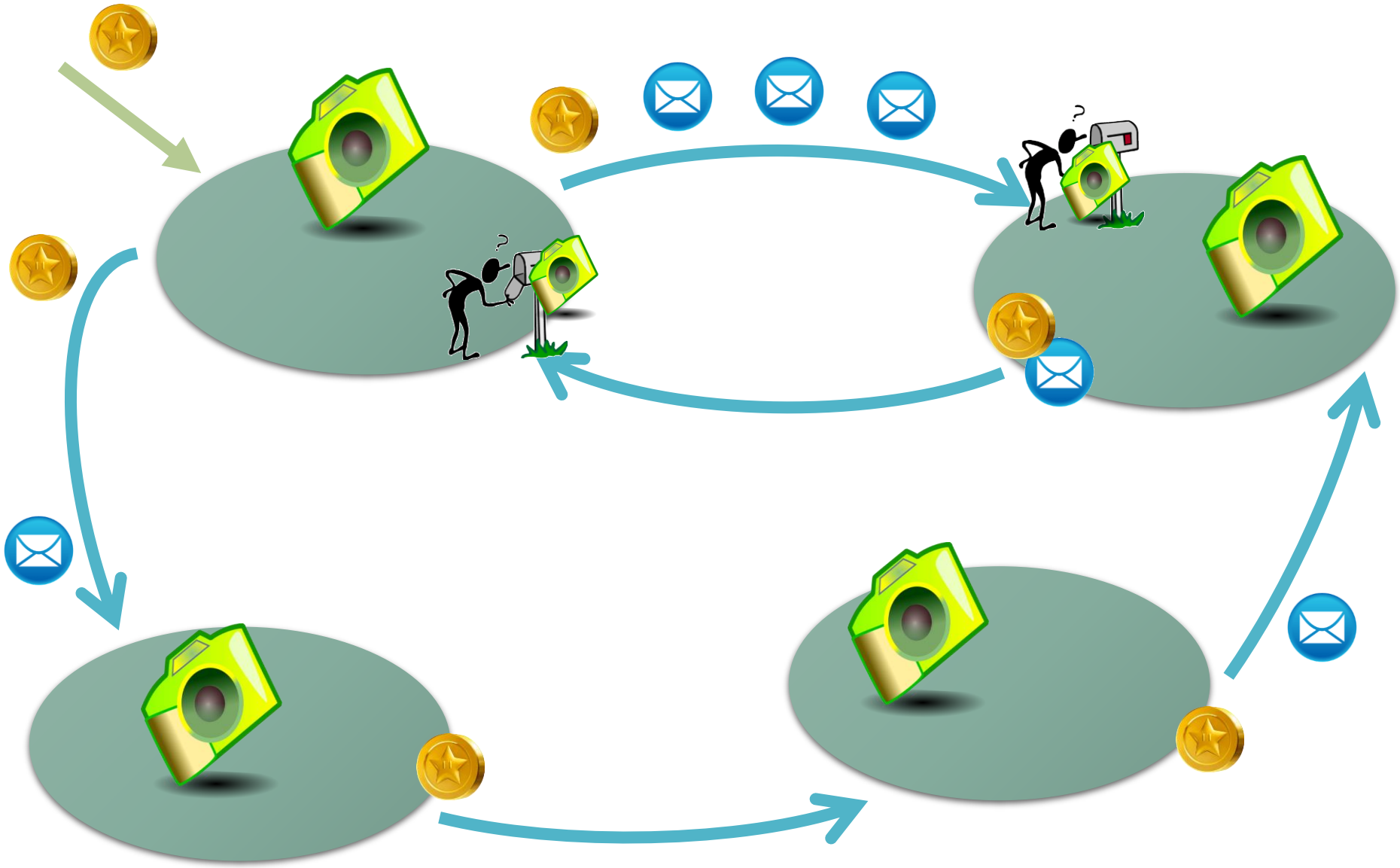
V only mentions variables that have been recorded, and

Programs with channels

State of a channel cannot be read directly

State of a channel can be computed as a difference between
- The sender's record of what has been sent
- The receiver's record of what has been received

Chandy & Lamport algorithm achieves this more efficiently with markers



Conclusions

Global Snapshots let you take piecewise checkpoints of processes are combine them into a consistent view

The state of channels can be computed from what has been sent and received along the channel

Rule R is a basis for verifying many Global Snapshots algorithms

Plugs:

- Verification Corner channel on youtube
- Jean-Raymond Abrial's Event-B lecture series (2011) on resnet