

Kamil SZOSTEK, Łukasz KRAUZOWICZ
Adam PIÓRKOWSKI, Paweł OLEKSIK
Akademia Górniczo-Hutnicza
Katedra Geoinformatyki i Informatyki Stosowanej

KONSTRUKCJE WYBRANYCH ŚRODOWISK DEDYKOWANYCH DO OBLICZEŃ ROZPROSZONYCH

Streszczenie. W pracy tej zostały przedstawione dwie technologie do obliczeń rozproszonych – Apache Hadoop oraz XtreamOS. Pierwsza z nich jest szeroko stosowana dla usług sieciowych i internetowych. Druga technologia oferuje możliwość rozsyłania zadań pomiędzy węzły klastra, z uwzględnieniem wymaganych zasobów. Technologie te różnią się budową, co powinno być uwzględnione podczas wyboru dla danego problemu obliczeniowego oraz podczas implementacji.

Słowa kluczowe: obliczenia rozproszone, przetwarzanie rozproszone, klastry

DEDICATED ENVIRONMENTS FOR DISTRIBUTED COMPUTATIONS

Summary. In this paper two technologies for distributed computations are presented – Apache Hadoop and XtreamOS. The first of them is widely used for web services. Its main purpose is an analysis of large data. The second one is developed to distribute tasks between cluster nodes regarding the available resources. This technologies differ in construction context, which should be considered when applied.

Keywords: distributed processing, distributed numerical computations, clusters

1. Wstęp

Czasochłonne obliczenia numeryczne wymagają znacznych mocy obliczeniowych [1]. W ciągu dwóch dekad metodyka postępowania przy rozwiązywaniu tego problemu zmieniała się kilkakrotnie [2-12]. Ostatnimi czasy można było odnieść wrażenie, iż klastry obliczeniowe zostaną wyparte przez obliczenia na kartach graficznych. Tak się jednak nie stało, a co więcej, są tworzone już klastry przetwarzające w jednostkach, wykorzystując procesory GPU.

Warto zatem zwrócić uwagę na możliwości, jakich dostarczają technologie dedykowane do obliczeń rozproszonych.

W niniejszym artykule skupiono się na dwóch ogólnodostępnych technologiach rozproszonych – Apache Hadoop i XtreamOS. Pierwsza z nich jest szeroko stosowana w rozbudowanych serwisach internetowych, takich jak Google, Yahoo! oraz Amazon. Jej głównym przeznaczeniem jest szeroko rozumiana analiza dużej ilości danych, gromadzonych w środowisku rozproszonym. Druga zaś jest stworzona do elastycznego rozdzielania wykonywanych zadań pomiędzy węzły systemu rozproszonego.

Omawiane technologie zostaną przedstawione z uwzględnieniem ich najważniejszych cech oraz możliwości zastosowań do obliczeń numerycznych.

2. Apache Hadoop

Apache Hadoop jest środowiskiem stanowiącym szkielet struktury przeznaczonej do budowania skalowalnych i niezawodnych systemów rozproszonych (w postaci klastra, gdy środowisko sprzętowe jest homogeniczne lub w postaci grida, gdy poszczególne maszyny różnią się od siebie), których głównym celem jest analiza ogromnych ilości danych liczonych nawet w petabajtach. Jest to projekt tworzony na zasadach Open Source pod patronatem Apache Software Foundation. Sortowania przeprowadzone w 2009 roku przez portal Yahoo! – 1 TB w czasie 62 sekund oraz 1 PB w czasie 16,25 godzin – są przykładem wysokiej wydajności tego środowiska.

Hadoop to nie tylko aplikacja, ale również całe zaplecze architektury rozproszonej, które pozwala przeprowadzać obliczenia, czy też różnego rodzaju przetwarzania (m.in. obrazów) w klastrach komputerowych, składających się z ogromnej, sięgającej tysiący, ilości maszyn. Opisywane oprogramowanie zostało stworzone przede wszystkim jako narzędzie do analizy danych.

Hadoop składa się z wielu powiązanych ze sobą podprojektów. Najważniejsze z nich to:

- Hadoop Common – zestaw narzędzi napisanych w Javie, obsługujących rozproszony system plików oraz operacje I/O oraz sama dokumentacja projektu,
- Hadoop Distributed File System (HDFS) – rozproszony system plików używany w klastrach obsługiwanych przez Hadoop, nazywany poprzednio Nutch Distributed File System,
- MapReduce – rozproszony model przetwarzania danych i tworzenia oprogramowania uruchamianego zazwyczaj na klastrach komputerowych.

Inne ciekawe projekty powiązane ze środowiskiem Hadoop to m.in.:

- Avro – system serializacji danych,
- HBase – rozproszona, nierelacyjna baza danych oparta na założeniach BigTable firmy Google,
- ZooKeeper – system koordynowania aplikacji rozproszonych.

2.1. Model MapReduce

Zastosowanie opracowanego przez firmę Google modelu MapReduce jest głównym powodem sukcesu, jaki odniósł projekt Hadoop [13]. Głównym założeniem aplikacji napisanej na podstawie tego modelu jest podzielenie rozwiązywanego problemu na dwie fazy:

- Faza „map” – gdzie węzeł główny (master node) wczytuje dane wejściowe, dzieli je na wiele mniejszych fragmentów i następnie rozsyła je w formie pewnych podproblemów do węzłów liczących czy też nazywanych inaczej zadaniowymi (*worker nodes*), które to przeprowadzają zasadnicze operacje na tychże danych. Oczywiście, istnieje możliwość dalszego podziału danych z zachowaniem struktury ważności poszczególnych węzłów.
- Faza „reduce” – gdzie węzeł główny zbiera wyniki podproblemów wyliczone przez węzły zadaniowe i scala je w pewien sposób, tworząc dane wynikowe głównego problemu, który był rozpatrywany.

W obydwu tych fazach dane wejściowe i wyjściowe dla każdej z nich są przekazywane w formie zbioru par (klucz, wartość) – w dużym uproszczeniu można tutaj znaleźć analogię do tablic asocjacyjnych. Implementacja modelu MapReduce w projekcie Hadoop przyjęła następującą formę:

```
Faza „map”:      (Klucz1, Wartość1) --> lista(Klucz2, Wartość2)
Faza „reduce”: (Klucz2, lista(Wartość2)) --> lista(Klucz3, Wartość3)
```

W założeniu typy danych wejściowych dla fazy „map” (Klucz1 i Wartość1) powinny być odmienne od typów wyjściowych (Klucz2, Wartość2), ale należy pamiętać, że typy wejściowe dla fazy „reduce” winny być takie same jak te wyjściowe w fazie „map”.

Taka realizacja problemu przez MapReduce zapewnia, że dane wejściowe, które otrzymują poszczególne węzły obsługujące fazę „reduce”, są posortowane – jest to element mechanizmu *shuffle and sort*.

2.2. Hadoop Distributed File System

Głównym założeniem projektu Apache Hadoop jest analiza ogromnych ilości danych, dlatego też został opracowany system Hadoop Distributed File System (HDFS) – rozproszony system plików, który w istotny sposób wspomaga przetwarzanie [14]. Istnieje również moż-

liwość integracji z innymi systemami plików, jak np. używanym przez twórców usługi Amazon S3 lub FTP, albo HTTP (tylko do odczytu).

Głównymi założeniami HDFS są:

- obsługa bardzo dużych plików, liczonych w terabajtach,
- komputery zbudowane z łatwo dostępnego i taniego sprzętu,
- strumieniowy dostęp do danych – ważniejszy jest czas dostępu do całości danych niż do pojedynczego rekordu.

Należy jednak pamiętać, że w niektórych przypadkach zastosowanie HDFS będzie powodowało spadek wydajności, nie można więc wykorzystywać go do każdego typu problemu. Na dzień dzisiejszy HDFS będzie miał duże problemy przy zagadnieniach wymagających:

- dużej ilości małych plików – informacje o pliku są przechowywane w pamięci jednego z węzłów,
- szybkich dostępu do plików,
- wielu procesów zapisujących do plików – w HDFS pliki mogą być zapisywane tylko przez jeden proces, a zapisy do systemu plików odbywają się zawsze na końcu pliku.

Dostęp do plików w systemie HDFS jest możliwy wyłącznie przez API Hadoop i tylko przed i po wykonaniu pewnego zadania (*job*), czy też podzadania (*task*).

Zgodnie z modelem zaprezentowanym w Google File System, w HDFS rozmiar tzw. „kęsu”, nazwanego również blokiem, czyli minimalnej wielkości danych, które mogą zostać poddane operacjom I/O, jest stosunkowo wysoki w porównaniu z typowymi systemami plików – domyślnie wynosi 64 MB, gdzie w drugim przypadku waha się on w granicach kilku kilobajtów. Jest to związane z minimalizacją czasu przeszukiwania w celu odnalezienia potrzebnych w danej chwili plików – należy pamiętać, że HDFS jest rozproszonym i sieciowym systemem plików i ten aspekt jest bardzo istotny przy takich rozważaniach, ponieważ narzut transmisji sieciowej w wybranych problemach obliczeniowych potrafi być zauważalny.

2.3. Konfiguracja systemu rozproszonego

Niezależnie, czy celem danego projektu jest stworzenie klastra komputerowego, czy też grida, model zaimplementowany w Apache Hadoop dzieli poszczególne węzły przez pryzmat dwóch warstw – pierwszą z nich jest rozproszony system plików HDFS, a drugą – platforma MapReduce [15].

W pierwszej z nich można wyróżnić następujące węzły:

- Name Node – przechowuje informacje o modyfikacjach HDFS jako plik natywnego systemu plików węzła (*edits*).
- Secondary Name Node – węzeł pełni rolę węzła kopii zapasowej systemu plików, tworzy tzw. punkty kontrolne (*checkpoints*) HDFS.

- Data Node – węzeł przechowujący dane systemu plików HDFS, który na swoim starcie komunikuje się z węzłem Name Node, od którego to otrzymuje rozkazy operacji na HDFS. Klaster zazwyczaj posiada dużo takowych węzłów, które replikują między sobą dane.

Warstwa MapReduce natomiast wprowadza podział na:

- Job Tracker – węzeł nadrzędny, który komunikuje się z właściwą aplikacją włączoną w środowisku Hadoop i przyjmuje od niej główne zadanie, które następnie rozdziela pomiędzy węzły typu Task Tracker (węzłami obsługującymi podzadania), jak również je monitoruje. W przypadku usterki Task Tracker zostaje wyłączony, a jego zadanie jest przekazywane do innego węzła. Job Tracker również monitoruje status zadania oraz odpowiada za komunikację z węzłem Name Node w celu lokalizacji plików w HDFS,
- Task Tracker – jak już zostało wspomniane, węzeł obsługujący podzadania przydzielane przez węzeł nadrzędny. Posiada on pewien zestaw slotów, który determinuje, ile zadań faz „map” oraz „reduce” jest w stanie obsłużyć.

Hadoop przewiduje tworzenie klastra wielowęzłowego z podziałem na węzły typu master i worker (*worker node*). Dla małych klastrów węzeł master pełni rolę węzła: JobTracker, TaskTracker, Name Node oraz Data Node, czyli jednocześnie jest węzłem nadrzędnym i podrzędnym, natomiast pozostałe węzły są typu Task Tracker oraz Data Node. Przy dużej ilości komputerów istnieje możliwość utworzenia osobnych i autonomicznych węzłów pełniących tylko rolę odpowiednio: Name Node, Secondary Name Node oraz Job Tracker.

Ważną cechą projektu Hadoop, której nie można pominąć, jest łatwa konfiguracja klastra i niskie wymagania dotyczące sprzętu, który będzie wchodził w jego skład. Komputer musi umożliwiać uruchomienie systemu GNU/Linux wraz z wirtualną maszyną Java (preferowaną jej wersją jest ta dostarczana przez firmę Oracle, bowiem otwarta implementacja potrafi być problematyczna w niektórych sytuacjach) oraz posiadać kartę sieciową. Cały interfejs Hadoop został stworzony w Javie, jest to projekt wieloplatformowy, istnieje nawet możliwość podłączenia do klastra komputerów, będących pod kontrolą systemu Microsoft Windows, ale równocześnie nie jest wskazane włączanie takowych maszyn do systemu produkcyjnego z powodu różnej stabilności takiego rozwiązania.

Hadoop udostępnia również cały zestaw narzędzi do monitorowania pracy całego systemu rozproszonego – każdy węzeł udostępnia minimalny serwer WWW, który utrzymuje witrynę, dzięki której istnieje możliwość sprawdzenia jego stanu, historii pracy oraz logów systemowych.

2.4. Aplikacja MapReduce w Hadoop

Apache Hadoop, zgodnie z założeniami modelu MapReduce potrafi obsłużyć aplikacje napisane teoretycznie w każdym języku programowania – aplikacja musi implementować funkcję „map” oraz „reduce”. Klaster obsługiwany przez Hadoop udostępnia trzy sposoby do komunikacji z aplikacją:

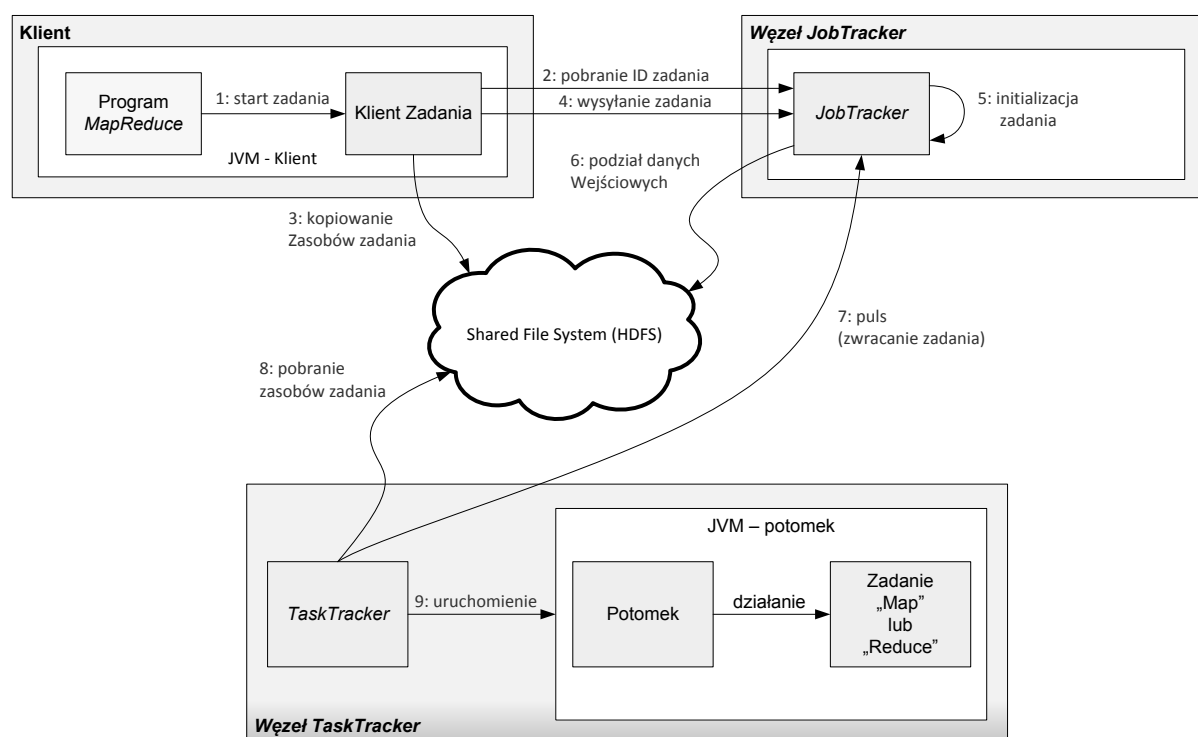
- standardowy – przeznaczony do aplikacji napisanych w Javie,
- potoki – (*pipes*) obsługujące programy napisane w języku C++,
- strumienie – (*streams*) przeznaczone dla aplikacji napisanych w innych językach aniżeli przedstawione powyżej, takich jak np. Python czy C# (przy użyciu platformy Mono).

Ta różnorodność zapewnia pewną elastyczność dla programistów, ponieważ teoretycznie nie uzależnia ich od konkretnej technologii i konkretnego języka programowania. Jednakże, jak pokazują testy, najwydajniejsze są rozwiązania napisane w języku Java, a następnie w C++.

2.4.1. Sposób wykonywania zadania MapReduce przez Apache Hadoop

Na rys. 1 przedstawiono schemat wykonania zadania w środowisku Hadoop.

1. Program jest uruchamiany przez klienta – może to być np. Job Tracker, który jest węzłem nadrzędnym.
2. Klient dostaje numer identyfikacyjny zadania od węzła typu Job Tracker.
3. Klient kopiuje dane do HDFS.
4. Klient potwierdza zadanie.
5. Job Tracker inicjalizuje zadanie (*job*).
6. Job Tracker otrzymuje dane wejściowe przez HDFS, następnie je dzieli i ponownie zwraca do HDFS.



Rys. 1. Schemat idei działania technologii Hadoop

Fig. 1. Hadoop technology schema

7. Task Tracker co pewien czas wysyła do węzła typu Job Tracker „heartbeat” – przez to jest określany jego stan – węzeł jest w pełni sprawny i zdolny do pracy. Gdy Job Tracker ma gotowe podzadania (*tasks*) do wykonania, przekazuje je do węzłów typu Task Tracker.
8. Następnie poszczególne węzły typu Task Tracker pobierają dane (już podzielone przez węzeł Job Tracker) z HDFS.
9. Task Tracker uruchamia podzadanie (*task*).
10. Na każdym węzle typu Task Tracker, w osobnej instancji, są uruchamiane podzadania, czyli jego fazy „map” i „reduce”, zgodnie z dostępną ilością slotów.
11. Po zakończeniu pracy Task Tracker zwraca wynik do HDFS.

2.5. Zastosowanie praktyczne

Klastry stworzone za pomocą Hadoop (i niektórych jego podprojektów, jak np. HBase) mogą być wykorzystywane do rozwiązywania wielu różnych problemów – od gromadzenia danych (Adobe, Facebook), poprzez optymalizację przeszukiwań (eBay), konwersję dużych ilości obrazów o wysokiej rozdzielczości (The New York Times), uczenie maszynowe (ang. *machine learning*), aż po tworzenie komercyjnych serwisów typu Cloud Computing (Amazon). Niektóre z nich śmiało można nazwać nawet superkomputerami (Yahoo! – 4500 węzłów).

Należy jednak pamiętać, że klaster taki nie jest typowy – owszem, może rozwiązywać wszelakie algorytmy, ale w pewnych zagadnieniach (np. numerycznych) w porównaniu z sys-

temami rozproszonymi obsługiwanymi przez np. MPI, jego wydajność może być wysoce niesatysfakcjonująca. Hadoop został stworzony przede wszystkim do analizy, ale również przechowywania dużych ilości danych, liczonych nawet w dziesiątkach czy setkach petabajtów.

3. XtreamOS

System XtreamOS jest finansowanym przez Komisję Europejską projektem, udostępnianym na licencji *General Public License* w wersji drugiej (GPLv2) oraz *Berkeley Software Distribution* (BSD). Jego celem jest stworzenie kompletnego, opartego na rozwiązaniach klastrowych, wieloplatformowego systemu operacyjnego, który obsługuje Organizacje Wirtualne (VO).

XtreamOS jest otwartym systemem operacyjnym bazującym na systemie Linux. Jego przeznaczeniem są systemy rozproszone, zarówno lokalne (klastry), jak i zdalne (np. poprzez sieć Internet). XtreamOS zapewnia użytkownikom transparentny dostęp do sprzętowych i programowych zasobów komputerów będących w sieci. Uruchomienie przez użytkownika systemu aplikacji (również interaktywnych) powoduje automatyczne wyszukanie i przydzielenie zasobów dla danego zadania. Możliwe jest również wykorzystanie zasobów poszczególnych komputerów klastra jako zasoby jednego systemu, w którym przykładowo sumowana jest dostępna w klastrze pamięć operacyjna. Ponadto, system ten dostarcza rozbudowanych narzędzi administracji prawami dostępu użytkowników do zasobów poprzez wykorzystanie Organizacji Wirtualnych [16].

XtreamOS zapewnia dobrą skalowalność dzięki stosunkowo nieskomplikowanej metodzie dodawania nowych węzłów oraz użytkowników.

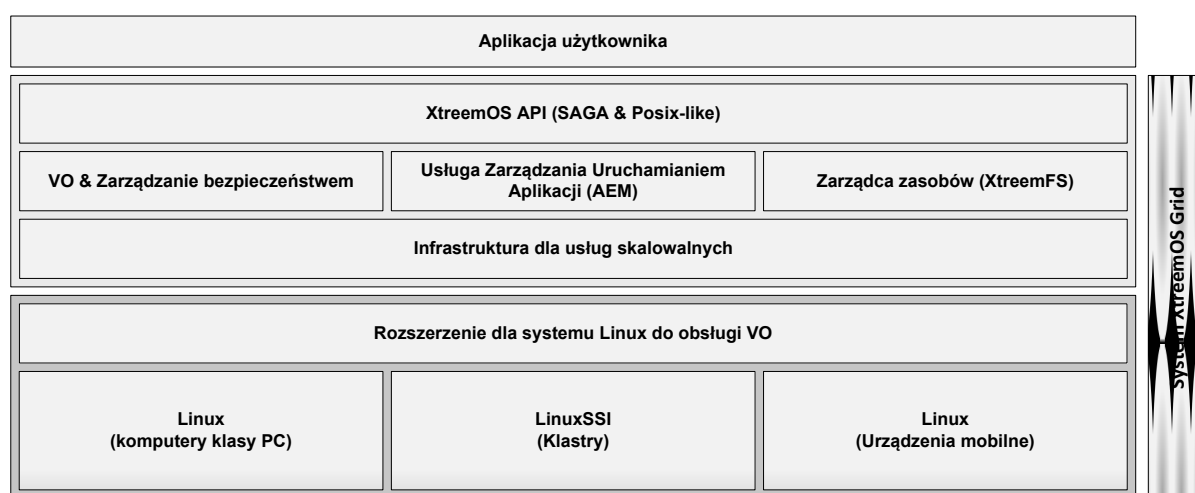
3.1. Zastosowane technologie

W projekcie systemu XtreamOS wykorzystano wiele technologii, które pozwalają wygodnie korzystać z zasobów maszyn tworzących klastry (rys. 2).

Pierwszą z nich jest natywnie obsługiwana w tym systemie usługa Zarządzania Organizacjami Wirtualnymi. Pozwalają one tworzyć i zarządzać Organizacjami Wirtualnymi oraz innymi usługami zabezpieczeń, takimi jak Jednostka Zarządzania Certyfikatami (*Resource Certification Authority* – RCA). Mają one na celu konstruowanie reguł dostępu do sprzętu i usług uruchomionych w systemie XtreamOS [17].

Kolejną istotną usługą jest Zarządzania Uruchamianiem Aplikacji (*Application Execution Management* – AEM), która między innymi pozwala na monitorowanie dostępnych zasobów sprzętowych i programowych w klastrze, automatyczne kolejkowanie zadań oraz ich migrację.

Ostatnią, niemniej niezwykle ważną usługą jest XtreamFS, czyli system zarządzania danymi. Jest to rozproszony system plików, zaprojektowany dla rozwiązań klastrowych i sieci rozległych. Obsługuje replikację oraz fragmentację danych, które to procesy są automatycznie optymalizowane dzięki wbudowanym mechanizmom monitorującym. Dodatkowo XtreamFS jest zintegrowany z VO w celu zapewnienia autoryzacji dostępu do danych.



Rys .2. Konstrukcja systemu XtreamOS z podziałem na podstawowe usługi
Fig. 2. A schema of XtreamOS construction

XtreemFS jest transparentnym systemem plików, co pozwala zalogowanemu użytkownikowi na swobodny dostęp do danych rozproszonych w klastrze.

3.2. Konfiguracja systemu XtreamOS

Konfiguracja klastra działającego na podstawie systemu XtreamOS nie należy do skomplikowanych, co niewątpliwie zaliczyć należy do zalet tego systemu. Każdy komputer, który będzie należał do klastra, musi posiadać zainstalowane odpowiednie usługi pobrane bezpośrednio ze strony *xtreemos.eu* lub przygotowane pakiety z repozytoriów dla wybranych systemów operacyjnych. Obecnie dostępne są pakiety tylko dla systemu OpenSUSE. Następnie jeden wybrany komputer należy skonfigurować jako master, co sprowadza się do opisania struktury i określenia lokalizacji sieciowej pozostałych komputerów. W kolejnym etapie jest wymagane wygenerowanie odpowiednich certyfikatów oraz stworzenie kont użytkownikom, którzy będą mogli korzystać z klastra. Pojawia się tu szerokie spektrum możliwości konfiguracji, poczynając od rozproszenia wspomnianych usług, kończąc na dostosowaniu praw dostępowych dla użytkowników systemu.

Po wykonaniu niezbędnej konfiguracji należy przekopiować katalog z ustawieniami na inne komputery w klastrze i uruchomić na każdym komputerze przygotowany przez twórców systemu skrypt automatycznej konfiguracji. Wadą takiego rozwiązania jest konieczność wy-

konania tych czynności za każdym razem na wszystkich węzłach, gdy zaistnieje potrzeba zmiany ich liczby.

3.3. Zadania w systemie XtreamOS

W systemie XtreamOS, dzięki AEM, uruchamianie zadań jest intuicyjne i elastyczne. Czynność ta sprowadza się do stworzenia wg specyfikacji pliku XML, który definiuje zadanie, czyli między innymi ścieżkę do programu, pliki wynikowe, wymagane zasoby oraz parametry przekazywane do programu. Następnie zadanie jest dodawane do kolejki uruchomieniowej za pomocą programu *xsub*. System XtreamOS wyszukuje wymagane zasoby i uruchamia proces w swoim wirtualnym środowisku. Dane wejściowe muszą być umieszczone w wirtualnej przestrzeni danych XtreamFS. Zadania mogą być uruchamiane również w sposób interaktywny, poprzez przekazanie odpowiednich argumentów do programu *xsub* [18].

Ponadto, jest możliwa prezentacja szczegółów aktywnych zadań, oczekiwanie na zakończenie wybranego procesu, a nawet migracja do innego zbioru zasobów. Dodatkowo zaimplementowana jest obsługa zdarzeń, która może służyć np. do zakończenia wybranego procesu.

3.4. Możliwości aplikacyjne

System XtreamOS, dzięki swojej budowie i zaimplementowanym rozwiązaniom pozwala w wygodny sposób korzystać z zasobów klastra. Aplikacje użytkownika mogą być uruchamiane w sposób całkowicie transparentny, a automatyczne ich rozdzielenie w zależności od dostępnych zasobów pozwala na dodanie do kolejki wielu instancji tego samego programu z różnymi parametrami wejściowymi i ich równoczesne wykonanie. Może być to wykorzystane np. przy modelowaniu zjawisk fizycznych, czy też rozwiązywaniu zagadnień metodą Monte Carlo.

Kolejnym zastosowaniem systemu XtreamOS może być konstrukcja serwisów usług internetowych, do których działania są wymagane duże moce obliczeniowe, jak również pożądanym jest automatyczny i dynamiczny przydział zasobów klastra. Dzięki aplikacji WS-Escape, bazującej na systemie XtreamOS, aplikacje sieciowe mogą być rozdzielane w klastrze w zależności od jego obciążenia, natomiast usługa XtreamFS może skutecznie zajmować się sieciowym zarządzaniem danych dla tych aplikacji [16].

Inną ciekawą cechą systemu XtreamOS jest możliwość uruchamiania aplikacji interaktywnych, takich jak trójwymiarowe programy konstrukcyjne, edukacyjne czy gry sieciowe.

4. Podsumowanie

W pracy tej przedstawiono dwie technologie służące do obliczeń rozproszonych: Apache Hadoop oraz XtreamOS. Środowiska te pozwalają użytkownikowi wykorzystać dostępne moce obliczeniowe klastrów przy stosunkowo nieskomplikowanej obsłudze wspomnianych systemów.

Zarówno Hadoop, jak i XtreamOS charakteryzują się wysoką wydajnością oraz szeregiem mechanizmów, które pozwalają w bezpieczny sposób przeprowadzać operacje na danych w klastrze. Hadoop oferuje użytkownikom dostęp do modelu MapReduce, dzięki któremu analiza nawet dużej ilości danych, liczonych w terabajtach, może być przeprowadzana w sposób wydajny. XtreamOS zaś udostępnia transparentny system rozproszony, umożliwiając użytkownikom elastyczną pracę na podstawie zasobów dowolnego komputera w klastrze.

Możliwości aplikacyjne obydwu systemów sprawiają, że są to technologie o niezwykle dużym potencjale i wykorzystanie ich nawet w niewielkich klastrach może znacznie przyspieszyć prowadzone analizy lub obliczenia.

Praca ta wiąże się z profilem naukowym jednostki Autorów. Geoinformatyka bowiem, oprócz części związanej z geomatyką, obejmuje zagadnienia geologii i geofizyki, a te ostatnie są źródłem zapotrzebowań na wysokie moce obliczeniowe.

Uzupełnieniem niniejszej pracy jest artykuł [19], przedstawiający możliwości zastosowania i testy wydajnościowe technologii Hadoop do obliczeń rozproszonych dla wybranego zagadnienia numerycznego z dziedziny nauk o ziemi.

BIBLIOGRAFIA

1. Kozielski S.: Klastry jako źródła mocy obliczeniowej w sieciach komputerowych [w]: Węgrzyn S., Pochopień B., Czachórski T. (red.): Współczesne problemy sieci komputerowych. Nowe technologie. Wydawnictwa Naukowo-Techniczne, Warszawa 2004, s. 221÷230.
2. Gruca A.: Programowanie w systemach rozproszonych [w:] Węgrzyn S., Znamirowski L., Czachórski T., Kozielski S. (red.): Nowe technologie sieci komputerowych. T. 1, Wydawnictwa Komunikacji i Łączności, Warszawa 2006, s. 479÷488
3. Kruk T., J., Mazurowski M.: Praktyczne aspekty równoważenia obciążenia w rozproszonych systemach operacyjnych. Konferencja Sieci Komputerowe 2002, Studia Informatica, Nr 50, Gliwice 2002, s. 35÷44.

4. Bylina J., Bylina B., Czachórski T.: Rozproszona aplikacja do rozwiązywania markowskich modeli mechanizmów sieciowych. Konferencja Sieci Komputerowe 2006, [w:] Węgrzyn S., Znamirowski L., Czachórski T., Kozielski S. (red.): Nowe technologie sieci komputerowych. T. 1, Wydawnictwa Komunikacji i Łączności, Warszawa 2006, s. 331÷338.
5. Onderka Z., Cichy M.: Porównanie wydajności komunikacji dla standardów DCOM i CORBA w systemach typu klient-serwer. Konferencja Sieci Komputerowe 2011 (SK 11), *Studia Informatica*, vol. 32, no. 3A (98), Gliwice 2011.
6. Stanko W., Trawiński B.: Badanie technologii Microsoft .NET Remoting w rozproszonym systemie wspomagającym publikowanie katalogów reklamowych [w:] Nowe technologie sieci komputerowych, Wydawnictwa Komunikacji i Łączności, Warszawa 2006.
7. Piórkowski A.: Metody tworzenia rozproszonych aplikacji bazodanowych w technologii .NET [w:] *Sieci komputerowe: Aplikacje i zastosowania*, WKŁ, Warszawa, 2007, tom 2, s. 195÷202.
8. Berlińska J., Stipczyński P.: Zastosowanie metod statystycznych do oceny wydajności aplikacji rozproszonych. Konferencja Sieci Komputerowe 2004, [w:] Kwiecień A., Grzywak A. (red.): Współczesne problemy sieci komputerowych. Zastosowanie i bezpieczeństwo, Wydawnictwa Naukowo-Techniczne, Warszawa 2004, XVII, s. 145÷152.
9. Kocoń M.: Przetwarzanie rozproszone w sieci Internet. Konferencja Sieci Komputerowe 2003, *Studia Informatica*, Nr 55, s. 319÷328.
10. Augustyn D. R., Wyciślik Ł.: Technologia Web Services i .NET w rozproszonej realizacji zadania optymalizacji parametrycznej w modelowaniu ciągłych układów dynamicznych. Konferencja Sieci Komputerowe 2004, [w:] Kwiecień A., Grzywak A. (red.): Współczesne problemy sieci komputerowych. Zastosowanie i bezpieczeństwo, Wydawnictwa Naukowo-Techniczne, Warszawa 2004, XXIX, s. 251÷256.
11. Juszczak-Januszewska J.: Aspekty przetwarzania danych biznesowych w siatce obliczeniowej w systemach baz danych. Konferencja Sieci Komputerowe 2005, [w:] Kwiecień A., Grzywak A. (red.): Wysokowydajne sieci komputerowe. Zastosowanie i bezpieczeństwo, Wydawnictwa Komunikacji i Łączności, Warszawa 2005, s. 435÷444.
12. Golak S., Grochowicz W.: Zastosowanie publicznych zasobów obliczeniowych do uczenia sieci neuronowych. Konferencja Sieci Komputerowe 2006, [w:] Węgrzyn S., Znamirowski L., Czachórski T., Kozielski S. (red.): Nowe technologie sieci komputerowych. T. 1, Wydawnictwa Komunikacji i Łączności, Warszawa 2006, s. 277÷286.
13. Lam Ch.: *Hadoop in Action, Anatomy of MapReduce program*. Manning Publications Co., 2011

14. White T.: Hadoop The Definitive Guide 2nd Edition, The Hadoop Distributed Filesystem. O'Reilly Media, Inc., 2010.
15. White T.: Hadoop The Definitive Guide 2nd Edition, Hadoop Configuration. O'Reilly Media, Inc., 2010.
16. XtreamOS : A Linux-based Operating System to support Virtual Organizations for next generation Grids. Strona domowa projektu, [[:@:]] <http://www.xtreemos.eu/>
17. Third Prototype Implementation of Security and VO Management Services -D3.5.16 [[:@:]] <http://www.xtreemos.eu/project/publications/project-deliverables/D3.5.16.pdf>
18. Requirements and specification of XtreamOS services for job execution management D3.3.1 [[:@:]] <http://www.xtreemos.eu/project/publications/project-deliverables/d3-3-1.pdf>
19. Krauzowicz Ł., Szostek K., Dwornik M., Oleksik P., Piórkowski A.: Numerical Calculations for Geophysics Inversion Problem Using Apache Hadoop Technology. Computer Networks 2012, CCIS vol. 291, Springer, 2012, s. 439-446.

Wpłynęło do Redakcji 14 marca 2012 r.

Abstract

In this work two cluster technologies were presented – Apache Hadoop and XtreamOS. As the cluster solution provides high computational power for data analysis or numerical calculations, selected technologies help to take full advantage of this approach.

The first of them – Apache Hadoop – is widely used by Internet companies, like Microsoft or Yahoo!, to process large amount of data, counted in terabytes. This software provides user with a flexible framework to write and run MapReduce tasks, written in one of the popular language, like Java or C++. Nevertheless, tasks written in Java are reported to be the fastest, as it is Hadoop's native language.

One of the main services running under Hadoop environment is Hadoop Distributed File System (HDFS), designed to operate on distributed data in secure and reliable way. This file system is perfect solution for large amount of data, but it suffers significantly when working on numerous small files.

The second technology described in this paper is XtreamOS, an open source operating system running on top of the Linux system. Its main idea is to provide cluster's users a simple to use and reliable environment to execute their applications that require specific resources available in cluster. Thanks to the high transparency of the XtreamOS system, users can take advantage of the cluster resources almost as they were using one machine or submit their jobs

directly to the most suitable node. Implemented mechanisms in this system provide automatic and dynamic load balancing, data replications and fault protection. What is more, XtreamOS natively supports Virtual Organizations with set of tools to manage users and their permissions in the system.

The presented cluster solutions provide highly efficient tools to run and manage distributed computations. Each has a specific purpose and should be chosen according to the problem, in the context of its algorithm and character of data.

Adresy

Kamil SZOSTEK: Akademia Górniczo-Hutnicza, Katedra Geoinformatyki i Informatyki Stosowanej, Al. Mickiewicza 30, 30-059, Kraków, Polska, szostek@geol.agh.edu.pl

Łukasz KRAUZOWICZ: Akademia Górniczo-Hutnicza, Katedra Geoinformatyki i Informatyki Stosowanej, Al. Mickiewicza 30, 30-059, Kraków, Polska

Adam PIÓRKOWSKI: Akademia Górniczo-Hutnicza, Katedra Geoinformatyki i Informatyki Stosowanej, Al. Mickiewicza 30, 30-059, Kraków, Polska, pioro@agh.edu.pl

Paweł OLEKSIK: Akademia Górniczo-Hutnicza, Katedra Geoinformatyki i Informatyki Stosowanej, Al. Mickiewicza 30, 30-059, Kraków, Polska, oleksik@agh.edu.pl