

Cloud in images

Cloud	statusc	idlec
	valc	n

pc-1	statusp	idlep
	valp	k
	tmp	0

pc-2	statusp	idlep
	valp	l
	tmp	0

...

pc-n	statusp	idlep
	valp	m
	tmp	0

Cloud in images

Cloud	statusc	idlec
	valc	n

transition: modval

k replaced by *new* if pc-1 is *idlep*

new is a (arbitrary) new value

pc-1	statusp	<i>idlep</i>
	valp	<i>new</i>
	tmp	0

pc-2	statusp	idlep
	valp	l
	tmp	0

...

pc- n	statusp	idlep
	valp	m
	tmp	0

Cloud in images

Cloud	statusc	busy
	valc	n

transition: gotvalue

pc-1	statusp	gotvalue
	valp	new
	tmp	n

pc-2	statusp	idlep
	valp	/
	tmp	0

...

pc-n	statusp	idlep
	valp	m
	tmp	0

Cloud in images

Cloud	statusc	busy
	valc	<i>new</i>

transition: update assuming $new \geq n$

pc-1	statusp	update
	valp	<i>new</i>
	tmp	<i>new</i>

pc-2	statusp	idlep
	valp	/
	tmp	0

...

pc-n	statusp	idlep
	valp	<i>m</i>
	tmp	0

Cloud in images

Cloud	statusc	idlec
	valc	new

transition: gotoidle

pc-1	statusp	idlep
	valp	new
	tmp	0

pc-2	statusp	idlep
	valp	/
	tmp	0

...

pc-n	statusp	idlep
	valp	m
	tmp	0

Transitions

- ▶ GetValue: if PC and Cloud is idle, fetch Cloud value
- ▶ Update: update Cloud/PC according to larger value
- ▶ Gotoidle: both PC and Cloud go back to idle

We need to define the value of each observer after applying each of the transitions above.

GOAL

If PC is in updated state, then the values of Cloud and PC agree