

Prof. dr hab inż. Marek Tudruj

Warszawa, 30 września 2011

ul. Raclawicka 8 m 22

02-601 Warszawa

Instytut Podstaw Informatyki PAN

Polsko-Japońska Wyższa Szkoła Technik Komputerowych



RECENZJA ROZPRAWY DOKTORSKIEJ

Tytuł rozprawy: Wirtualizacja rozproszonej pamięci operacyjnej wielokomputera dla systemu Linux w oparciu o koncepcję SDDS.

Autor rozprawy: mgr inż. Arkadiusz Chrobot.

Promotor rozprawy: prof. dr hab. inż. Krzysztof Sapiecha.

1. Cel, zakres i charakter rozprawy

Rozprawa dotyczy bardzo istotnej dziedziny metod budowy wydajnych systemów pamięciowych dla systemów rozproszonych a w szczególności klastrów komputerów i sieci typu Grid. Badania w tej dziedzinie są stymulowane przez wymagania narzucane przez współczesne systemy dostępu do rozproszonych baz danych takie jak np. Oracle Coherence In-Memory Data Grid, w których współdzielone obiekty aplikacji bazodanowej, stanowiące przedmioty zdalnie wykonywanych usług, są rozproszone między pamięciami wielu serwerów z wykorzystaniem redundancji pamiętania danych. Tymi wymaganiami są niski czas odpowiedzi, wysoka przepustowość, skalowalność funkcjonowania względem liczby serwerów, stała dostępność oraz niezawodność dostępu do danych. Narzucającym się rozwiązaniem jest tutaj wirtualizacja dostępu do pamięci rozproszonej, szeroko wspierana przez istniejące produkty np. firmy RNA Networks. Wirtualizacja pamięci rozproszonej powinna być zrealizowana w taki sposób aby programowy system zarządzania pamięcią był łatwo instalowalny na wielu serwerach i był przezroczysty dla aplikacji użytkowych, a więc nie wymagał modyfikowania tych aplikacji. Innym wymaganiami jest aby zastosowany sposób wirtualizacji pamięci rozproszonej był zgodny z tendencjami realizacji zdalnych usług w systemie rozproszonym. Jedną ze znanych technik wirtualizacji pamięci operacyjnej systemów wielokomputerowych są tzw. Skalowalne Rozproszone Struktury Danych (ang. Scalable Distributed Data Structures - SDDS). W tej technice przestrzeń pamięci operacyjnej dostępnej bezpośrednio dla aplikacji w danym komputerze jest zorganizowana za pomocą tzw. wiaderek (analogów ramek w klasycznej wirtualnej pamięci stronicowanej), które zapełniane są tzw. rekordami (analogami stron z pamięci stronicowanej). W znanej w literaturze koncepcji Skalowalnych Rozproszonych Struktur Danych (w odróżnieniu od klasycznej pamięci wirtualnej), sterowanie przesyłaniem informacji między kubelkami a rekordami jest usytuowane na poziomie programu użytkowego w postaci dodatkowej aplikacji SDDS. Aplikacja SDDS pośredniczy między programami aplikacyjnymi a systemem operacyjnym, który efektywnie wykonuje operacje wejścia/wyjścia sprowadzające potrzebne dane do lokalnej pamięci operacyjnej komputera. Takie usytuowanie operacji związanych z zarządzaniem pamięcią wirtualną jest niewygodne, gdyż powoduje konieczność adaptacji programów aplikacyjnych użytkownika dla skorzystania z takiej formy pamięci wirtualnej. Powstała więc koncepcja realizacji pamięci wirtualnej w systemie wielokomputerowym opartej na Skalowalnych

RE CE NZ JA	Biuro Dziekana	
	Wpłynęło dnia	10.10.2011
	Nr	17 / zal.

Rozproszonych Strukturach Danych, których obsługa i sterowanie transmisjami danych jest zlokalizowane w warstwie systemu operacyjnego, prowadząc do bardziej wygodnego korzystania z tego rodzaju pamięci wirtualnej a więc bez potrzeby ingerencji w kod rozproszonych programów użytkowych. Taka koncepcja realizacji rozproszonej pamięci wirtualnej nie była dotąd objęta badaniami, których wyniki są znane w literaturze i jest przedmiotem niniejszej rozprawy doktorskiej. Recenzowana rozprawa stawia sobie za cel wypełnienie tej luki w całokształcie dotychczasowych badań.

Celem recenzowanej rozprawy było zaproponowanie architektury realizacji rozproszonej pamięci wirtualnej opartej na Skalowalnych Rozproszonych Strukturach Danych zrealizowanych częściowo na poziomie systemu operacyjnego, implementacja modelu programowego tej architektury, zbadanie wydajności i własności takiej architektury dla wybranych aplikacji rozproszonych oraz określenie dla jakiej klasy rozproszonych aplikacji użytkowych stosowanie zaproponowanego w rozprawie rozwiązania jest korzystne.

W ramach prac i badań związanych z rozprawą dokonano szerokiej analizy możliwości realizacji praktycznej wymienionej wyżej koncepcji. W jej wyniku wybrano wariant architekuralny, w którym oprogramowanie pamięci wirtualnej opartej na SDDS jest zaimplementowane jako sterownik wirtualnego urządzenia blokowego zarządzanego przez system operacyjny, uzyskując możliwość wykorzystania tej koncepcji dla dowolnego systemu operacyjnego akceptującego urządzenia zewnętrzne z blokową transmisją danych. W ten sposób zaproponowane rozwiązanie ma szeroką stosowalność, gdyż jest zgodne z systemami operacyjnymi typu Unix oraz MS Windows.

Rozprawa ma przede wszystkim charakter praktyczny. Przedstawiono w niej najpierw założenia architekuralne dla zaproponowanego rozwiązania, potem przedstawiono implementację architektury a następnie wyniki badań eksperymentalnych wykonanych przy użyciu prototypu opracowanego środowiska wykonanego dla systemu operacyjnego Linux. W ramach badań eksperymentalnych wykonano testy porównawcze zachowania się prototypu w stosunku do znanych rozwiązań pamięciowych służących dla magazynowania danych, dla programów aplikacyjnych z dziedziny transakcyjnych baz danych, procesów z intensywnymi operacjami wejścia/wyjścia, procesów z intensywnymi obliczeniami oraz podsystemu wymiany stron pamięci.

2. Zawartość rozprawy

Przedstawiona mi do recenzji praca doktorska zawiera 103 strony i składa się ze Wstępu, 5 zasadniczych rozdziałów, wniosków, dodatku związanego z aspektami odporności na błędy oraz bibliografii.

We Wstępie (rozdział 1) nakreślono zakres niniejszej rozprawy na tle ogólnego scharakteryzowania problematyki systemów wielokomputerowych, wirtualizacji pamięci rozproszonej tych systemów, problematyki skalowalnych rozproszonych struktur danych. W tym rozdziale zamieszczono cel i tezę pracy.

W rozdziale 2 omówiono stan wiedzy w dziedzinie wirtualizacji pamięci w systemach wielokomputerowych. Uwzględniono tutaj podział rozwiązań na systemy

jednokomputerowe oraz systemy wielokomputerowe. Wśród rozwiązań dla systemów wielokomputerowych omówiono takie zagadnienia jak wsparcie wirtualizacji dla pamięci operacyjnej systemów rozproszonych, w tym wspomaganie dla optymalizacji wykorzystania tej pamięci, rozwiązania dla rozproszonej pamięci współdzielonej oraz wsparcie dla usług zdalnych. Wśród rozwiązań dla wirtualizacji pamięci operacyjnej omówiono organizację urządzeń wymiany danych oraz organizację rozwiązań RAM dysków. Dla rozproszonej pamięci współdzielonej scharakteryzowano stan wiedzy w dziedzinie rozwiązań sprzętowych, programowych oraz hybrydowych - sprzętowo-programowych. Przedstawiono posumowanie, które podaje oszacowanie istotnych cech Skalowalnych Rozproszonych Struktur Danych w porównaniu z innymi rozwiązaniami. Jednym ze wniosków jest stwierdzenie, że Skalowalne Rozproszone Struktury Danych charakteryzują się wysoką skalowalnością, wydajnością oraz odpornością na błędy. Jednakże ich wykorzystanie w aplikacjach użytkowych wymaga dodatkowych zabiegów polegających na adaptacji kodu programu.

W rozdziale 3 omówiono dokładniej motywacje, które skłoniły Doktoranta do podjęcia badań nad tematem rozprawy. Zamieszczono tam najpierw wnioski z analizy stanu rozwoju dziedziny wirtualizacji pamięci rozproszonej dla wielokomputerów a następnie charakterystykę Skalowalnych Rozproszonych Struktur Danych. W podsumowaniu przedstawiono cel rozprawy z odpowiednią motywacją oraz szkicową charakterystykę treści rozprawy. W rozdziale tym wprowadzono nazwę dla architektury opracowanego rozwiązania pamięci wirtualnej - SDDSfL, która wynika z przyjęcia sposobu reakcji na brak miejsca w pamięci operacyjnej dla zapisu nowego rekordu wymaganego przez aplikację użytkową, nazywanego w skrócie w literaturze algorytmem "haszowania liniowego" - (ang. linear hashing, w skrócie: SDDS LH*). Reakcja na brak miejsca w wiaderku polega na wprowadzenia nowego wiaderka w pamięci z podziałem przepelnionego i migracją połowy jego starej zawartości.

W rozdziale 4 opisano założenia dla architektury SDDSfL. Rozdział ten zawiera 7 podrozdziałów. W pierwszym z nich opisano zasady haszowania liniowego jako elementu SDDSfL. W drugim rozdziale przedstawiona jest analiza potencjalnych metod przeniesienia SDDS na poziom systemu operacyjnego. Podano tu wady i zalety lokalizacji odpowiedniego oprogramowania SDDS na poziomie systemu operacyjnego lub na poziomie programu użytkowego. Aspekty problemu dla architektury SDDSfL omówiono w odniesieniu do trzech podstawowych elementów realizacyjnych SDDSfL: klienta SDDSfL - zajmującego się odbieraniem żądań dostępu do danych, aktywacją obsługi wiaderk i aktualizacją obrazów pliku w komputerach, serwerów SDDSfL - zajmujących się manipulacjami na wiaderkach w tym odczytów/zapisów danych oraz podziałów wiaderk oraz koordynatora podziałów wiaderk - kontrolującego operacje podziałów, w tym wyznaczanie wiaderk do podziału. W rozdziale 4 opisano również szczegółowo rozwiązania algorytmów protokołów operacji SDDSfL na poziomie aplikacji.

Rozdział 5 jest poświęcony opisowi implementacji SDDSfL. W pierwszym podrozdziale opisano implementację klienta SDDSfL. Został on zaimplementowany jako moduł systemu operacyjnego, który może być ładowany/usuwany bez konieczności restartu systemu. Klient SDDSfL udostępnia użytkownikowi wirtualne urządzenie blokowe, które komunikuje się z resztą infrastruktury poprzez sieć. W drugim podrozdziale opisano implementację serwera SDDSfL, który jest procesem użytkownika, realizującym fizyczne operacje dotyczące wiaderk, włączając w to ich podziały i aktualizację stanu pliku aplikacji. Każdy serwer zarządza jednym

wiaderkiem i jest alokowany na odrębnym komputerze systemu wykonawczego. Proces serwera jest zbudowany jako zespół trzech wątków. Kolejny podrozdział opisuje implementację koordynatora podziałów wiaderek. Następny podrozdział opisuje realizację protokołów dla komunikacji danych między strukturalnymi elementami oprogramowania SDDSfL: klient, serwer oraz koordynator podziałów wiaderek. Zastosowanymi protokołami warstwy transportowej są TCP/IP oraz UDP/IP. Ostatni podrozdział opisuje typy błędów, które mogą zajść w działaniu SDDSfL oraz sposoby reakcji systemu i użytkownika.

Rozdział 6 przedstawia badania eksperymentalne wykonane na uruchomionym prototypie SDDSfL. W pierwszym podrozdziale opisano ogólną metodologię testów. W drugim podrozdziale zaprezentowano wyniki przeprowadzonych testów. Wykonano zestaw eksperymentów porównujących wydajność różnych aplikacji z wykorzystaniem infrastruktury SDDSfL oraz innych rozwiązań przechowywania danych jak dyski twarde (SATA, SATA2, PATA, SCSI Ultra-320) i rozproszone systemy plików (NFS, GlusterFS oraz LUSTRE). Przebadane aplikacje obejmowały bazy danych, programy sortowania plików, obliczenia oraz programy systemowe. Wykonane eksperymenty zakładały systemy wielokomputerowe wykorzystujące komputery typu IBM PC połączone lokalnymi sieciami komunikacyjnymi Gigabit Ethernet oraz Infiniband. W olbrzymiej większości eksperymenty wykazały wyższość wykonania aplikacji z wykorzystaniem SDDSfL. Transakcyjna baza danych PostgreSQL wykonywała około 25% więcej operacji w jednostce czasu przy użyciu SDDSfL niż na standardowym dysku SATA. Algorytm sortowania plików QuickSort przy protokole sieciowym Gigabit Ethernet jest lepszy z użyciem SDDSfL od standardowego wykonania z użyciem dysków twardych dla rozmiaru rekordów 2KB ale ustępuje dyskom z protokołem SATA2 dla rozmiaru rekordu 4KB. Sortowanie plików z użyciem SDDSfL przy sieci Gigabit Ethernet w porównaniu z użyciem rozproszonego systemu plików NFS jest nieznacznie lepsze, natomiast nieporównywalnie lepsze niż wykorzystanie systemu plików LUSTRE. Sortowanie plików w systemie z siecią Infiniband jest z użyciem SDDSfL wielokrotnie szybsze niż przy użyciu rozproszonego systemu plików GlusterFS. Stwierdzono, że wykonanie sortowania plików z użyciem SDDSfL praktycznie mało zależy od zastosowanego planisty operacji wejścia/ wyjścia. W kolejnych eksperymentach zbadano przydatność SDDSfL dla wykorzystania wirtualnej pamięci rozproszonej jako urządzenia wymiany danych. Przeprowadzone testy wykazały wyraźną wyższość rozwiązania SDDSfL nad wykorzystaniem dysku z protokołem SATA 2 dla rozmiarów pamięci powyżej 19210 MB i porównywalne zachowanie dla mniejszych rozmiarów pamięci. Wykonanie obliczeń (np. znajdowanie wzorca w pliku tekstowym) przy użyciu SDDSfL okazało się porównywalne z wykonaniami w oparciu o dyski twarde z protokołem SATA 2 oraz systemy plików NFS i GlusterFS. Dalsze testy dotyczyły skalowalności wykonania aplikacji rozproszonych z użyciem środowiska SDDSfL. Skalowalność rozumiana jest jako wzrost wydajności wykonania programów przy rozbudowie infrastruktury systemu wykonawczego. W przeprowadzonych eksperymentach zakładano różne liczby wiaderek i różne ich pojemności. Przy założeniu, że w SDDSfL serwery zarządzają jednym wiaderkiem i są przypisywane odrębnym stacjom roboczym, zmiana liczby wiaderek powoduje zmianę konfiguracji systemu wykonawczego. Wykonano eksperymenty dla algorytmu sortowania plików dla różnej liczby wiaderek i serwerów SDDSfL (od 2 do 8) z siecią Gigabit Ethernet oraz Infiniband dla różnej wielkości rekordów (od 512 KB do 2 GB).

Wszystkie te eksperymenty wykazały mało różniące się czasy wykonania sortowania, przy czym czasy wykonania malały przy wzroście liczby wiaderek. Świadczy to o bardzo dobrej skalowalności działania SDDSfL przy wzroście liczby współpracujących stacji roboczych i braku czułości na wielkość wiaderek i rekordów. Następne eksperymenty badały zachowanie SDDSfL przy wykonaniu sortowania pliku w systemie z siecią Infiniband przy różnej liczbie klientów (w liczbie jeden albo dwa), korzystających z danego pliku SDDSfL, tzn. wtedy gdy wzrastała liczba odwołań do danych. Podobnie jak w poprzednich testach wyniki były pozytywne, co świadczy o tym, że wydajność wykonania aplikacji nie była czuła na badaną liczbę klientów. Wyniki eksperymentów świadczą o tym, że podjęte problemy zostały zbadane w sposób dostateczny.

Rozdział 7 zawiera podsumowanie wyników rozprawy oraz omówienie kierunków dalszych badań.

W Dodatku A przedstawiono dyskusję problemu odporności architektury SDDS na błędy, wykraczającego poza podstawowy zakres rozprawy. Problemy te omówiono w rozbiu na tolerancję błędów sterowania oraz błędów danych..

3. Opinia merytoryczna

Poprawność i oryginalność postawionej tezy.

Praca dotyczy oryginalnego i aktualnego problemu naukowego, gdyż poszukiwania nowych wydajnych rozwiązań dla wirtualizacji pamięci w systemach rozproszonych są zgodne z bieżącymi tendencjami rozwoju takich systemów. Problem naukowy został w rozprawie poprawnie sformułowany. Teza pracy jest sformułowana w sposób jasny i poprawny. Z tezy wynika, że celem pracy jest implementacja nowej metody realizacji pamięci wirtualnej przy częściowym usytuowaniu niezbędnego oprogramowania na poziomie systemu operacyjnego i weryfikacja praktyczna wydajności zaproponowanej metody. Podjęte w rozprawie zagadnienie stanowi istotny element metodologii realizacji obliczeń w systemach rozproszonych.

Czy teza rozprawy została wykazana.

Teza pracy została wykazana. Świadczą o tym opracowany prototyp nowego środowiska pamięci wirtualnej oraz wyniki wykonanych badań eksperymentalnych. Wyniki te pokazują, że zaproponowane rozwiązanie daje nie gorsze a często lepsze wyniki niż standardowe wykorzystanie pamięci dyskowych oraz rozproszonych systemów plików. Dla realizacji eksperymentów wybrano reprezentatywne aplikacje charakteryzujące się zróżnicowanymi wymaganiami jeśli chodzi o cechy wspomagania ze strony środowiska SDDS. W szczególności uzyskane wyniki świadczą o lepszej wydajności środowiska z użyciem SDDSfL dla aplikacji typu bazy danych, sortowanie plików oraz aplikacji z intensywnymi operacjami wejścia/wyjścia oraz nie gorszej wydajności dla aplikacji obliczeniowych.

Analiza źródeł i wiedza autora w danej dyscyplinie naukowej.

Przegląd literatury problemu jest obszerny i został wykonany w sposób wykazujący wysokie kompetencje Doktoranta w dziedzinie rozprawy. Cytowana i wykorzystana

literatura jest bardzo liczna i zawiera 117 pozycje. Uwzględnia ona moim zdaniem wszystkie podstawowe problemy cząstkowe dziedziny pracy.

Niniejsza rozprawa doktorska świadczy o bardzo dobrej wiedzy Doktoranta w zakresie metod realizacji oprogramowania systemowego we współczesnych systemach rozproszonych a w szczególności metod wirtualizacji pamięci w tych systemach. Należy podkreślić pokonanie przez Doktoranta bardzo trudnych problemów i duże nakłady pracy przy opracowaniu architektury systemu i jego implementacji.

Pozycja rozprawy na tle stanu wiedzy w literaturze.

Problemy rozwiązane w rozprawie stanowią istotną część ogólnej problematyki rozwijania wydajnych metod realizacji systemów rozproszonych. W szczególności rozprawa dotyczy bardzo aktualnej obecnie tematyki związanej z wirtualizacją zasobów w systemach rozproszonych. Propozycje wirtualizacji pamięci dla systemów rozproszonych oparte na mechanizmie Skalowalnych Rozproszonych Struktur Danych z częściową realizacją na poziomie systemu operacyjnego stanowią istotny oryginalny wkład Doktoranta w technikę realizacji współczesnych rozproszonych pamięci wirtualnych. Wyniki uzyskane w rozprawie były prezentowane na dwu znanych konferencjach międzynarodowych oraz są opublikowane lub zgłoszone do publikacji w skali międzynarodowej przez wydawnictwa IEEE CS oraz Springer, w tym wkrótce w jednym artykule w czasopiśmie z listy MNiSW.

Znaczenie wyników dla dyscypliny naukowej.

Uzyskane wyniki eksperymentów są pozytywne i świadczą o praktycznej użyteczności zaproponowanych w rozprawie rozwiązań. Na podstawie tych eksperymentów można sądzić, że zaproponowane rozwiązania znajdą zastosowanie w praktyce. W szczególności opracowane rozwiązania są zgodne tendencjami do łączenia w jednym środowisku przetwarzania rozproszonego z usługami zdalnymi, takich rozwiązań systemowych jak rozproszona pamięć wspólna (DSM), rozproszone urządzenia wymiany danych, rozproszone RAM dyski oraz sieci typu "peer-to-peer", o czym świadczą produkty znanych firm software'owych Oracle Coherence oraz RNA Networks. Pomimo praktycznego charakteru rozprawy, przedstawione w niej wyniki mają znaczenie dla rozwoju metod wirtualizacji pamięci w systemach rozproszonych, dostarczając nowych sprawdzonych rozwiązań. Badania nad opracowanym rozwiązaniem powinny być dalej kontynuowane a ich wyniki powinny zostać opublikowane w literaturze światowej.

Umiejętność autora poprawnego przedstawienia wyników

Doktorant rozwiązał postawiony problem używając poprawnych metod. Uzyskane w pracy wyniki są oryginalne. Opracowanych algorytmy i rozwiązania systemowe zostały zaimplementowane w postaci działającego prototypu. Zarówno architektura opracowanego systemu jak i jego implementacja są przedstawione z użyciem właściwych rodzajów reprezentacji. Język rozprawy jest poprawny. Tekst rozprawy jest napisany w sposób przystępny i czyta się dobrze. Tekst jest wspomagany rysunkami a wyniki eksperymentów są podane w tabelach oraz są zobrazowane

graficznie na wykresach. Wszystko to świadczy o bardzo dobrym opanowaniu przez Doktoranta warsztatu naukowego i umiejętności przedstawiania wyników. Moja ocena tego warsztatu jest bardzo wysoka.

Słabe strony rozprawy i uwagi szczegółowe

Nie zauważyłem w rozprawie istotnych błędów merytorycznych.

Pewnego ulepszenia wymaga miejscami układ rozprawy. Teza pracy jest moim zdaniem umieszczona zbyt wcześnie, we Wstępie pracy. Tutaj możnaby zostawić cel rozprawy a tezę przenieść, już po podsumowaniu analizy dziedziny, do Rozdziału 3, który jak się wydaje został wprowadzony właśnie dla tego celu.

Praca jest napisana w olbrzymiej większości bardzo poprawnym i zrozumiałym językiem, jednakże pozostawiono w niej dość liczne błędy pisarskie takie jak np.

str. 48, 9 w.d. wielości -> wielkości

str. 77, 15 w.d. odpowiadającym -> odpowiadające

str. 77, 13 w.d. przydzielony -> przydzielonym

str. 77, 13 w.d. 7 w.d. Co to jest "pojemność serwera SDDSfL" ?

str. 77, 13 w.d., 1 w.d. rozwiązaniach-> rozwiązań

Przeoczono małe błędy logiczne w konstruowaniu tekstu.

str. 50, 22 w.d., str. 51, 11 w.g., str. 51, 2 w.d., powinno być raczej "zawodność" lub "niska niezawodność", inna możliwość to wprowadzenie czegoś na wzór "aspekty niezawodnościowe".

str. 51, 14 w.g., wydajność -> niska wydajność

str. 50, 5 w.d., możliwości -> wady

W opisie rysunku 2.2 brak wygodnego, dodatkowego wyjaśnienia użytych skrótów.

Na rysunku 6.15 wyjaśnienia wymaga umieszczona w ramce notacja - co oznacza SDDSfL, SDDSfL klient nr 1, SDDSfL klient nr 2. W tekście nie są wystarczająco szczegółowo opisane założenia dla tego eksperymentu

Terminologia używana w rozprawie jest miejscami dość specyficzna i wymaga moim zdaniem dodatkowych wyjaśnień. Dodatkowych wyjaśnień wymagają pewne podstawowe terminy używane w rozprawie. O ile pojęcie "wirtualizacja" zostało wyjaśnione w rozprawie z wszelkimi znanymi wariantami semantycznymi, to często używane pojęcie "skalowalności" nie zostało potraktowane w ten sposób. Dotyczy to przede wszystkim użycia w podstawowym dla rozprawy terminie "skalowalne, rozproszone struktury danych" (skrót angielski SDDS), które dodatkowo jako nazwę własną w języku polskim należałoby pisać raczej bez przecinka. Odnośnik do pracy W. Litwina, który jak się wydaje jest twórcą tego terminu oraz wyjaśnienie podane na dole str. 8 nie są dostatecznie wyczerpujące i nie jest w rozprawie wyjaśnione dlaczego SDDS są skalowalne i w jakim sensie. Podobnie, moim zdaniem wyjaśnienia wymaga "skalowalność serwerów" i "skalowalność klientów", patrz rysunki 6.11, 6.12, 6.13, 6.15 na których, zgodnie z powszechnie używanym znaczeniem pojęcia skalowalności, spodziewałem się zobaczyć raczej wykres uzyskiwanego przyspieszenia równoległego w funkcji liczby użytych procesorów.

Termin "pamięć masowa" jest używany (patrz str. 8 i inne) jako synonim "pamięci pomocniczej" czy też "zewnętrznej", lecz raczej ten drugi wariant powinien być używany.

Interesującym problemem jest optymalizacja rozmiaru pliku SDDS LH*, gdy stopień wypełnienia wiaderek rekordami jest niski. W definicji SDDS LH* jest wprowadzona możliwość redukcowania wielkości pliku SDDS w wyniku łączenia wiaderek. Należałoby się zastanowić czy nie należy uzupełnić wprowadzonego mechanizmu wirtualizacji pamięci rozproszonej o odsyłanie rekordów do pamięci pomocniczej systemu, zwłaszcza gdy pojemność pamięci operacyjnej użytych stacji roboczych jest krytyczna. Obecnie stosowana strategia, polegająca na jedynie łączeniu wiaderek, wydaje się bardzo optymistyczna. Ten problem oraz porównanie ze strategią odsyłania stron do pamięci pomocniczej w klasycznej pamięci wirtualnej komputerów mogłoby stanowić ciekawy temat dalszych badań eksperymentalnych, które usasadniałyby wyższość architektury typu SDDS.

Innym ciekawym problemem jest wyposażenie klientów w możliwość współdzielenia danych, co stworzyłoby nowe rozwiązanie dla współdzielonej pamięci rozproszonej. Taka organizacja pamięci rozproszonej, aczkolwiek rodząca problem spójności danych przy ich współdzieleniu, mogłaby być korzystna z punktu widzenia optymalizacji komunikacji danych dla obliczeń rozproszonych, prowadząc być może do redukcji zajętości sieci w sensie liczby wykonywanych transakcji.

Powyższe uwagi nie umniejszają podstawowej wartości merytorycznej niniejszej rozprawy.

4. Wniosek końcowy

Podsumowując stwierdzam, że mgr inż. Arkadiusz Chrobot uzyskał wyniki merytoryczne spełniające w pełni wymagania dla rozpraw doktorskich określone w obowiązującej ustawie o stopniach i tytule naukowym, wykazał się ogólną wiedzą teoretyczną w dziedzinie rozprawy jak również wykazał się bardzo dobrym ogólnym opanowaniem warsztatu naukowego, dlatego wnoszę o dopuszczenie niniejszej rozprawy doktorskiej do publicznej obrony oraz Doktoranta do dalszych faz przewodu doktorskiego.

