CLOUD COMPUTING
CONCEPTS

with Indranil Gupta (Indy)

SNAPSHOTS

Lecture A

WHAT IS GLOBAL SNAPSHOT

HERE'S A SNAPSHOT

Wikimedia commons

Sommet de Paris
pour le soutien au peuple libyen
Samedi 19 mars 2011

Distributed Snapshot

- More often, each country's representative is sitting in their respective capital and sending messages to each other (say emails).

- How do you calculate a "global snapshot" in that distributed system?

- What does a "global snapshot" even mean?

IN THE CLOUD

- **In a cloud: each application or service is running on multiple servers**
- **Servers handling concurrent events and interacting with each other**
- **The ability to obtain a "global photograph" of the system is important**
- **Some uses of having a global picture of the system**
  - *Checkpointing*: can restart distributed application on failure
  - *Garbage collection* of objects: objects at servers that don't have any other objects (at any servers) with pointers to them
  - Deadlock detection: Useful in database transaction systems
  - Termination of computation: Useful in batch computing systems like Folding@Home, SETI@Home
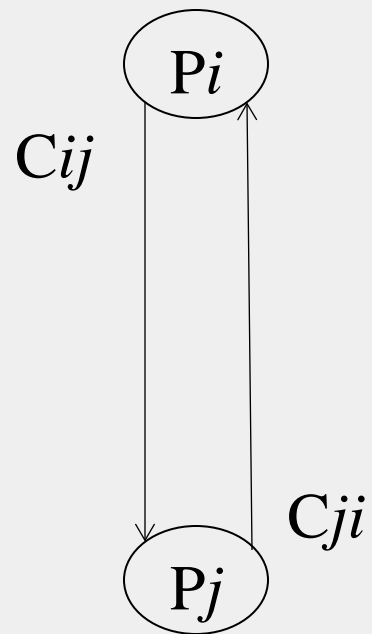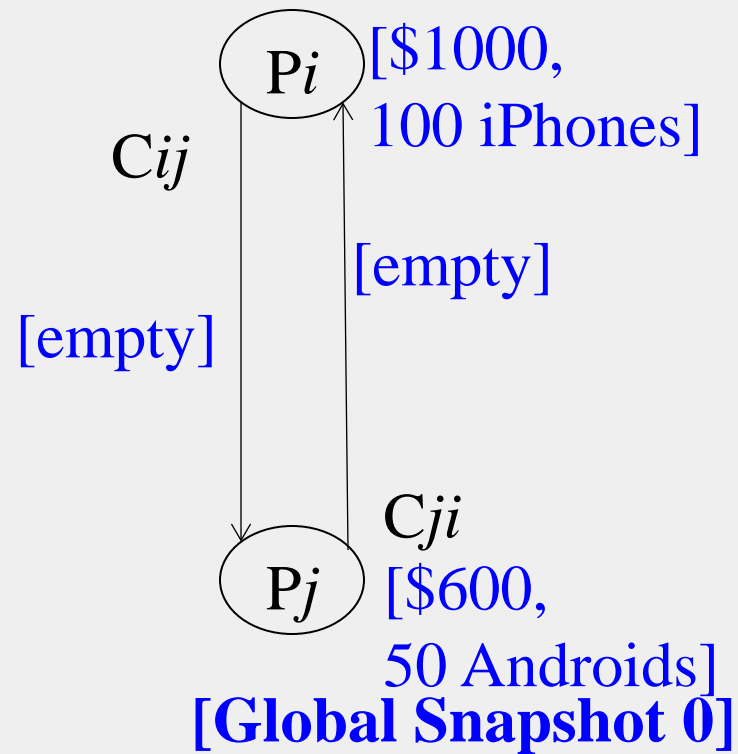
WHAT'S A GLOBAL SNAPSHOT?

- **Global Snapshot = Global State =**
  Individual state of each process in the distributed system
  +
  Individual state of each communication channel in the distributed system

- Capture the instantaneous *state* of each process

- And the instantaneous *state* of each communication channel, i.e., *messages* in transit on the channels
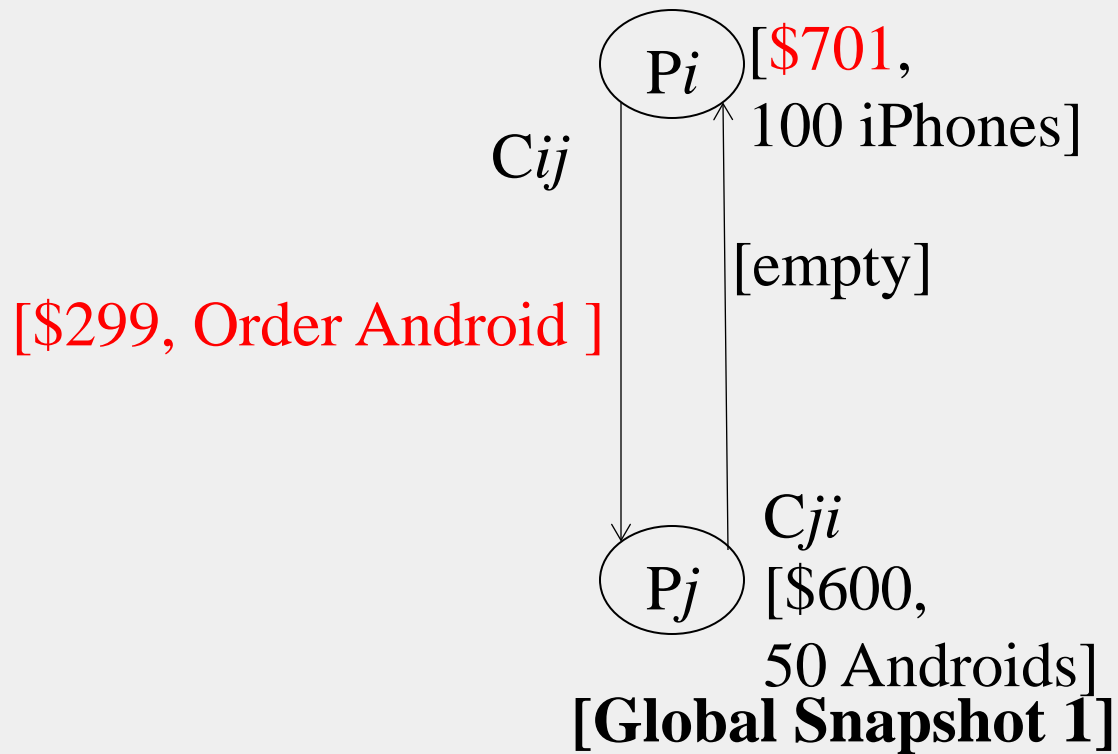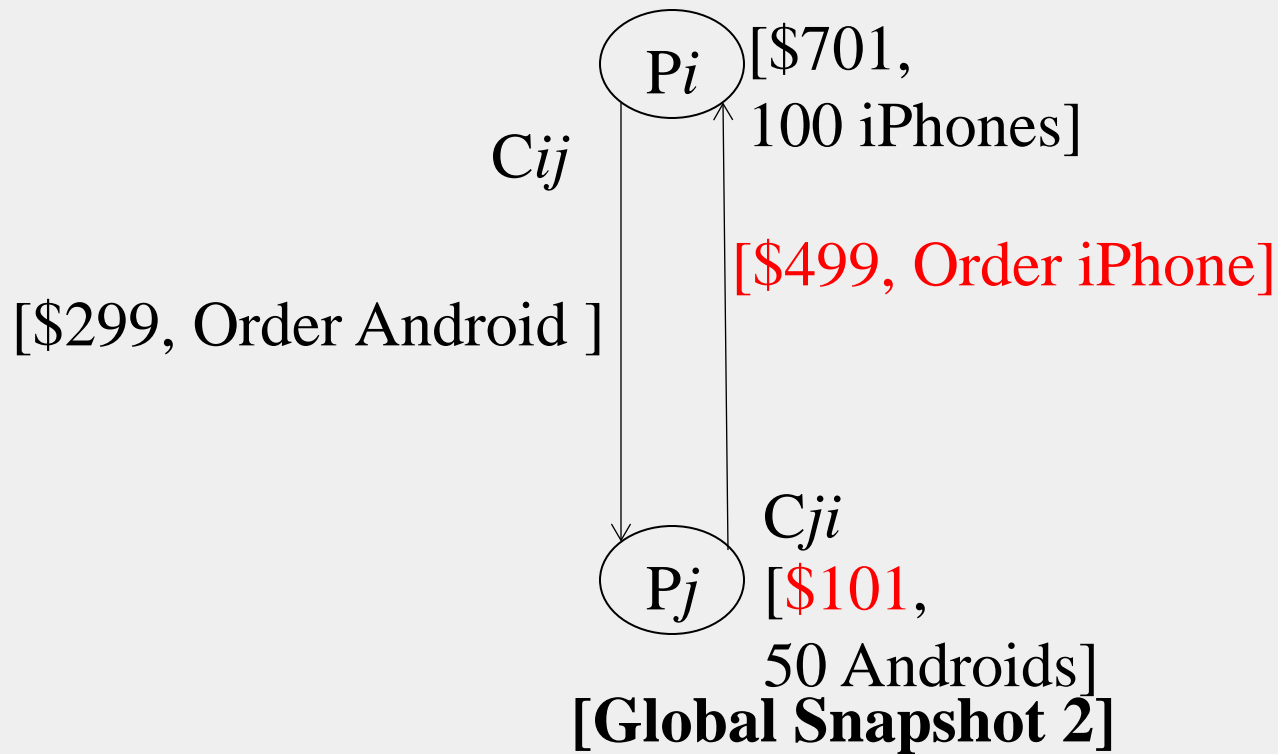
Obvious First Solution

- Synchronize clocks of all processes
- Ask all processes to record their states at known time t
- Problems?
  - Time synchronization always has error
    - Your bank might inform you, "We lost the state of our distributed cluster due to a 1 ms clock skew in our snapshot algorithm."
  - Also, does not record the state of messages in the channels

- Again: synchronization not required – causality is enough!
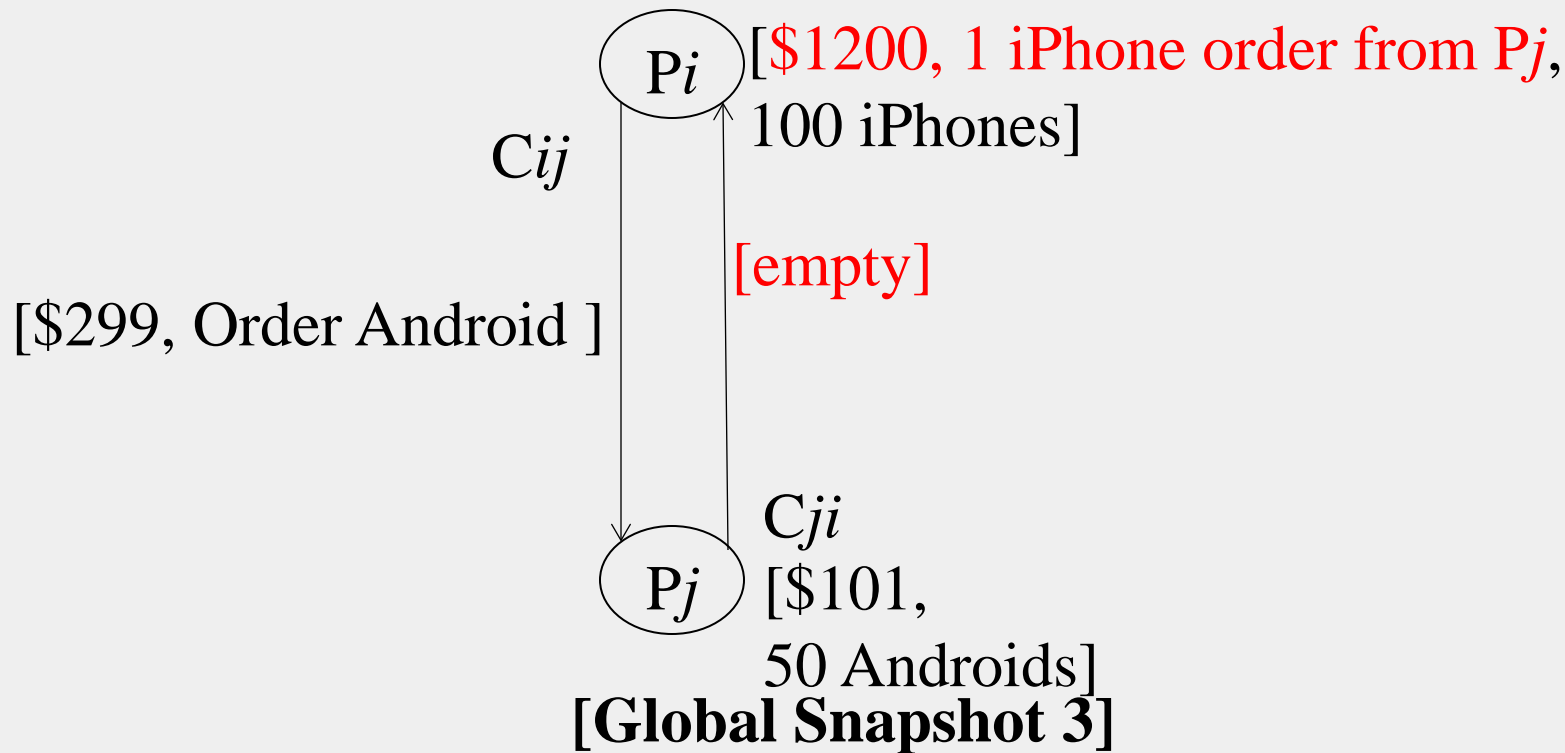
EXAMPLE

P*i* [$1000,
100 iPhones]

C*ij*

[empty]

[empty]

C*ji*
P*j* [$600,
50 Androids]
[Global Snapshot 0]

P*i* [$701,
100 iPhones]

C*ij*

[empty]

[$299, Order Android]

C*ji*
[$600,

P*j*

50 Androids]
[Global Snapshot 1]

Pi [$701, 100 iPhones]

Cij

[$499, Order iPhone]

[$299, Order Android]

Cji

Pj [$101, 50 Androids]

[Global Snapshot 2]

Pi [$1200, 1 iPhone order from Pj, 100 iPhones]

Cij

[empty]

[$299, Order Android]

Cji

Pj [$101, 50 Androids]

[Global Snapshot 3]

Pi [$1200, 99 iPhones]

Cij

[empty]

[
($299, Order Android),
(1 iPhone)
]

Cji
Pj [$101, 50 Androids]

[Global Snapshot 4]

Pi [$1200, 99 iPhones]

Cij

[empty]

Cji

Pj [$400, 1 Android order from Pi, 50 Androids]

[
(1 iPhone)
]

[Global Snapshot 5]

Pi [$1200, 99 iPhones]

Cij

[empty]

[empty]

… and so on …

Cji
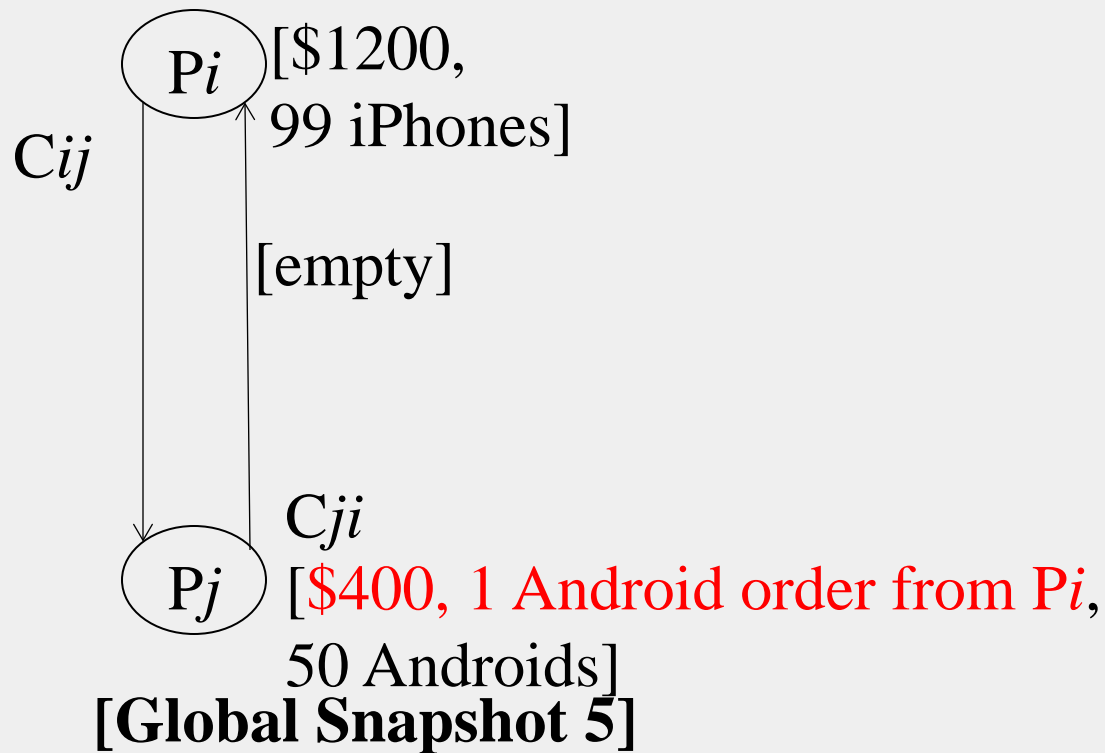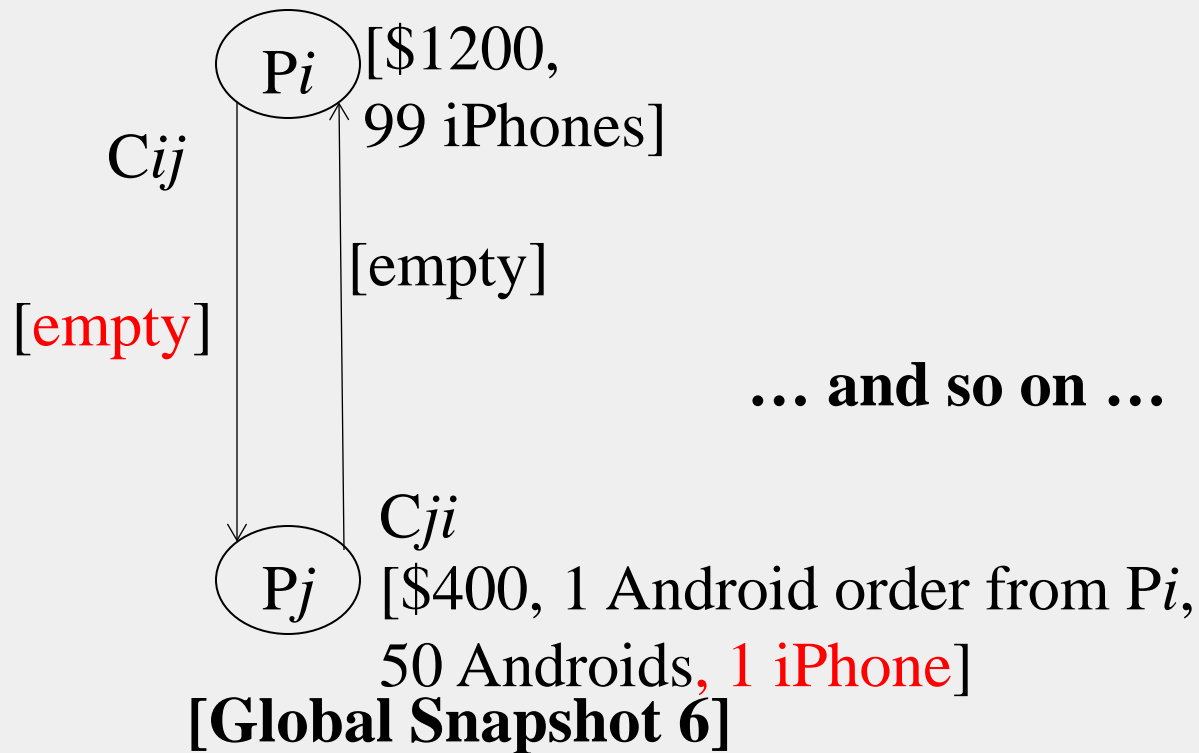Pj [$400, 1 Android order from Pi, 50 Androids, 1 iPhone]

[Global Snapshot 6]

MOVING FROM STATE TO STATE

- **Whenever an event happens anywhere in the system, the global state changes**
  - Process receives message
  - Process sends message
  - Process takes a step
- **State to state movement <u>obeys causality</u>**
  - Next: Causal algorithm for Global Snapshot calculation