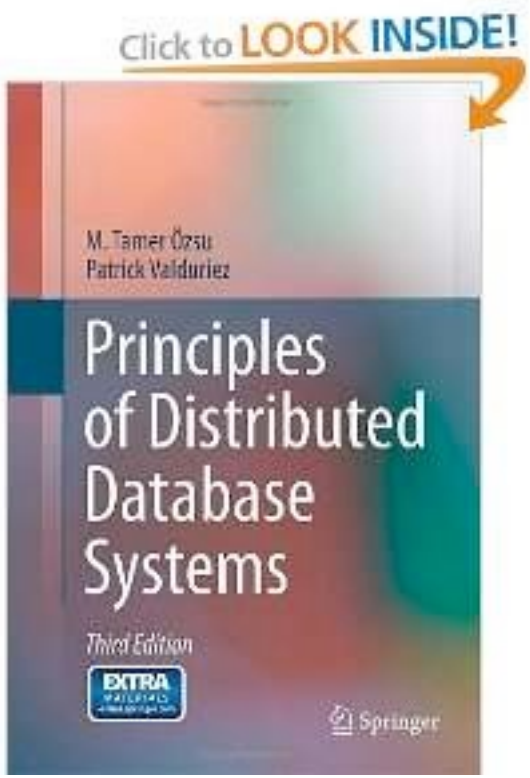
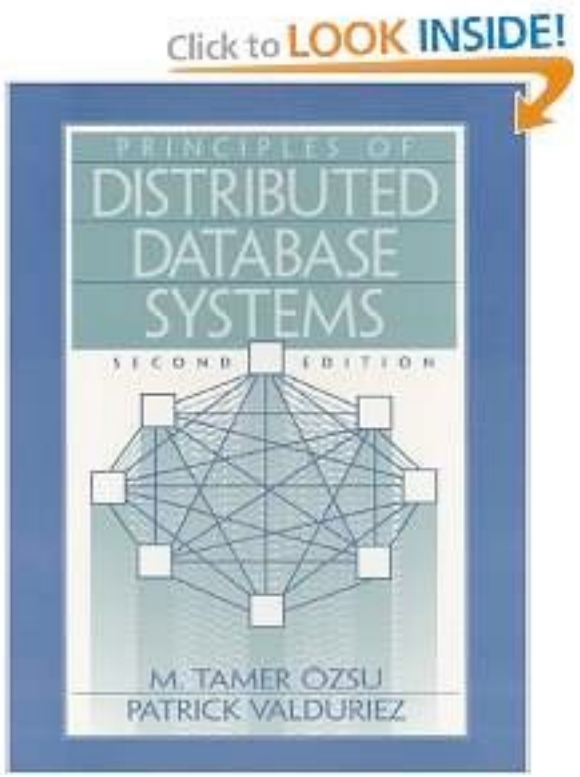
Rozproszone i Mobilne systemy baz danych

Henryk Telega

Zatwierdzanie dwufazowe (2PC)





Zatwierdzanie dwufazowe (2PC)

- Transakcję rozproszoną koordynuje centralnie koordynator globalny. Przebieg transakcji rejestrowany jest w globalnym dzienniku transakcji i lokalnych dziennikach węzłów uczestniczących.
- Dwie fazy: faza głosowania (przygotowania, ang. *prepare phase*) i faza decyzyjna, zatwierdzania (ang. *commit phase*). Opisy wersji protokołu 2PC zaiplementowanej w systemie Oracle zawierają jeszcze fazę o nazwie „zapomnij” (*forget phase*).

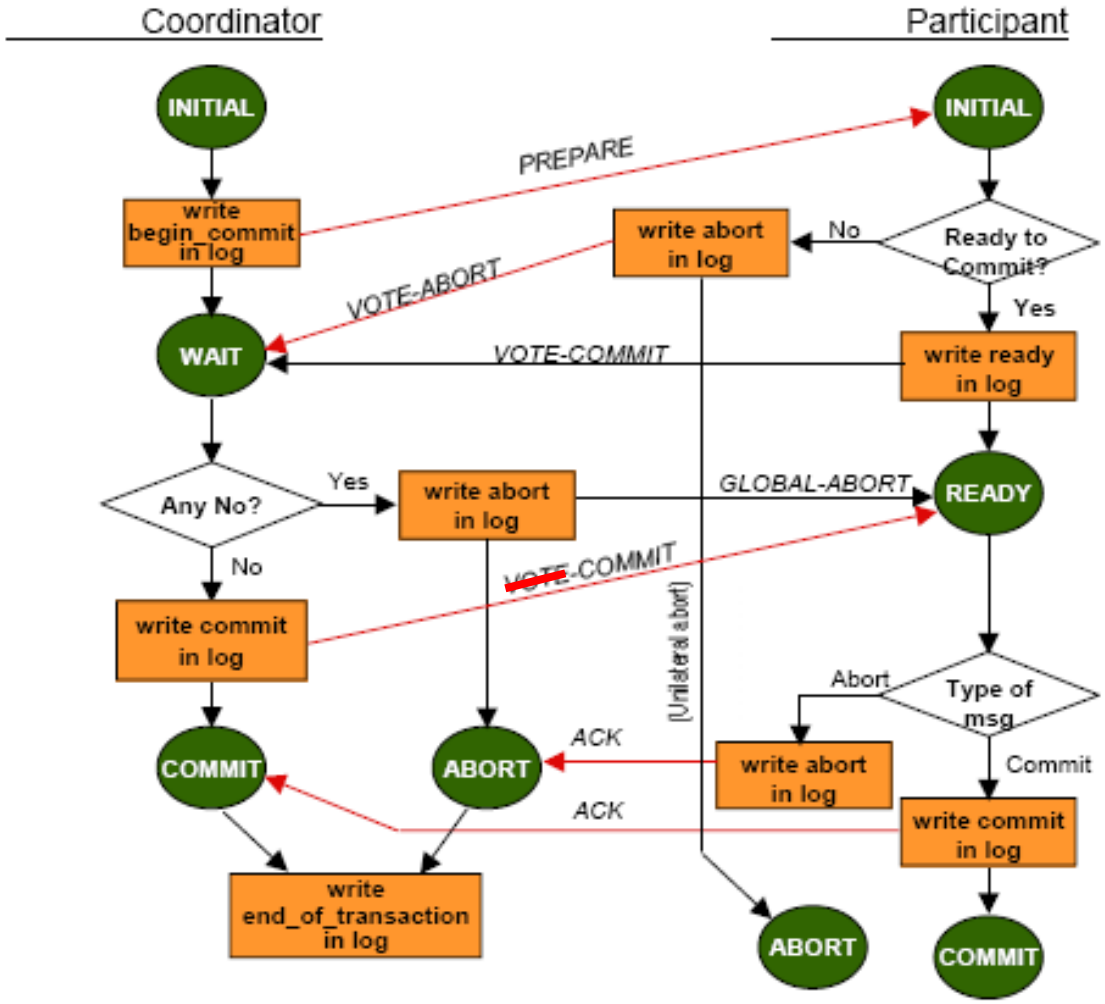
Zatwierdzanie dwufazowe (2PC)

- Koordynator pyta wszystkich uczestników o gotowość do zatwierdzenia (wypełnienia) transakcji.
- Jeśli choć jeden z uczestników głosuje za odrzuceniem transakcji bądź nie udzieli odpowiedzi w ustalonym limicie czasowym, koordynator poleca wszystkim uczestnikom odrzucenie transakcji.
- Jeśli wszyscy zagłosują za zatwierdzeniem, koordynator poleca im zatwierdzenie. Taka globalna decyzja musi już być przyjęta przez wszystkich uczestników (nawet jeśli do uczestnika ta decyzja nie dotrze w wyniku awarii łącz lub węzłów, musi on w końcu ją przyjąć, po naprawieniu awarii i/lub w wyniku działań administratora).

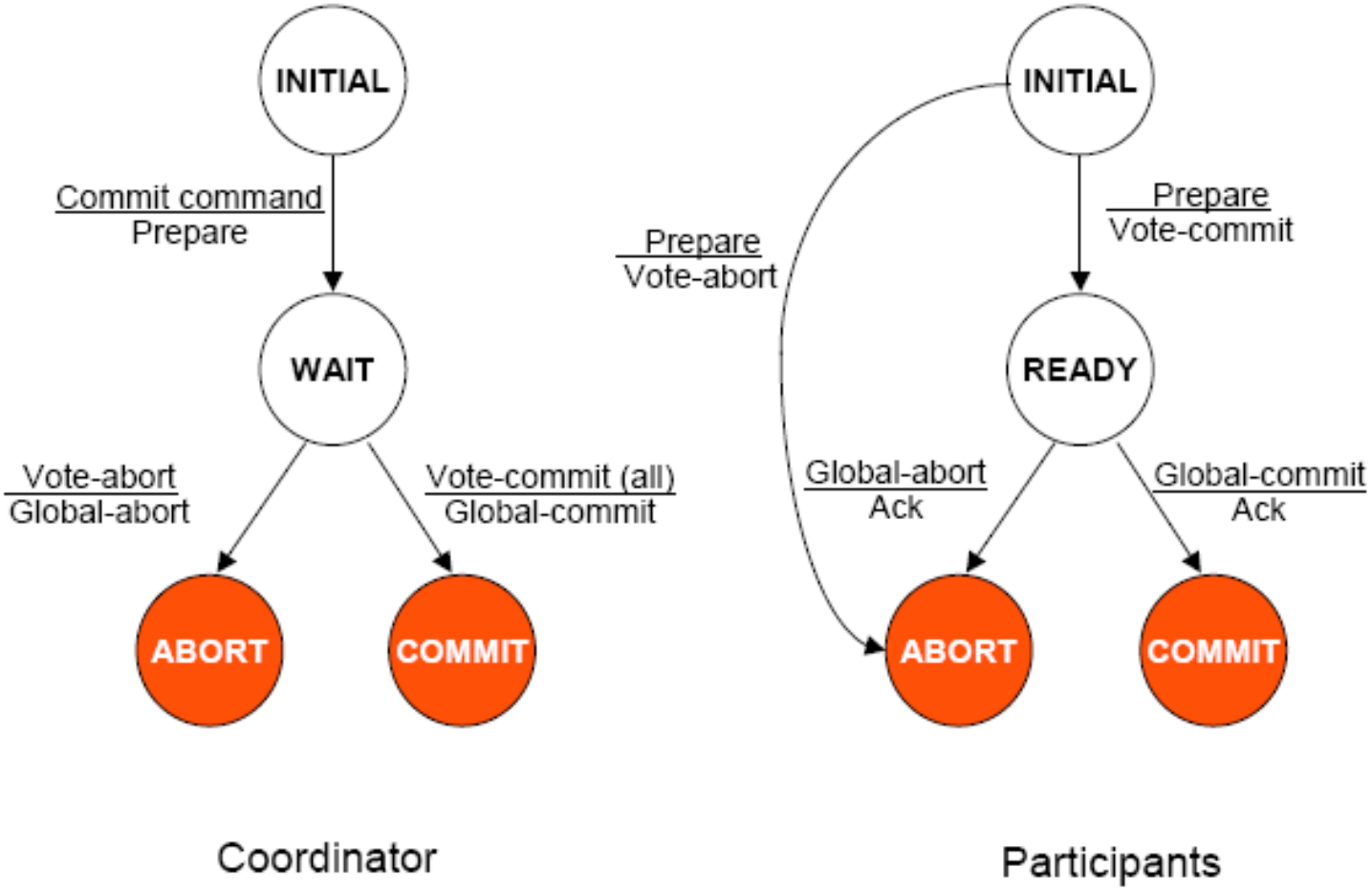
Zatwierdzanie dwufazowe (2PC)

- Uczestnicy powinni czekać aż do uzyskania od koordynatora komunikatu GLOBAL_COMMIT albo GLOBAL_ABORT.
- Jeśli uczestnik w ustalonym czasie nie otrzyma żadnego z tych komunikatów od koordynatora lub koordynator nie otrzyma odpowiedzi od uczestnika (o tej sytuacji mówi się, że nastąpił *time-out*), wówczas zakłada się, że węzeł lub połączenie uległo awarii i konieczne jest uruchomienie tzw. **protokołu zakończenia** (*termination protocol*).
- W realizacji protokołu zakończenia uczestniczą tylko węzły sprawne, natomiast węzły, które uległy uszkodzeniu, wywołują po ponownym uruchomieniu tzw. **protokół odtwarzania** (*recovery protocol*).

2PC Protocol Actions



State Transitions in 2PC



Zatwierdzanie dwufazowe (2PC)

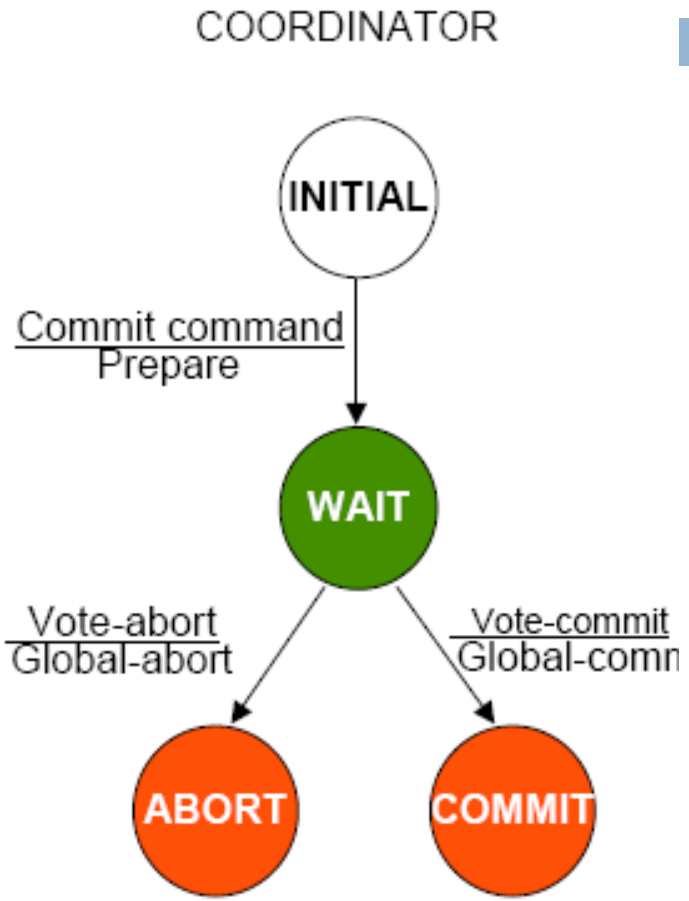
- Procesy koordynatora i uczestników przechodzą przez pewne stany, w których procesy te oczekują na komunikaty. Aby zapewnić wyjście ze stanów stosowane są zegary (timers).
- Protokół zakończenia (termination protocol) uruchamiany jest wtedy, gdy do koordynatora lub uczestnika nie dotrze oczekiwany komunikat w ustalonym limicie czasowym.
- Podejmowane w tej sytuacji działania zależą od tego, czy przekroczenie limitu dotyczy uczestnika czy też koordynatora i na jakim etapie ono wystąpiło.
- Uwaga do poprzedniego slajdu – można rozważać jeszcze jeden stan koordynatora – transakcja zakończona.

Zatwierdzanie dwufazowe (2PC)

- Protokół zatwierdzenia musi zapewnić atomowość i spójność zatwierdzania transakcji.
- Nieblokujący (nonblocking) protokół zakończenia to taki protokół, który pozwala na zakończenie transakcji w sprawnych węzłach bez konieczności oczekiwania na odzyskanie (naprawę) węzłów uszkodzonych i naprawę uszkodzonych łączy.
- Niezależny (independent) protokół odtwarzania to taki protokół, który określa jak zakończyć transakcję (która wykonywała się w chwili awarii) przy odtwarzaniu węzła, bez konieczności komunikacji z innym węzłem.
- Przyjrzyjmy się protokołowi zakończenia dla 2PC.

Site Failures - 2PC Termination

- Timeout in INITIAL
  - Who cares
- Timeout in WAIT
  - Cannot unilaterally commit
  - Can unilaterally abort
- Timeout in ABORT or COMMIT
  - Stay blocked and wait for the acks



2PC: time-out u koordynatora

- Objaśnienia do poprzedniego slajdu:

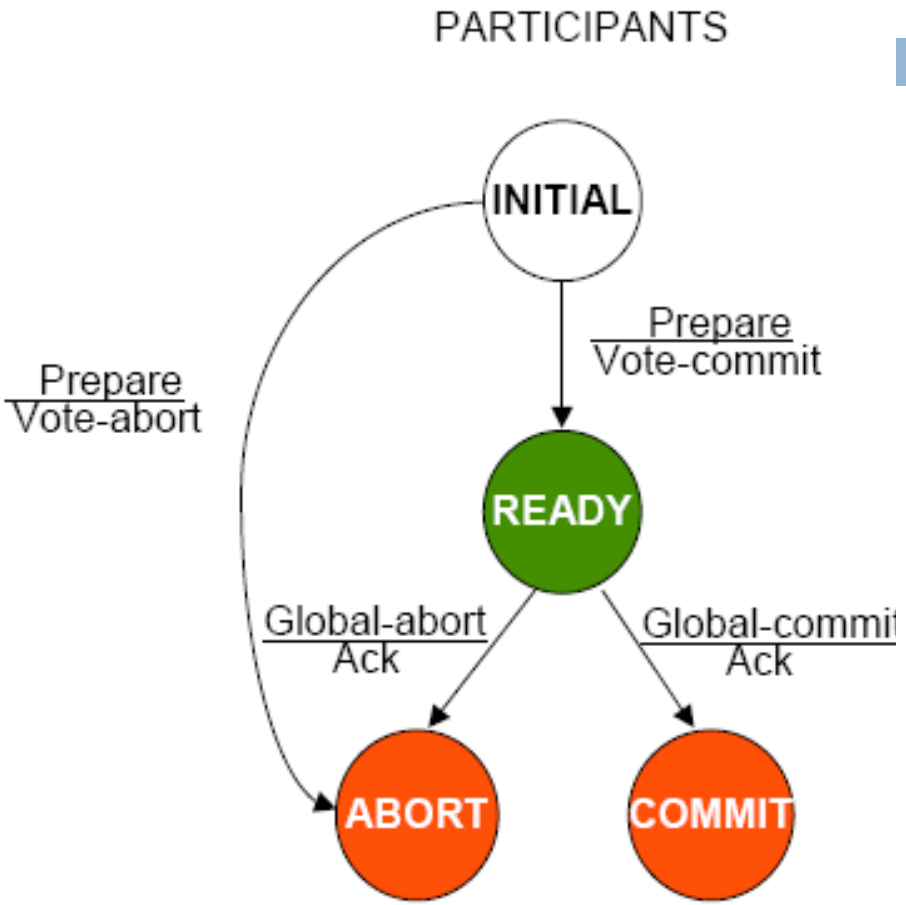
- Time-out w stanie *INITIAL*: koordynator nie rozpoczął wykonywania operacji COMMIT. Oznacza to, że protokół 2PC nie będzie wykonywany w koordynatorze.
- Time-out w tym stanie oznacza ograniczenie czasowe na wykonanie transakcji (nie musi być narzucane). Taki time-out może wynikać np. ze zbyt długiego oczekiwania na wynik jakiejś operacji. Przyczyną time-out'u może być zakleszczenie (jeśli w systemie zakleszczenia nie są wykrywane np. przez wykrywanie cyklu w grafie Wait-For-Graph). Transakcja może zostać przerwana. Może być realizowana próba wykonania całej transakcji jeszcze raz.

2PC: time-out u koordynatora

- W stanie *WAIT*: nie można transakcji zatwierdzić, ale można ją przerwać.
- W stanie *COMMIT* lub *ABORT*: należy czekać na potwierdzenie od tych węzłów, do których została przesłana końcowa decyzja, czyli do tych, które głosowały za zatwierdzeniem transakcji (nie jest ona wysyłana do tych węzłów, które głosowały za wycofaniem transakcji). Globalna decyzja może być okresowo wysyłana do węzłów, od których nie uzyskano potwierdzenia.
- W tych stanach los transakcji jest już znany i nie można go zmienić (w żadnym węźle), zatem wszelkie blokady chroniące pewne elementy danych (jeśli takie blokady są założone) można zdjąć.

Site Failures - 2PC Termination

- Timeout in INITIAL
  - Coordinator must have failed in INITIAL state
  - Unilaterally abort
- Timeout in READY
  - Stay blocked



2PC: time-out u uczestnika

- Objaśnienia do poprzedniego slajdu:

- W stanie *INITIAL*: np. koordynator uległ awarii i instrukcja COMMIT nie została wykonana na koordynatorze lub wskutek awarii sieci komunikat *prepare* nie dotarł do uczestnika. Uczestnik realizował polecenia w ramach tej transakcji. Nie można jednostronnie zatwierdzić, ale można wycofać.
- W stanie *READY* : Nie można jednostronnie zatwierdzić, nie można też wycofać. Przykład – koordynator podjął już decyzję *GLOBAL COMMIT*, być może nawet wysłał ją do pewnych węzłów, ale uległ awarii zanim wysłał do węzła, który doświadczył timeout'u w stanie *READY*.

2PC Termination - uwagi

- (**Uczestnik**) Timeout w stanie INITIAL: uczestnik może przerwać transakcję, koordynator po odtworzeniu prześle komunikat *prepare*, który może być obsłużony przez
  - odczyt lokalnego dziennika transakcji i odszukanie wpisu *abort* bądź
  - może być zignorowany, wówczas koordynator będzie działał według swojego protokołu zakończenia (timeout w stanie WAIT).

2PC Termination, struktura scentralizowana

- Scentralizowana struktura komunikacji: uczestnik, który chce zakończyć transakcję musi spytać koordynatora o jego decyzję i musi czekać aż otrzyma odpowiedź. Uczestnik porozumiewa się wyłącznie ze swoim koordynatorem.
- W rzeczywistych systemach węzeł może być koordynatorem lokalnym, jego węzeł podrzędny może być koordynatorem lokalnym dla innych węzłów itd. (struktura drzewiasta).
- Jeśli koordynator nie działa, uczestnik pozostaje zablokowany.

2PC Termination, struktura rozproszona

- W przypadku, gdy jest możliwa bezpośrednia komunikacja między uczestnikami, uczestnik, który nie otrzyma oczekiwanego komunikatu od koordynatora może poprosić pozostałych uczestników o informacje, które pomogą podjąć decyzję.
- Przyjmijmy takie oznaczenia:
- P_i – uczestnik, który nie otrzymał komunikatu od koordynatora w określonym czasie (doświadczył time-out'u)
- P_k – pozostali uczestnicy.

2PC Termination, uczestnicy mogą się komunikować nawzajem

- Każdy P_k odpowiada w zależności od stanu:
  - P_k jest w stanie INITIAL. To oznacza, że P_k jeszcze nie zagłosował (vote-commit lub vote-abort) a być może nawet nie dostał komunikatu „prepare”. Zatem uczestnik P_k może zdecydować o wycofaniu transakcji i wysłać do P_i komunikat „vote-abort”.
  - P_k jest w stanie READY. W tym stanie P_k głosował za zatwierdzeniem transakcji, ale nie dostał informacji od koordynatora, jaka jest ostateczna decyzja. Taki uczestnik nie może pomóc w poprawnym zakończeniu transakcji przez P_i.
  - P_k jest w stanie COMMIT lub ABORT. W tych stanach albo P_k zadecydował jednostronnie aby przerwać transakcję (unilateral abort) albo dostał od koordynatora decyzję o globalnym zakończeniu transakcji. Ten uczestnik może wysłać do P_i komunikat „vote-abort” (ew. „global-abort”) albo „global-commit”.

2PC Term., jak uczestnik P_i może interpretować odpowiedzi od P_k?

1. P_i dostał komunikaty „vote-abort” od wszystkich P_k. To oznacza, że żaden z pozostałych uczestników jeszcze nie zagłosował, ale wszyscy zdecydowali jednostronnie aby wycofać transakcję. P_i może przerwać transakcję.
2. P_i otrzymał komunikaty „vote-abort” od pewnych P_k, ale pewni uczestnicy odpowiedzieli, że są w stanie READY. W tym przypadku P_i może również wycofać transakcję.
3. P_i otrzymał od wszystkich (działających) P_k informację, że są w stanie READY. W tym przypadku żaden z uczestników nie może pomóc w poprawnym zakończeniu transakcji (trzeba czekać na odtworzenie koordynatora ew. wybrać nowego koordynatora jeśli wszyscy uczestnicy realizujący działania na danych w ramach tej transakcji są sprawni (co oznacza, że koordynator w ramach tej transakcji nie wykonuje operacji na danych, czyli nie jest jednocześnie zwykłym uczestnikiem).

2PC Term., jak uczestnik P_i może interpretować odpowiedzi od P_k?

4. P_i otrzymał komunikaty „global-abort” lub „global-commit” od wszystkich P_k. W tym przypadku wszyscy poza P_i otrzymali decyzję od koordynatora. P_i może zakończyć transakcję zgodnie z otrzymanymi komunikatami.
5. P_i otrzymał komunikaty „global-abort” lub „global-commit” od pewnych P_k, podczas gdy inni wskazują, że są w stanie READY. Oznacza to, że część węzłów otrzymała decyzję od koordynatora, podczas gdy inne węzły jeszcze czekają na komunikat. P_i może zakończyć transakcję jak powyżej.
6. Inne możliwości nie mogą być wynikiem poprawnego wykonania protokołu 2PC.

2PC jest synchroniczny

- Procesy działające według protokołu 2PC w koordynatorze i u uczestników nie mogą być bardziej odległe od siebie (w sensie poziomu stanu – patrz slajd 6 i 7) niż o jedno przejście.
- Na przykład, jeśli uczestnik jest w stanie INITIAL, to wszyscy pozostali uczestnicy muszą być w stanie INITIAL lub READY lub ABORT. Koordynator musi być albo w stanie INITIAL albo WAIT.
- Mówi się, że procesy działające według protokołu 2PC są synchroniczne w obrębie jednego przejścia między stanami (*synchronous within one state transition*).

2PC Wybór nowego koordynatora

- W przypadku przekroczenia czasu w stanie READY uczestnik jest zablokowany, nie może zakończyć transakcji. Przy pewnych założeniach można zakończyć tę blokadę.
- Jeśli wszyscy uczestnicy odkryją, że tylko koordynator nie działa (i nie jest on równocześnie uczestnikiem), mogą wybrać nowego koordynatora i zrestartować proces zatwierdzania.
- Wybór nowego koordynatora: np. według skonfigurowanych priorytetów węzłów lub z wykorzystaniem procedury głosowania między uczestnikami.

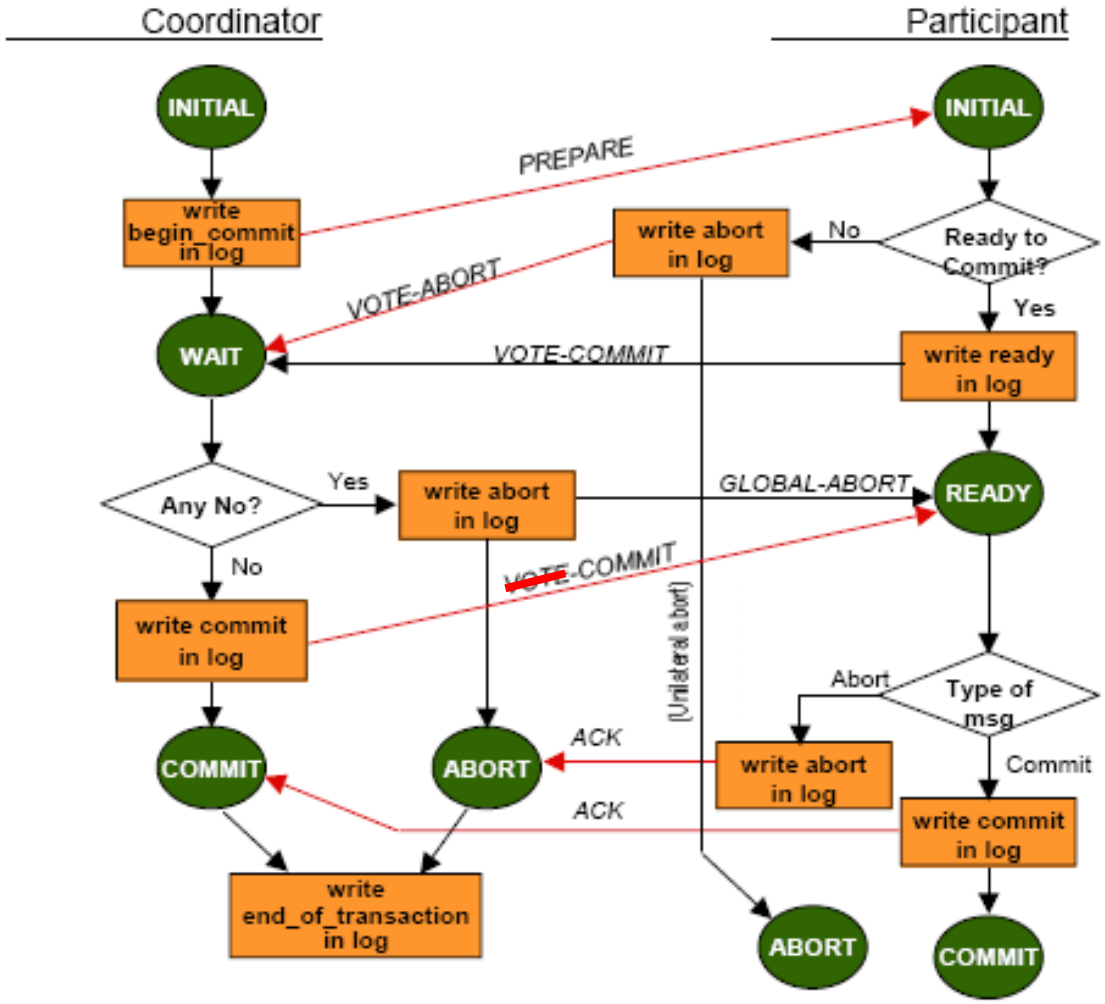
2PC Wybór nowego koordynatora

- Jeśli uszkodzeniu uległ oprócz koordynatora jeszcze jeden węzeł, to wybór nowego koordynatora jest bezużyteczny. Jest możliwe, że węzeł uczestnik tuż przed uszkodzeniem jako jedyny dostał od koordynatora (który też za chwilę uległ uszkodzeniu) globalną decyzję. Ta decyzja nie jest znana pozostałym uczestnikom.
- Nie jest możliwe skonstruowanie nieblokujących protokołów zakończenia dla 2PC.
- Protokół 2PC jest blokujący.

2PC – **Odtwarzanie** (recovery)

- Chcemy, żeby **protokół odtwarzania** (protokół uruchamiany po odtworzeniu w węźle, który uległ awarii) był niezależny.
- Na początek załóżmy, że akcja zapisania rekordu w dzienniku transakcji oraz wysłanie wiadomości stanowią nierozłączną całość. Zrezygnujemy z tego założenia później.
- Przejście między stanami zachodzi po wysłaniu komunikatu.
- Na przykład, jeśli koordynator jest w stanie WAIT, to oznacza, że zapisał begin_commit w dzienniku i wysłał komunikat „prepare”.
- Nie zakładamy nic o tym, czy komunikat dotarł do swojego celu. Np. komunikat „prepare” mógłby nie dotrzeć do pewnych uczestników wskutek awarii łącza komunikacyjnego.

2PC Protocol Actions

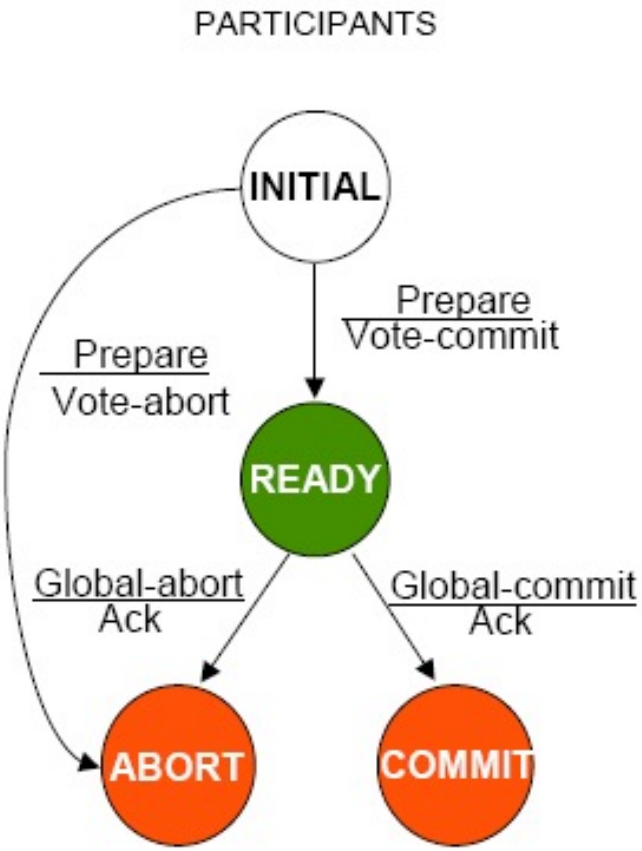


Site Failures - 2PC Recovery

- Failure in INITIAL
  - Start the commit process upon recovery
- Failure in WAIT
  - Restart the commit process upon recovery
- Failure in ABORT or COMMIT
  - Nothing special if all the acks have been received
  - Otherwise the termination protocol is involved

Site Failures - 2PC Recovery

- Failure in INITIAL
  - Unilaterally abort upon recovery
- Failure in READY
  - The coordinator has been informed about the local decision
  - Treat as timeout in READY state and invoke the termination protocol
- Failure in ABORT or COMMIT
  - Nothing special needs to be done



2PC Recovery Protocols – Additional Cases

Arise due to non-atomicity of log and message send actions

- Coordinator site fails after writing "begin_commit" log and before sending "prepare" command
  - treat it as a failure in WAIT state; send "prepare" command
- Participant site fails after writing "ready" record in log but before "vote-commit" is sent
  - treat it as failure in READY state
  - alternatively, can send "vote-commit" upon recovery
- Participant site fails after writing "abort" record in log but before "vote-abort" is sent
  - no need to do anything upon recovery

2PC Recovery Protocols – Additional Case

- Coordinator site fails after logging its final decision record but before sending its decision to the participants
  - coordinator treats it as a failure in COMMIT or ABORT state
  - participants treat it as timeout in the READY state
- Participant site fails after writing "abort" or "commit" record in log but before acknowledgement is sent
  - participant treats it as failure in COMMIT or ABORT state
  - coordinator will handle it by timeout in COMMIT or ABORT state