Konstrukcja spójnego obrazu stanu globalnego - wprowadzenie

Plan wykładu

Celem wykładu jest zaznajomienie studenta z problematyką konstrukcji obrazu spójnego stanu globalnego. Wykład obejmie omówienie podstawowych pojęć związanych z przedstawianą tematyką, takich jak konfiguracja, konfiguracja spójna, odcięcie, linia odcięcia, predykaty globalne, modele stanów globalnych. Pokazane zostaną relacje między tymi pojęciami. Wyjaśnione zostanie znaczenie problemu wyznaczania stanu globalnego oraz przyczyny dla których jest to problem nietrywialny. Na zakończenie zostanie omówiona koncepcja konstrukcji spójnego obrazu stanu globalnego przy przyjęciu pewnych dodatkowych, upraszczających założeń.

Pojęcia podstawowe

Wykład bieżący wymaga od studenta znajomości pewnych podstawowych pojęć przedstawionych już uprzednio. Między innymi są to pojęcia *procesu rozproszonego*, *wykonania* oraz *częściowego wykonania* procesu rozproszonego, pojęcie *śladu wykonania* oraz *historii wykonania*, a także *stanu osiągalnego* oraz *globalnego stanu osiągalnego*.

Obecnie tylko nieformalnie i pobieżnie przypomnimy, że jako proces rozproszony rozumiemy współbieżne wykonanie zbioru procesów sekwencyjnych. Ciąg stanów procesu nazywamy śladem wykonania zaś ciąg zdarzeń historią wykonania. Stanem osiągalnym nazywamy stan procesu taki, który może pojawić się w czasie jego wykonania, zaś stan ten jest stanem końcowym, to określa się go globalnym stanem osiągalnym.

Konfiguracja

Iloczyn kartezjański $\mathcal{S}_1 \times \mathcal{S}_2 \times \ldots \times \mathcal{S}_n$ będziemy oznaczać przez $\boldsymbol{\Gamma}$ i nazywać **zbiorem konfiguracji** (obrazów stanu globalnego) procesu rozproszonego $\Pi = \langle \boldsymbol{\Sigma}, \boldsymbol{\Sigma}^0, \boldsymbol{\Lambda}, \boldsymbol{\Phi} \rangle$.

Konfiguracja $\Gamma \in \boldsymbol{\Gamma}$ jest wektorem $\langle S_1^{k1}, S_2^{k2}, \ldots, S_n^{kn} \rangle$ stanów lokalnych (historii lokalnych) wszystkich procesów $P_1, P_2, \ldots, P_n$, takim że dla każdego u, $1 \le u \le n$, $S_u^{ku} \in \mathcal{S}_u$.

Łatwo zauważyć, że $\boldsymbol{\Gamma}$ zawiera zbiór stanów osiągalnych procesu Π.

Konfiguracja (1)

Konfigurację Γ nazwiemy konfiguracją spójną lub obrazem spójnym jeżeli $\forall E \; \forall E'$ zachodzi:

$$(E' \in \Gamma \wedge E \mapsto E') \Rightarrow (E \in \Gamma)$$

Warunek ten oznacza, że jeśli jakieś zdarzenie E' jest elementem konfiguracji spójnej Γ (to znaczy, jeżeli $E' \in \Gamma$), to również wszystkie zdarzenia E, od których E' przyczynowo zależy (czyli $E \mapsto E'$) również są elementami tej konfiguracji spójnej.

Konfiguracja (2)

Twierdzenie 7.1

Konfiguracja $\Gamma = \langle S_1^{k1}, S_2^{k2}, \ldots, S_n^{kn} \rangle$, reprezentująca stan osiągalny przetwarzania rozproszonego Π jest konfiguracją spójną.

Twierdzenie to wynika wprost z definicji stanu osiągalnego. Wszystkie zdarzenia wchodzące w skład stan osiągalnego procesu rozproszonego pojawiły się w wyniku pewnego wykonania. Oznacza to, że jeżeli jakieś zdarzenie E wchodzi w skład stanu osiągalnego, to także muszą się w nim znaleźć wszystkie zdarzenia, od których E jest przyczynowo zależne.

Konfiguracja (3)

Twierdzenie 7.2

Jeżeli konfiguracja $\Gamma = \langle S_1^{k1}, S_2^{k2}, \dots, S_n^{kn} \rangle$ jest konfiguracją spójną, w której lokalne stany składowe S_u^{ku} są osiągalne w pewnej realizacji przetwarzania rozproszonego Π, to istnieje stan osiągalny $\Sigma(\tau)$, taki że dla każdego, $1 \le u \le n$, $S_u(\tau) = S_u^{ku}$.

Dowód wynika wprost z definicji konfiguracji spójnej i stanu osiągalnego.

Linia odcięcia

Oznaczmy przez $\sigma_1^{k1}, \sigma_2^{k2}, \dots, \sigma_n^{kn}$ ciąg wybranych punktów (zdarzeń pozornych) odcinków czasu odpowiadających stanom $S_1^{k1}, S_2^{k2}, \dots, S_n^{kn}$ poszczególnych procesów zaznaczonych na pewnym diagramie przestrzenno-czasowym. Linię łamaną łączącą punkty $\sigma_1^{k1}, \sigma_2^{k2}, \dots, \sigma_n^{kn}$ nazywać będziemy **linią odcięcia** (lub linią obcięcia). Linia odcięcia dzieli zbiór zdarzeń na **przeszłość** (te zdarzenia, które zaszły przed linią odcięcia) i **przyszłość** (te zdarzenia, które zaszły po linii odcięcia).

Odcięciem $\boldsymbol{\Psi}$ (albo: obcięciem) zbioru zdarzeń $\boldsymbol{\Lambda}$ nazwiemy skończony zbiór $\Psi \subseteq \boldsymbol{\Lambda}$, taki że:

$$(E' \in \Psi \wedge E \mapsto_i E') \Rightarrow (E \in \Psi)$$

Definicja ta mówi, że jeżeli jakieś zdarzenie *E'* należy do odcięcia, to także wszystkie zdarzenia *E,* które są lokalnie poprzedzane przez *E'*, należą do tego odcięcia.

Powiemy, że odcięcie Ψ_2 jest **późniejsze** od odcięcia Ψ_1, jeżeli $\Psi_1 \subseteq \Psi_2$. Oznacza to, że odcięcie Ψ_2 nie zawiera żadnego takiego zdarzenia *E'* nie należącego do Ψ_1, które by lokalnie poprzedzało jakiekolwiek zdarzenie *E* znajdujące się w odcięciu Ψ_1.

Odcięcie spójne (1)

Odcięcie Ψ zbioru zdarzeń $\boldsymbol{\Lambda}$ nazwiemy **odcięciem spójnym***,* gdy:

$$(E' \in \Psi \wedge E \mapsto E') \Rightarrow (E \in \Psi)$$

Definicja ta mówi, że jeżeli jakieś zdarzenie *E'* należy do odcięcia, to także wszystkie zdarzenia *E* od których *E'* jest zależne przyczynowo, należą do tego odcięcia.

Odcięcie spójne - przykład

Na rysunku widzimy dwa odcięcia spójne, Ψ_1 oraz Ψ_2. Pierwsze z nich obejmuje dwa zdarzenia E_1^1 oraz E_2^1. Reprezentuje ono konfigurację, w której wiadomość M_1 jest w kanale. Drugie jest późniejsze od Ψ_1 ($\Psi_1 \subseteq \Psi_2$ – wszystkie zdarzenia należące do Ψ_1 należą także do Ψ_2) i obejmuje dodatkowo zdarzenia E_1^2, E_1^3, E_2^2, E_2^3, E_2^4, E_3^1 oraz E_3^2. Odcięcie to reprezentuje sytuację, w której wszystkie kanały są puste. Gdyby odcięcie Ψ_1 obejmowało dodatkowo zdarzenie E_2^2, nie byłoby odcięciem spójnym, gdyż istniałoby takie zdarzenie (E_3^1), od którego E_2^2 byłoby przyczynowe zależne, które równocześnie nie należałoby do odcięcia Ψ_1.

Natomiast rozszerzenie Ψ_1 o zdarzenie E_3^1 również byłoby odcięciem spójnym.

Odcięcie spójne (2)

Twierdzenie 7.3

Dla odcięcia spójnego z linią odcięcia σ_1^{k1} , σ_2^{k2} , ... , σ_n^{kn} żadna para stanów (zdarzeń) odpowiadających linii odcięcia nie jest wzajemnie zależna.

Odcięcie spójne (3)

Twierdzenie 7.4

Dla każdego diagramu przestrzenno – czasowego z odcięciem spójnym określonym przez linię odcięcia σ_1^{k1} , σ_2^{k2} , ... , σ_n^{kn} istnieje równoważny diagram przestrzenno-czasowy, w którym linię odcięcia σ_1^{k1} , σ_2^{k2} , ... , σ_n^{kn} tworzą zdarzenia równoczesne w sensie czasu globalnego τ.

Twierdzenie to oznacza, że linie łamane reprezentujące odcięcia na diagramach przestrzenno-czasowym można „przesuwać” tak, by otrzymywać równoległe linie proste reprezentujące równoważne odcięcia.

Maksymalne odcięcie spójne

Mając dane pewne odcięcie Ψ, zawsze można wyznaczyć zbiór spójnych odcięć $\mathbf{\Psi}$, takich, że:

$$\forall \Psi_1 \in \mathbf{\Psi} :: \Psi_1 \subseteq \Psi \qquad (7.4).$$

Trywialnym przykładem jest odcięcie obejmujące zbiór zdarzeń początkowych.

Maksymalnym odcięciem spójnym określa się *najświeższe* (*najnowsze*) odcięcie należące do $\mathbf{\Psi}$. Formalnie zdefiniować je można jako takie odcięcie spójne $\Psi_{max} \in \mathbf{\Psi}$, że $\nexists \Psi_2 \in \mathbf{\Psi} :: \Psi_{max} \subset \Psi_2$.

Wyznaczenie odcięcia spójnego

Wyznaczenie najświeższego (ang. *most recent*) odcięcia spójnego można wyznaczyć za pomocą następujących kroków:

- Każdy proces otrzymuje jako dane wejściowe pewne odcięcie Ψ (które nie może pochodzić z *przyszłości*).
- Każdy proces niezależnie wylicza, które z jego własnych zdarzeń należących do Ψ mogą należeć do maksymalnego odcięcia spójnego.

Wyliczenie to może odbyć się w następujący sposób, zakładając, że wszystkie wiadomości posiadają załączone wektorowe etykiety czasowe. Procesy posiadają nieograniczoną co do wielkości tablicę $store_i$ zawierającą wartość wektorowego zegara czasowego dla każdego zdarzenia w P_i. Kiedy P_i otrzymuje Ψ, przeszukuje tablicę $store_i$ zaczynając od m-tego wpisu (gdzie m jest numerem zdarzenia odpowiadającego P_i w nadesłanym odcięciu) dopóki nie odnajdzie wpisu o największym możliwym indeksie m' nie większym od m, takiego że wektorowa etykieta czasowa $store_i[m']$ jest nie większa od wektorowej etykiety czasowej utworzonej z numerów zdarzeń zawartych w odcięciu Ψ. Zdarzenie, któremu odpowiada odnaleziona etykieta czasowa jest poszukiwanym zdarzeniem.

Odcięcie spójne a konfiguracja spójna

Zgodnie z definicją, każdemu odcięciu Ψ opisanemu przez linię odcięcia σ_1^{k1} , σ_2^{k2} , ..., σ_n^{kn} odpowiada konfiguracja $\Gamma = \langle S_1^{k1}, S_2^{k2}, \ldots, S_n^{kn} \rangle$.

Twierdzenie 7.5

Niech Γ będzie konfiguracją a Ψ odpowiadającym jej odcięciem. Konfiguracja Γ jest konfiguracją spójną, wtedy i tylko wtedy, gdy Ψ jest odcięciem spójnym.

Odcięcie niespójne – przykład

Odcięcia przedstawione na slajdzie nie są odcięciami spójnymi, co łatwo wykazać. Odcięcie Ψ_3 zawiera E_1^1, które jest zdarzeniem odbioru wiadomości M_1 wysłanej w wyniku zdarzenia E_2^1. Tak więc zachodzi przyczynowa zależność między tymi dwoma zdarzeniami ($E_2^1 \mapsto E_1^1$) podczas gdy E_2^1 nie należy do odcięcia Ψ_3. Zgodnie z twierdzeniem 7.5 odcięcie to nie reprezentuje więc żadnej spójnej konfiguracji.

Analogiczna sytuacja zachodzi w przypadku odcięcia Ψ_4. W tym przypadku, zawiera ono zdarzenie E_3^2 odbioru wiadomości M_4 wysłanej w wyniku zdarzenia E_2^3 . Zachodzi więc, podobnie jak poprzednio, przyczynowa zależność między E_2^3 oraz E_3^2 ($E_2^3 \mapsto E_3^2$), podczas gdy zdarzenie nie należy do odcięcia Ψ_4.

Aby te odcięcia stały się odcięciami spójnymi, należy albo je poszerzyć o dodatkowo zdarzenia, albo pewne zdarzenia z nich usunąć. W przypadku odcięcia Ψ_3 należałoby albo usunąć zdarzenie E_1^1, albo dodać zdarzenie E_2^1. W przypadku odcięcia Ψ_4, aby stało się ono odcięciem spójnym, należałoby poszerzyć je o zdarzenie E_2^3, albo usunąć z niego zdarzenie E_3^2.

Predykaty globalne i ich własności

Przez *predykat globalny* $\vartheta(\Sigma)$ będziemy rozumieć predykat zdefiniowany na zbiorze osiągalnych stanów globalnych przetwarzania rozproszonego.

Predykaty opisują właściwości przetwarzania w poszczególnych stanach. Szczególne znaczenie mają w praktyce *predykaty stabilne* (określane czasami własnościami stabilnymi. ang. *stable properties*), których zajście w pewnym stanie globalnym Σ implikuje, że dla każdego stanu Σ' osiągalnego ze stanu Σ, predykat ten jest również prawdziwy. Innymi słowy, predykat jest nazywany stabilnym, gdy spełniany jest następujący warunek:

$$(\vartheta(\Sigma) \wedge (\Sigma \rightsquigarrow \Sigma')) \Rightarrow \vartheta(\Sigma')$$

gdzie $\Sigma \rightsquigarrow \Sigma'$ oznacza, że stan Σ' jest osiągalny ze stanu Σ.

Predykaty, które nie spełniają tego warunku nazwiemy *predykatami niestabilnymi*.

Przykładami predykatów stabilnych są predykaty definiujące stan zakleszczenia, zakończenia przetwarzania, utraty znacznika, przekroczenia czasu obliczeń czy czasu transmisji itp.

Predykaty można także klasyfikować w inny sposób. Przykładem są tutaj predykaty słabe, które uznaje się za predykaty prawdziwe w przetwarzaniu rozproszonym, jeżeli istnieje taki spójny stan globalny, w którym są spełnione, oraz predykaty *silne*, które uznaje się za prawdziwe w przetwarzaniu rozproszonym wtedy, gdy są uznawane za prawdziwe w pewnej chwili przez wszystkich uczestników przetwarzania.

Duże znaczenie praktyczne posiadają też predykaty zdefiniowane na zbiorze wykonań $\boldsymbol{\Upsilon}$, a w szczególności predykaty *possibly* oraz *definitely*.

Predykat possibly(ϑ) zachodzi wtedy i tylko wtedy, gdy istnieje wykonanie $\Upsilon \in \boldsymbol{\Upsilon}$ zawierające stan globalny Σ, dla którego zachodzi predykat $\vartheta(\Sigma)$.

Predykat definitely(ϑ) jest prawdziwy wtedy i tylko wtedy, gdy w każdym możliwym wykonaniu osiągalny jest stan Σ, dla którego zachodzi $\vartheta(\Sigma)$.

W ogólności, wyznaczenie predykatów globalnych w pełni asynchronicznym systemie rozproszonym bez przyjęcia dodatkowych założeń jest niemożliwe, jeżeli chociaż jeden proces może ulec awarii.

Stan globalny systemu

W systemach rozproszonych, bardzo wiele problemów praktycznych i teoretycznych sprowadza się wprost do ciągłej lub okresowej obserwacji stanu globalnego systemu, określonego jako zbioru stanów lokalnych procesów oraz wiadomości znajdujących się w kanałach komunikacyjnych.

Znajomość takiego stanu pozwala bowiem na wykrycie specyficznych sytuacji, niewidocznych z perspektywy pojedynczego procesu, i podjęcie stosownych działań.

Przykładami problemów redukujących się w istocie do oceny stanu globalnego są:

- śledzenie i sterowanie wykonywaniem programu rozproszonego (ang. *monitoring and debugging*),
- detekcja stanów awaryjnych (np. utraty wiadomości, zakleszczenia) lub też oczekiwanych (np. zakończenia obliczeń rozproszonych). Stany takie można wykryć tworząc obraz stanu globalnego i następnie dokonując jego analizy (Należy jednak wspomnieć w tym miejscu, że rozwiązania specjalizowane są często bardziej efektywne).
- dostosowywanie konfiguracji i funkcji systemu do zmieniającego się obciążenia (np. podział i równoważenie obciążenia, adaptacyjny wybór połączeń).
- wyliczenie pewnej globalnej wartości zarządzanej przez program rozproszony.
- utworzenie kopii zapasowych stanu globalnego programu.

Obraz stanu globalnego nazywa się też czasami *migawką stanu globalnego* (ang. *global snapshot*).

Nieformalnie, obraz stanu globalnego określić można jako spójny, jeżeli dla każdego z procesów wygląda, jak gdyby został wyznaczony w jednej chwili w wszystkich częściach systemu. Można tutaj określić stan globalny jako zbiór stanów wyznaczonych w jednej chwili w wszystkich częściach systemu – w asynchronicznym środowisku rozproszonym taka definicja jest jednak nierealna. Zamiast tego więc mówi się raczej o zbiorze stanów niezależnych wzajemnie od siebie.

Modele stanów globalnych (1)

Jako przykładowy problem wyznaczania obrazu stanu globalnego rozważmy przetwarzanie rozproszone obejmujące trzy procesy P_1, P_2 i P_3, które współdzielą pewien zasób w trybie wzajemnego wykluczania. Jak wiadomo, problem wzajemnego wykluczania można stosunkowo łatwo rozwiązać przyjmując, że warunkiem koniecznym dostępu do współdzielonego zasobu (sekcji krytycznej) jest posiadanie w danej chwili unikalnego komunikatu – znacznika (ang. *token*). Znacznik krążąc między procesami połączonymi w logiczny pierścień (reprezentowany przez graf cykliczny) wskazuje kolejno procesy uprawnione do operowania na zasobie współdzielonym. Innymi słowy, tylko proces posiadający w danej chwili znacznik posiada równocześnie dostęp do zasobu.

Modele stanów globalnych (2)

Załóżmy, że stan $S_i(\tau)$ procesu P_i w każdej chwili τ czasu globalnego (rzeczywistego) zdefiniowany jest przez trzy zmienne: $present_i(\tau)$, $outLog_i(\tau)$, $inLog_i(\tau)$.

$$S_i(\tau) = \langle\, present_i(\tau), outLog_i(\tau), inLog_i(\tau) \,\rangle$$

- $present_i(\tau)$ przyjmuje wartość *True*, tylko wówczas, gdy znacznik typu TOKEN znajduje się w chwili τ w procesie P_i (proces ten posiada znacznik i zarazem dostęp do współdzielonego zasobu)
- $outLog_i(\tau)$ jest kolejką znaczników wysłanych do chwili τ przez proces P_i.

- $inLog_i(\tau)$ jest kolejką znaczników odebranych przez proces P_i do chwili τ.

Modele stanów globalnych – przykład 1 (1)

Zakładamy, że w globalnym stanie początkowym $\Sigma^0 = \Sigma(\tau_0)$ znacznik znajduje się w procesie P_1, a wszystkie kolejki $outLog_i$, $inLog_i$, (dla i = 1, 2, 3) – są puste.

Tym samym:

$$\begin{array}{llll} \Sigma(\tau_0) = \langle & \langle\, present_1(\tau_0) = True & outLog_1(\tau_0) = \varnothing & inLog_1(\tau_0) = \varnothing \,\rangle \\ & \langle\, present_2(\tau_0) = False & outLog_2(\tau_0) = \varnothing & inLog_2(\tau_0) = \varnothing \,\rangle \\ & \langle\, present_3(\tau_0) = False & outLog_3(\tau_0) = \varnothing & inLog_3(\tau_0) = \varnothing \,\rangle \,\rangle \end{array}$$

Modele stanów globalnych – przykład 1 (2)

Po wysłaniu znacznika $token_1$ będącego konkretną wiadomością typu TOKEN (każda wiadomość, a więc również każdy znacznik, ma swój unikalny identyfikator *mId*) w kierunku procesu P_2, znacznik przez pewien czas znajduje się w kanale $C_{1,2}$ (przypomnijmy, że w ogólności czas przebywania wiadomości w kanale jest skończony, lecz nie znany *a priori*). Tak więc w chwili τ_1 stan globalny można opisać następująco:

$$\begin{array}{llll} \Sigma^1 = \langle & \langle\, present_1(\tau_1) = False & outLog_1(\tau_1) = \{token_1\} & inLog_1(\tau_1) = \varnothing \,\rangle \\ & \langle\, present_2(\tau_1) = False & outLog_2(\tau_1) = \varnothing & inLog_2(\tau_1) = \varnothing \,\rangle \\ & \langle\, present_3(\tau_1) = False & outLog_3(\tau_1) = \varnothing & inLog_3(\tau_1) = \varnothing \,\rangle \,\rangle \end{array}$$

Zauważmy, że znajomość stanów lokalnych procesów P1 oraz P2 w chwili τ_1 pozwala łatwo wyznaczyć stan kanału $C_{1,2}$ jako różnicę mnogościową $outLog_1(\tau_1)$ oraz $inLog_2(\tau_1)$.

Po odebraniu znacznika $token_1$ przez proces P_2 system osiąga stan Σ^2:

$$\begin{array}{llll} \Sigma^2 = \langle & \langle\, present_1(\tau_1) = False & outLog_1(\tau_1) = \{token_1\} & inLog_1(\tau_1) = \varnothing \,\rangle \\ & \langle\, present_2(\tau_1) = True & outLog_2(\tau_1) = \varnothing & inLog_2(\tau_1) = \{token_1\} \,\rangle \\ & \langle\, present_3(\tau_1) = False & outLog_3(\tau_1) = \varnothing & inLog_3(\tau_1) = \varnothing \,\rangle \,\rangle \end{array}$$

Jak widać, w chwili τ_2 kanał jest już pusty i istotnie $outLog_1(\tau_2) \setminus inLog_2(\tau_2) = \varnothing$.

Po wysłaniu przez P_2 znacznika do procesu P_3 osiągany jest kolejny stan, Σ^3:

$$\begin{array}{llll} \Sigma^3 = \langle & \langle\, present_1(\tau_1) = False & outLog_1(\tau_1) = \{token_1\} & inLog_1(\tau_1) = \varnothing \,\rangle \\ & \langle\, present_2(\tau_1) = False & outLog_2(\tau_1) = \{token_2\} & inLog_2(\tau_1) = \{token_1\} \,\rangle \\ & \langle\, present_3(\tau_1) = False & outLog_3(\tau_1) = \varnothing & inLog_3(\tau_1) = \varnothing \,\rangle \,\rangle \end{array}$$

Następnie znacznik jest odbierany przez P_3 i osiągany jest kolejny stan, a potem w analogiczny sposób następne.

Graf stanów osiągalnych

Graf stanów osiągalnych przedstawiony na slajdzie odpowiada przesyłaniu unikalnych znaczników w pierścieniu trzech procesów.

Przedstawiony przykład ilustruje dotychczasowe rozumienie stanu lokalnego procesów oraz stanu globalnego przetwarzania rozproszonego – jako historii przetwarzania.

Należy zaznaczyć, że rozumienie to pozwala na określenie w każdej chwili aktualnego stanu kanałów na podstawie stanów lokalnych procesów.

Modele stanów globalnych (3)

Przedstawione podejście jest powszechnie stosowane, a jego dużą zaletą jest ograniczanie liczby składowych stanu globalnego do liczby procesów ($\Sigma \subseteq \mathcal{S}_1 \times \mathcal{S}_2 \times \ldots \times \mathcal{S}_n$). Z drugiej jednak strony, ma ono z praktycznego punktu widzenia pewne określone wady :

- Podejście jest adekwatne tylko dla systemów z kanałami niezawodnymi
- Poważną trudność stanowi konieczność pamiętania wszystkich dotychczas wysłanych i odebranych wiadomości. Z upływem czasu rośnie bowiem istotnie koszt przechowywania i przetwarzania tej informacji

Modele stanów globalnych (4)

Wady przedstawionego podejścia powodują, że w praktyce stosowane jest też podejście alternatywne, polegające na definiowaniu stanu globalnego jako złożenia stanów lokalnych procesów i stanów kanałów komunikacyjnych. Prowadzi to do reprezentacji stanu globalnego przez $n+m$ składowych odpowiadających stanom lokalnym wszystkich n procesów i wszystkich m kanałów.

Liczba składowych jest w tym wypadku oczywiście większa, lecz reprezentacja stanu procesu może być dużo prostsza i w rezultacie efektywniejsza z punktu widzenia zajętości pamięci i kosztów przetwarzania.

Modele stanów globalnych (5)

W celu ilustracji podejścia alternatywnego rozważmy jeszcze raz przykład przedstawiony na poprzednich slajdach. Przyjmijmy, że stan procesu P_i jest w każdej chwili określony przez zmienną logiczną $present_i$, oraz przez liczniki $sentNo_i$ (ang. *sent number*) i $recvNo_i$ (ang. *receive number*), o wartościach równych liczbie dotychczas wysłanych i odebranych znaczników typu TOKEN. Z kolei stan kanału $L_{i,j}$ określony może być przez zbiór znaczników znajdujących się aktualnie w kanale $C_{i,j}$

Model stanów globalnych (6)

Możliwa jest też reprezentacja jeszcze prostsza, w której stan procesu określony przez zmienną $present_i$ a stan kanału określony przez analogiczną zmienną $present_{i,j}$. Zmienne te przyjmują wartość *true* tylko wówczas, gdy znacznik typu TOKEN znajduje się aktualnie w odpowiednim procesie bądź kanale.

Modele stanów globalnych – przykład 2

Niech dla prostoty zapisu „1” oznacza obecność znacznika, a „0” – jego brak. W efekcie stan globalny $\Sigma(\tau)$ możemy przedstawić w naszym przypadku w sposób następujący:

$$\Sigma(\tau) = \langle S_1(\tau), S_2(\tau), S_3(\tau), L_{1,2}(\tau), L_{2,3}(\tau), L_{3,1}(\tau) \rangle$$

Tym samym kolejne stany mają postać:

$\Sigma^0 = \langle 1, 0, 0, 0, 0, 0 \rangle$ - znacznik jest w posiadaniu procesu P_1, kanały są puste.

$\Sigma^1 = \langle 0, 0, 0, 1, 0, 0 \rangle$ - znacznik znajduje się w kanale $L_{1,2}$ (został wysłany przez P_1 ale jeszcze nie został odebrany przez P_2)

$\Sigma^2 = \langle 0, 1, 0, 0, 0, 0 \rangle$ - znacznik jest w posiadaniu procesu P_2, kanały są puste.

$\Sigma^3 = \langle 0, 0, 0, 0, 1, 0 \rangle$ - znacznik znajduje się w kanale $L_{2,3}$ (został wysłany przez P_2 ale jeszcze nie został odebrany przez P_3)

$\Sigma^4 = \langle 0, 0, 1, 0, 0, 0 \rangle$ - znacznik jest w posiadaniu procesu P_3, kanały są puste.

$\Sigma^5 = \langle 0, 0, 0, 0, 0, 1 \rangle$ - znacznik znajduje się w kanale $L_{3,1}$ (został wysłany przez P_3 ale jeszcze nie został odebrany przez P_1)

Ocena stanów globalnych

W rozważanych przypadkach ocena stanów globalnych w wybranych momentach czasu może mieć na celu sprawdzenie, czy rzeczywiście w każdej chwili w systemie jest dokładnie jeden znacznik. Zaginięcie znacznika, wynikające na przykład z błędu programu w którymkolwiek węźle, uniemożliwia procesom w sposób trwały wejście do sekcji krytycznej, co może prowadzić do blokady całego systemu. Z kolei stwierdzenie wystąpienia dwóch lub więcej znaczników oznacza możliwość jednoczesnego działania dwóch lub więcej procesów w sekcjach krytycznych, co w ogólności prowadzi do błędów współbieżnego działania procesów.

Porównanie reprezentacji stanów globalnych

Przedstawione reprezentacje stanów globalnych nie są równoważne.

- Pierwsza, operuje na informacji najpełniejszej: unikalnych znacznikach i całej historii komunikacji
- Druga reprezentacja, utożsamia wszystkie znaczniki i wyróżnia tylko stany ich obecności oraz nieobecności w poszczególnych procesach i kanałach

Powstaje zatem problem wyboru odpowiedniego modelu stanu. Warto tu podkreślić, że wybór ten zależy istotnie od zastosowania, które w dużej mierze narzuca wymaganą precyzję reprezentacji stanu lokalnego i globalnego.

Niedeterminizm przetwarzania rozproszonego

Na slajdzie tym widzimy różne realizacje przetwarzania rozproszonego *klient-serwer* jako wynik nieprzewidywalnych czasów transmisji. Są na nim przedstawione dwa procesy użytkowe P_1 i P_2, żądające realizacji usługi przez dwa serwery P_3 i P_4. Proces P_4 jest przy tym repliką procesu P_3. Przyjmijmy ponadto, że dla prawidłowego działania procesów użytkowych muszą one korzystać z obu serwerów w trybie wzajemnego wykluczania. Aby to osiągnąć, każdy z procesów P_1 i P_2 wysyła komunikaty zawierające żądania, odpowiednio do obu serwerów, w celu uzyskania wyłącznego do nich dostępu. Ponieważ zdarzenia wysyłania żądań są współbieżne, a kanały wnoszą nieprzewidywalne opóźnienia, żądania mogą dotrzeć do serwerów w różnej kolejności. Jeśli przyjmiemy jeszcze, że procesy P_1 i P_2 mogą kontynuować działanie dopiero po otrzymaniu potwierdzeń od oby procesów P_3 i P_4, skutki niedeterminizmu transmisji mogą być bardzo istotne.

Na górnej części rysunku, w chwili τ_1 (po zakończeniu przesyłania wszystkich wiadomości) został osiągnięty pewien stan $\Sigma(\tau_1)$. w którym proces P_1 może kontynuować działanie, a proces P_2 oczekuje na zwolnienie serwerów przez P_1. Stan ten można uznać za poprawny. Natomiast w dolnej części rysunku widzimy osiągnięcie pewnego stanu $\Sigma(\tau_2)$, w którym proces P_1 otrzymał potwierdzenie od P_3, a proces P_2 otrzymał potwierdzenie od P_4. Dla każdego z procesów P_1 i P_2 z osobna, otrzymane potwierdzenia nie są wystarczające do kontynuacji działania i dalej zwolnienia potwierdzonych serwerów. Zauważmy też, że jeżeli inne, zewnętrzne w stosunku do P_1, P_2, P_3 i P_4, procesy nie wymuszą na P_1 lub P_2 zwolnienia serwerów, to wówczas zarówno serwery jak i procesy żądające usług pozostaną trwale zablokowane. W rozważanym przykładzie, wykrycie wystąpienia stanu $\Sigma(\tau_2)$ odpowiadającego trwałej blokadzie wszystkich procesów, miałoby na celu zainicjowanie specjalnych działań prowadzących do osiągnięcia stanu, w którym dalsze poprawne przetwarzanie byłoby możliwe.

Metoda wyznaczania stanu globalnego

Analizując dotychczas przedstawione modele stanów globalnych, abstrahowano od problemu jednoczesnego wyznaczenia poszczególnych składowych stanu globalnego, a więc wyznaczenia stanów lokalnych wszystkich procesów i ewentualnie wszystkich kanałów w określonej, tej samej chwili. Oznaczało to niejawne założenie istnienia idealnego obserwatora zewnętrznego, który ma dostęp jednocześnie do wszystkich procesów tworzących przetwarzanie co nie jest niestety możliwe w systemie asynchronicznym.

Niestety, w asynchronicznym środowisku rozproszonym takie podejście ma tylko znaczenie teoretyczne z uwagi na jego cechy:

- jedyny mechanizm komunikacji i synchronizacji to wymiana komunikatów
- brak wspólnej pamięci

Proces chcący wyznaczyć stan globalny może zatem tylko wysyłać żądania wyznaczania stanów lokalnych, a po otrzymaniu odpowiedzi, konstruować na ich podstawie stan globalny.

Problemy związane z wyznaczaniem stanu globalnego

Niestety, asynchroniczne środowisko rozproszone posiada znacząco utrudniające to zadanie cechy:

- brak wspólnego zegara
- asynchronizm przetwarzania
- nieprzewidywalne czasy transmisji (opóźnienia komunikacyjne)

W wyniku tego otrzymane składowe stany lokalne procesów mogą być:

- przestarzałe
- niekompletne
- odpowiadać konfiguracjom niespójnym – reprezentującym stany nieosiągalne, których wystąpienia zewnętrzny, idealny obserwator nigdy nie mógłby stwierdzić.

W tym kontekście jest jasne, że żaden lokalny, rzeczywisty proces dążący do wyznaczenia stanu globalnego przetwarzania rozproszonego nie może w ogólności wyznaczyć chwilowych wartości stanu. Proste i naturalne zwiększenie prędkości transmisji może oczywiście poprawić obraz stanu globalnego w sensie aktualności i kompletności, lecz nie jest wystarczające dla zapewnienia spójności.

Dodatkowym źródłem trudności przy ocenie obrazu stanu globalnego jest relatywizm obserwacji, istotny zwłaszcza przy współbieżnej konstrukcji obrazów przez wiele procesów. Nawet, gdy każdy z utworzonych obrazów jest spójny, mogą one się różnić na tyle, że procesy monitorujące mogą podejmować działania konfliktowe.

Problem konstrukcji stanu globalnego

Trudność problemu konstrukcji spójnego obrazu stanu globalnego przetwarzania rozproszonego wynika z:

- braku zegara globalnego
- asynchronizmu przetwarzania

Rozważmy na początek problem przy następujących dodatkowych założeniach:

- dostępny jest dla wszystkich procesów globalny zegar czasu rzeczywistego
- znane jest maksymalne opóźnienie komunikacji,
- względne prędkości przetwarzania w poszczególnych węzłach są ograniczone

Koncepcja konstrukcji obrazu spójnego (1)

Koncepcja wyznaczenia spójnego obrazu przy przyjętych takich założeniach jest bardzo prosta. Wystarczy, by dowolny proces monitora Q_i, wybrał chwilę τ_s na tyle odległą w przyszłości, by zagwarantować, że wiadomość wysłana w danej chwili dotrze do wszystkich procesów Q_j przed τ_s. W celu wyznaczenia stanu kanałów, można przyjąć, że każda wiadomość zawiera etykietę czasową, określającą czas rzeczywisty wysłania wiadomości. Zakładając przy tym, że kanały są niezawodnymi kanałami typu FIFO, ideę algorytmu można przedstawić następująco:

Proces inicjatora Q_α wysyła do monitorów wszystkich procesów przetwarzania rozproszonego wiadomość: ,,zapamiętaj stan lokalny w chwili τ_s"

Monitory Q_i procesów zapamiętują stan lokalny S_i odpowiadających im procesów aplikacyjnych w chwili τ_s i natychmiast wysyłają wiadomość kontrolną do wszystkich monitorów $Q_j \in \mathcal{Q}_i^{OUT}$ (wiadomość ta jest wysyłana wszystkimi kanałami wyjściowymi). Zapamiętują też stan $L_{k,i}$ kanałów wejściowych jako zbiorów wszystkich wiadomości aplikacyjnych, które zostały wysłane przed τ_s a dotarły do Q_i po chwili τ_s. Zapamiętywanie stanu lokalnego oraz wysyłanie wiadomości kontrolnej poprzedza przy tym zajście jakiegokolwiek zdarzenia w procesie aplikacyjnym P_i, dla $\tau \geq \tau_s$

Koncepcja konstrukcji obrazu spójnego (2)

- Gdy monitor Q_i otrzyma przez kanał $C_{j,i}$ pierwszą wiadomość o etykiecie czasowej większej od τ_s (w szczególności może to być wiadomość kontrolna), aktualną wartość $L_{j,i}$ uznaje jako stan tego kanału w chwili τ_s.
- Gdy monitor Q_i otrzyma wiadomości o etykiecie czasowej większej od τ_s przez wszystkie swoje kanały wejściowe, przesyła stan lokalny zapamiętany w chwili τ_s oraz wyznaczone stany kanałów wejściowych do inicjatora detekcji Q_α.

Koncepcja konstrukcji obrazu spójnego (3)

- Po odebraniu od wszystkich monitorów wiadomości zawierających stany lokalne oraz stany ich kanałów wejściowych w chwili τ_s, inicjator Q_α konstruuje obraz stanu globalnego Γ

Zauważmy teraz, że jeżeli zapamiętane stany lokalne obejmowałyby historię komunikacji (kolejki wiadomości wysłanych i odebranych do danej chwili przez procesy aplikacyjne), to oczywiście przesyłanie wiadomości kontrolnych nie byłoby potrzebne. Wiadomo jednak z drugiej strony, że pamiętanie całej historii komunikacji może być zbyt kosztowne w praktyce i wówczas przedstawione podejście będzie w pełni uzasadnione. Warto też zwrócić uwagę, że przesyłanie wiadomości kontrolnych jest krytyczne dla żywotności przedstawionego wyżej algorytmu. Istotnie, w wypadku bowiem, gdyby proces aplikacyjny nie wysłał po chwili τ_s wiadomości przez pewien kanał wyjściowy, stan tego kanału nigdy nie zostałby wyznaczony, a zakończenie konstrukcji obrazu spójnego byłoby całkowicie uzależnione od własności przetwarzania aplikacyjnego.

Dowód poprawności koncepcji

Łatwo jest dowieść poprawności przedstawionej koncepcji konstrukcji obrazu spójnego. Skonstruowany obraz reprezentuje stan globalny, który wystąpił w chwili τ_s. Argumentując poprawność bardziej formalnie, zauważmy, że wyznaczony obraz uwzględnia wszystkie zdarzenia lokalne, które zaszły przed chwilą τ_s. Stąd,

$$(E' \in H_i(\tau_s) \wedge \mathcal{RT}(E) < \mathcal{RT}(E')) \Rightarrow (E \in H_i(\tau_s))$$

gdzie $\mathcal{RT}(E)$ jest czasem rzeczywistym zajścia zdarzenia E.

Ponieważ zegar czasu rzeczywistego spełnia warunek poprawności zegara

$$(E \mapsto E') \Rightarrow \mathcal{RT}(E) < \mathcal{RT}(E'),$$

otrzymujemy

$$(E' \in H_i(\tau_s) \wedge E \mapsto E') \Rightarrow (E \in H_i(\tau_s))$$

a to właśnie należało wykazać. □