

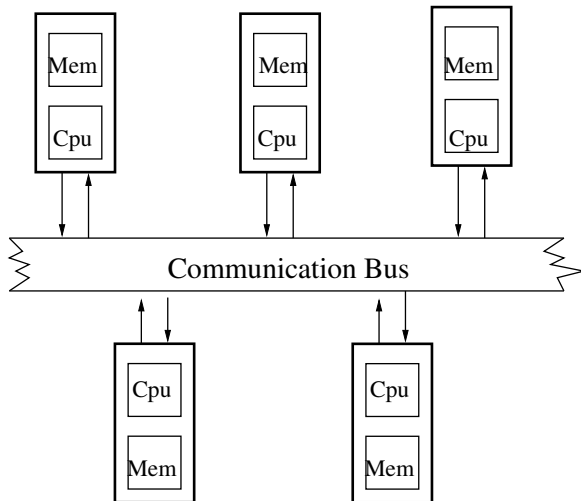
Weryfikacja protokołu zgodności cachu COMA

Krzysztof Nozderko

`kn201076@students.mimuw.edu.pl`

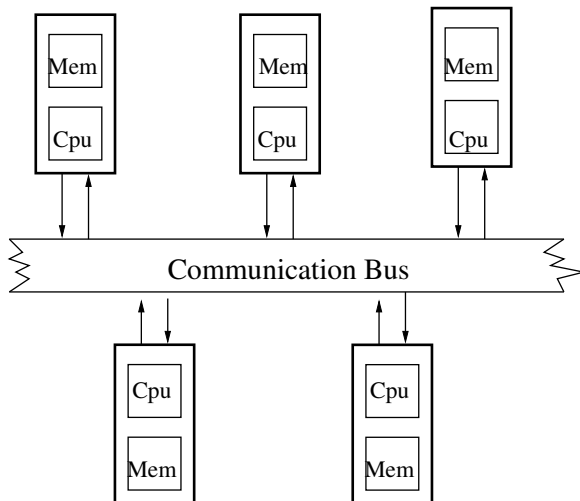
8 maj 2006

Architektura systemu



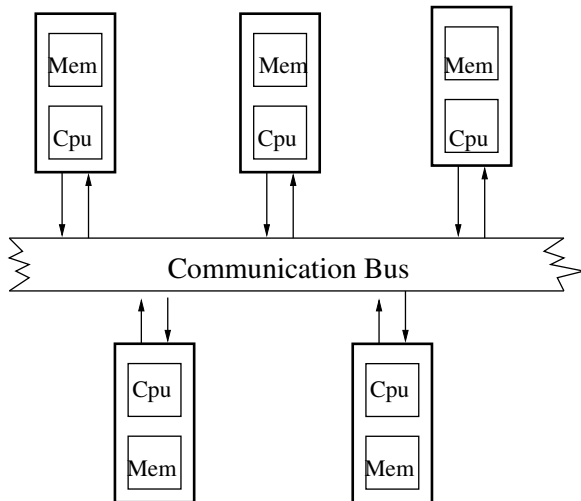
- ▶ Wiele **procesorów**
- ▶ Każdy procesor ma swoją lokalną pamięć (**cache**)
- ▶ Wspólna **szyna** służąca do komunikacji i wymiany danych

Architektura systemu



- ▶ Wiele **procesorów**
- ▶ Każdy procesor ma swoją lokalną pamięć (**cache**)
- ▶ Wspólna **szyna** służąca do komunikacji i wymiany danych

Architektura systemu



- ▶ Wiele **procesorów**
- ▶ Każdy procesor ma swoją lokalną pamięć (**cache**)
- ▶ Wspólna **szyna** służąca do komunikacji i wymiany danych

COMA - Cache Only Memory Architecture

- ▶ To mechanizm zapewniający spójność cachów
- ▶ Chcemy ograniczyć ilość przesyłów

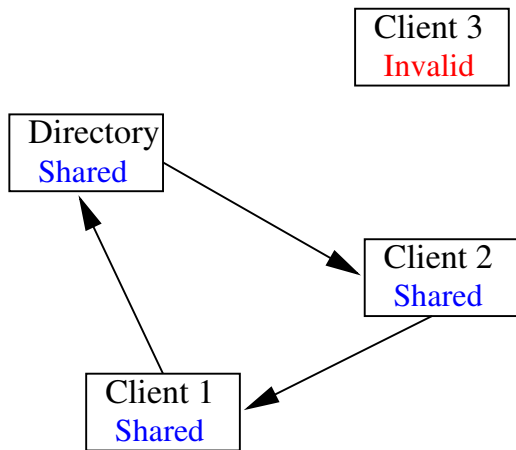
Stany procesora

Każdy z procesorów jest w jednym z 3 stanów:

- ▶ **exclusive** Procesor ma jedyną aktualną kopię danych
- ▶ **shared** Procesor ma aktualną kopię danych, ale jest ktoś inny, kto też ją ma
- ▶ **invalid** Procesor nie ma aktualnej kopii danych

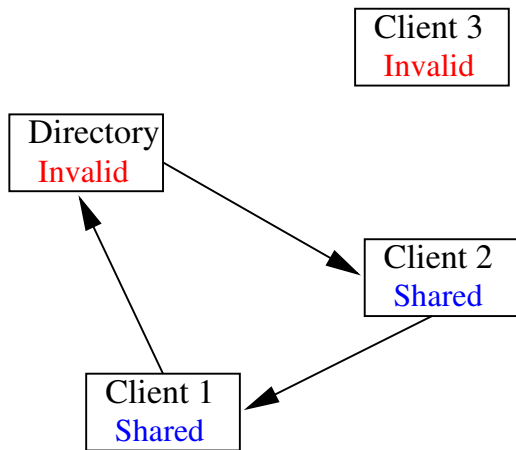
Protokół COMA

- ▶ lista procesorów mających aktualne dane – **validity chain**
- ▶ wyróżniony 1 procesor – węzeł katalogowy (**directory**)



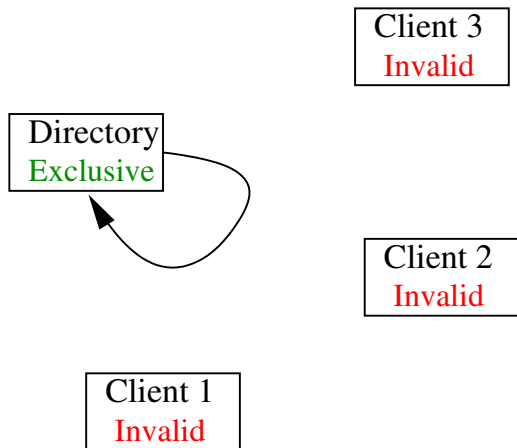
- ▶ **Directory** wskazuje na głowę łańcucha (Client 2)
- ▶ Łańcuch jest listą **cykliczną**
- ▶ Directory może być w stanie **invalid** ...
- ▶ ... lub w stanie **exclusive**

- ▶ lista procesorów mających aktualne dane – **validity chain**
- ▶ wyróżniony 1 procesor – węzeł katalogowy (**directory**)



- ▶ **Directory** wskazuje na głowę łańcucha (Client 2)
- ▶ Łańcuch jest listą **cykliczną**
- ▶ Directory może być w stanie **invalid** ...
- ▶ ... lub w stanie **exclusive**

- ▶ lista procesorów mających aktualne dane – **validity chain**
- ▶ wyróżniony 1 procesor – węzeł katalogowy (**directory**)



- ▶ **Directory** wskazuje na głowę łańcucha (Client 2)
- ▶ Łańcuch jest listą **cykliczną**
- ▶ Directory może być w stanie **invalid** ...
- ▶ ... lub w stanie **exclusive**

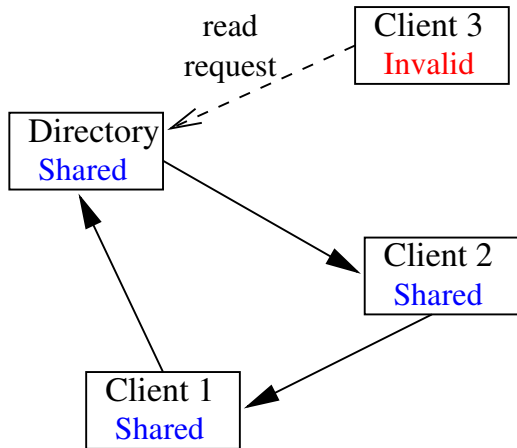
Proces, który chce **czytać dane**:

- ▶ Jeśli ma aktualne dane (stan **shared** lub **exclusive**)
⇒ **czyta** je
- ▶ Jeśli ma nieaktualne dane (stan **invalid**)
⇒ wysyła żądanie do **Directory** (read request)

Protokół COMA - odpowiedź Directory na read request

Przypadek 1

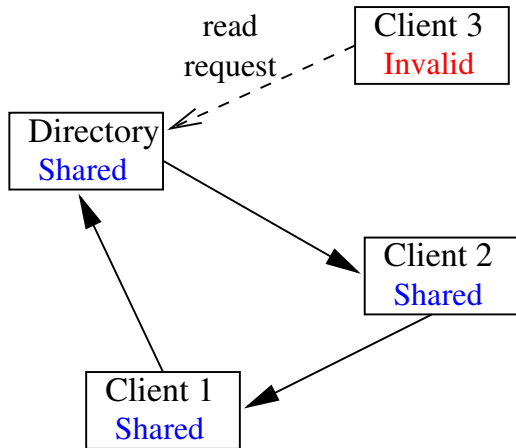
Directory ma aktualne dane (jest w stanie **shared** lub **exclusive**).



Protokół COMA - odpowiedź Directory na read request

Przypadek 1

Directory ma aktualne dane (jest w stanie **shared** lub **exclusive**).

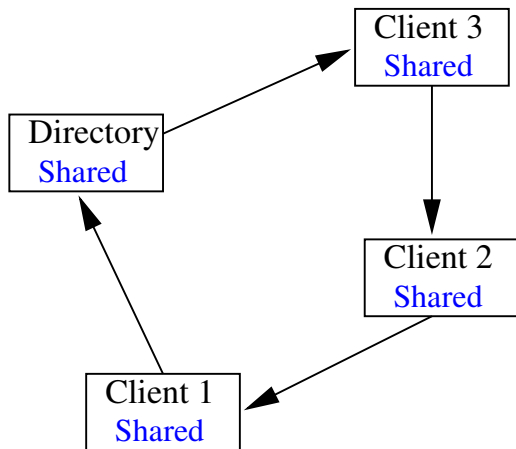


- ▶ stan(**Directory**) := **shared**
- ▶ next(**Directory**) := **Client 3**
- ▶ stan(**Client 3**) := **shared**

Protokół COMA - odpowiedź Directory na read request

Przypadek 1

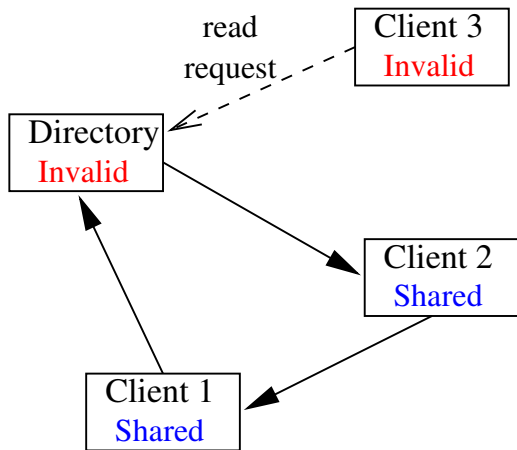
Directory ma aktualne dane (jest w stanie **shared** lub **exclusive**).



Protokół COMA - odpowiedź Directory na read request

Przypadek 2

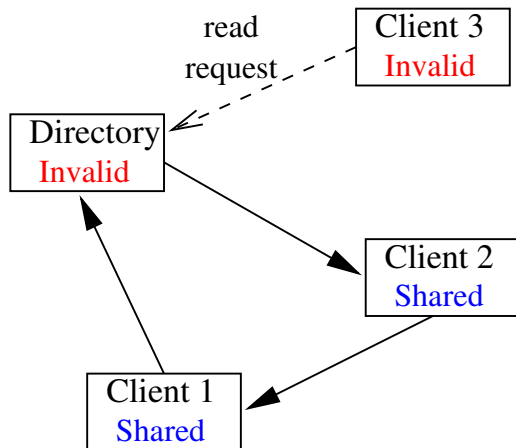
Directory nie ma aktualnych danych (jest w stanie **invalid**).



Protokół COMA - odpowiedź Directory na read request

Przypadek 2

Directory nie ma aktualnych danych (jest w stanie **invalid**).

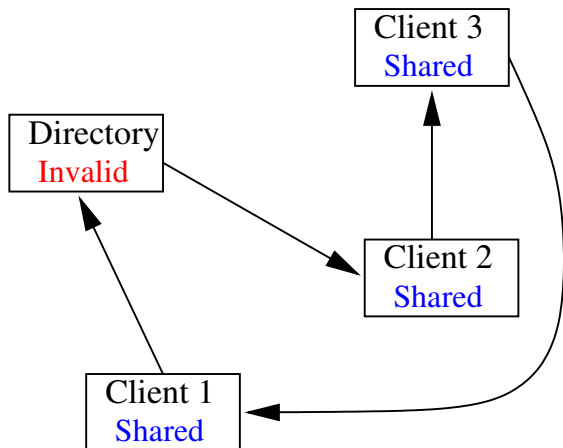


- ▶ **Directory** forwarduje żądanie do **Client 2**
- ▶ $\text{stan}(\mathbf{Client\ 2}) := \text{shared}$
- ▶ $\text{next}(\mathbf{Client\ 2}) := \mathbf{Client\ 3}$
- ▶ $\text{stan}(\mathbf{Client\ 3}) := \text{shared}$

Protokół COMA - odpowiedź Directory na read request

Przypadek 2

Directory nie ma aktualnych danych (jest w stanie **invalid**).



Proces, który chce **zapisać dane**:

- ▶ Jeśli ma aktualne dane na wyłączność (stan **exclusive**)
⇒ **zapisuje**
- ▶ Jeśli ma aktualne dane bez wyłączności (stan **shared**)
⇒ wysyła żądanie do **Directory** (write request), by uzyskać do nich dostęp na wyłączność.
 - ▶ **Directory** wysyła żądanie inwalidacji (invalidation request) wzdłuż łańcucha validity chain
 - ▶ Kiedy żądanie wraca do **Directory**, **Directory** nadaje proszącemu procesowi dostęp na wyłączność.
- ▶ Jeśli ma nieaktualne dane (stan **invalid**)
⇒ j.w. ale jednocześnie trzeba pobrać aktualne dane; po łańcuchu przesyłane jest **invalidate with data**

Proces, który chce **zapisać dane**:

- ▶ Jeśli ma aktualne dane na wyłączność (stan **exclusive**)
⇒ **zapisuje**
- ▶ Jeśli ma aktualne dane bez wyłączności (stan **shared**)
⇒ wysyła żądanie do **Directory** (write request), by uzyskać do nich dostęp na wyłączność.
 - ▶ **Directory** wysyła żądanie inwalidacji (invalidation request) wzdłuż łańcucha validity chain
 - ▶ Kiedy żądanie wraca do **Directory**, **Directory** nadaje proszącemu procesowi dostęp na wyłączność.
- ▶ Jeśli ma nieaktualne dane (stan **invalid**)
⇒ j.w. ale jednocześnie trzeba pobrać aktualne dane; po łańcuchu przesyłane jest **invalidate with data**

Proces, który chce **zapisać dane**:

- ▶ Jeśli ma aktualne dane na wyłączność (stan **exclusive**)
⇒ **zapisuje**
- ▶ Jeśli ma aktualne dane bez wyłączności (stan **shared**)
⇒ wysyła żądanie do **Directory** (write request), by uzyskać do nich dostęp na wyłączność.
 - ▶ **Directory** wysyła żądanie inwalidacji (invalidation request) wzdłuż łańcucha validity chain
 - ▶ Kiedy żądanie wraca do **Directory**, **Directory** nadaje proszącemu procesowi dostęp na wyłączność.
- ▶ Jeśli ma nieaktualne dane (stan **invalid**)
⇒ j.w. ale jednocześnie trzeba pobrać aktualne dane; po łańcuchu przesyłane jest **invalidate with data**

Model w CCS

- ▶ nie modelujemy przekazywanych wartości – nie musimy więc korzystać z value passing
- ▶ struktura komunikacji dla ustalonej liczby procesorów (**Directory** + 2 **Clients**)
- ▶ stan procesu w COMA modelowany jest w postaci kilku stanów w CCS, bo w stanie kodujemy wskaźnik next

```
agent Protocol = ( DIRE | CLIENT1i | CLIENT2i )  
                \ InternalEvents ;
```

Początkowo **Directory** jest w stanie **exclusive**, a dwaj **Clients** w stanie **invalid**

Zewnętrznie widoczne akcje

- ▶ **error** – wskazuje wystąpienie niemożliwego żądania w danym stanie
- ▶ **excl0**, **noexcl0**, **excl1**, **noexcl1**, **excl2**, **noexcl2** – wskazują, że dany procesor nabywa lub traci dostęp na wyłączność do danych
- ▶ **dirva**, **oneva**, **twova** – wskazują, że dany proces ma aktualną kopię danych

Ciąg dalszy ...
w plikach źródłowych.