

УДК 004.7(045)

Кременецький Г.М., канд. техн. наук

ПОБУДОВА ДИНАМІЧНОЇ КЛАСТЕРНОЇ НЕЙРОННОЇ МЕРЕЖІ З ВИКОРИСТАННЯМ WEB-СЕРВІСІВ

Інститут комп'ютерних технологій
Національного авіаційного університету

Запропоновано використання WEB-сервісів для побудови динамічної кластерної нейронної мережі. Проведено аналіз особливостей WEB-сервісів. Розглянуто їх позитивні та негативні боки щодо використання

Вступ

Тема розподіленої обробки даних є дуже поширеною, як у наукових колах, так і технічних спеціалістів. Вона є дуже перспективною, тому що достатньо простими засобами обіцяє підвищувати швидкість різних програм. Спочатку вважається, що скільки обчислювальних вузлів ми використаємо у стільки разів підвищиться швидкість обробки даних. Але професіонали, які займаються проблемами розподіленої обробки знають що це зовсім не так просто. У деяких випадках використання додаткових обчислювальних вузлів призводить до зниження швидкості обробки.

Постановка задачі

Побудова архітектури кластерних нейронних мереж стикається з проблемами передачі даних у такій гетерогенній мережі як Інтернет. Багата кількість різних локальних мереж, які використовують різні політики безпеки, різні швидкості передавання даних всі ці перешкоди можна подолати використовуючи новітню технологію Web-сервісів. Технологія Web-сервісів створена для розгортання розподілених обчислювальних систем у мережі Інтернет. Основна причина її появи – нездатність існуючих технологій, таких як об'єктні системи типу COM сімейства Microsoft і стандарти OMG CORBA, повною мірою забезпечити сумісність різних програмних продуктів для неоднорідних розподілених систем. Технологія Web сервісів запропонована і розвивається під егідою WWW-консорціуму W3C

провідних компаній-виробників програмного забезпечення.

Основу технології Web-сервісів складають:

- найпростіші комунікаційні Інтернет-протоколи HTTP і/або SMTP;
- протоколи SOAP (Simple Object Access Protocol) для управління повідомленнями в універсальному XML-форматі;
- мова WSDL (Web Services Definition Language) описи інтерфейсу взаємодії компонентів розподіленої системи.

Технології послуг мережі SOAP і WSDL застосовувалися і раніше (Userland, Microsoft, Developmentor); їх специфікації були використані для Web сервісів як основа для створення розширюваної структури управління повідомленнями (messaging) SOAP 1.2 і мови визначень інтерфейсів WSDL 1.2.

Можна дати визначення Web сервісів (або Web-служб) як набору послуг у вигляді програмних додатків, ідентифікованого мережною адресою URI (Uniform Resource Identifier), інтерфейси і зв'язки (binding) якого визначаються XML-засобами. Web-сервіси забезпечують прямі взаємодії через Інтернет з іншими агентами програмного забезпечення, використовуючи повідомлення, засновані також на XML-форматі. Дане визначення Web-сервісів припускає використання SOAP як формату або моделі обробки повідомлень. І при цьому воно припускає також використання WSDL як мови описів обслуговування. В даний час існують безліч послуг мережі, які використовують

HTTP як протокол передачі даних і деякі взаємоузгоджені XML-формати змісту повідомлення (контента).

Консорціум W3C розробляє рекомендації, що є стандартом для специфікацій і протоколів Web-сервісів; згідно цим рекомендаціям різні компанії-виробники програмного забезпечення розробляють власні реалізації SOAP і WSDL технологій. Ось деякі з них: *SOAP Toolkit (Microsoft)*, *WebSphere Application Server (IBM)*, *JWSDP (Sun)*. Одним з досягнень технології Web-сервісів є сумісність всіх їх реалізацій, незалежно від постачальників (провайдерів) обчислювальних послуг. Основу цієї сумісності є послідовне застосування на всіх рівнях надання послуг Web-сервісів стандартів XML-технології. Так наприклад, формат SOAP-повідомлень – основна одиниця передачі даних – представлений у вигляді XML-документа; опис інтерфейсу обчислювального сервісу в WSDL також представляється в XML-форматі.

В архітектурі Web-сервісів розглядаються дві її складові частини: базова (*Basic Architecture*) і розширена (*Extended Web Services Architecture*) архітектура. Базова архітектура включає стандартний (обов'язковий) набір засобів цієї технології, який вже знайшов застосування в різних реалізаціях Web-сервісів; розширена архітектура в більшій мірі орієнтована на додаткові або перспективні специфікації, розширювальні можливості технології наступні:

- асинхронні повідомлення;
- передача даних в режимі додатків (*attachment*) SOAP-повідомлень;
- ідентифікація і конфіденційність повідомлень і ін.

Базова архітектура включає наступні технології Web-сервісів:

- обмін повідомленнями;
- забезпечення віддаленого виклику процедур;
- опис послуг Web-служб;
- реєстрація (публікація) і пошук (виявлення) описів послуг.

Базова архітектура Web-сервісів визначає взаємодію між агентами програмного забезпечення як обмін повідомленнями між тими що запрошують сервіс (*requesters*) і постачальниками послуг (*provider* – провайдерами). *Requesters* – агенти програмного забезпечення, які запрошують виконання сервісу, провайдери – агенти програмного забезпечення, які забезпечують виконання сервісу. Кожний з агентів може бути одночасно і агентом і провайдером. Провайдери відповідальні за реалізацію сервісу і публікацію його опису, агенти – повинні мати спосіб пошуку опису послуг. Отже, базова архітектура Web-сервісів виявляється у виконанні трьох ролей: провайдера сервісу, сервісу реєстрації і сервісу пошуку клієнта, що запрошує сервіс. Взаємодія включає операції декларації (публікації), пошуку і скріплення (*bind*) сервісу. Ці ролі і операції є для Web-сервісів артефактами (*artifacts*): модулями програмного забезпечення і їх описами. В типовому сценарії Web-сервісів провайдер сервісу містить програмні модулі з мережним доступом, реалізуючи сервіс, він же формує опис сервісу і його декларацію (публікацію), доступну клієнту-споживачу або сервісу реєстрації (*service discovery agency*).

Клієнт-споживач виконує операції пошуку для отримання опису сервісу локально або через сервіс реєстрації і, використовуючи цей опис, виконує процедуру скріплення з провайдером сервісу, ініціює взаємодію з модулями, що реалізують даний сервіс.

На рис. 1 зображена схема взаємодії провайдера сервісу, сервісу реєстрації і клієнта-споживача сервісу.

Як вище зазначалось, основною одиницею обміну в технології Web-сервісів є повідомлення, що мають структуру XML-документа. Описи сервісу використовують також XML-нотацію, що містить всі необхідні деталі для взаємодії в мережі, включаючи опис форматів повідомлення, транспортних протоколів і його локалізацію. Інакше кажучи, формування і транспортування

XML-повідомлень підкоряються правилам і специфікаціям *SOAP*-протоколу; провайдер публікує *WSDL*-файл, що містить опис семантики і синтаксису повідомлення, кінцевої адреси сервісу (*endpoint*), що дає можливість клієнту-споживачу правильно згенерувати *SOAP*-повідомлення і відправити його в правильному напрямі. Сервіс-провайдер одержує це повідомлення, оброблює запит, який міститься в повідомленні і посилає результат клієнту також у вигляді *SOAP*-повідомлення. Технологія, типова для такого типу взаємодії *Web*-сервісів, включає *SOAP*, *WSDL* і *HTTP*.

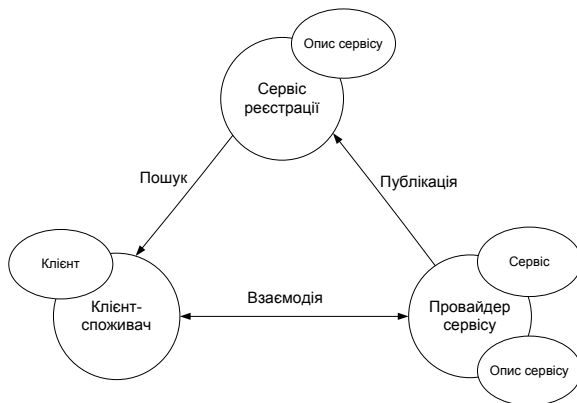


Рис. 1. Схема взаємодії провайдера сервісу, сервісу реєстрації і клієнта-споживача сервісу

Простий протокол доступу до об'єктів (*SOAP*) – заснований на *XML*-механізмі, спосіб створення структурованих пакетів даних для обмінів між мережними додатками. *SOAP* містить чотири основні компоненти:

- конверт (*envelope*), визначаючий рамкову структуру повідомлення у форматі *XML*;
- набір правил для представлення типів даних;
- домовленість про представлення виклику віддалених процедур (в режимі *RPC*);
- правила сумісного виконання протоколів *SOAP* і *HTTP*. *SOAP* може використовувати також комбінацію різних мережних протоколів, таких як *HTTP*, *SMTP*, *FTP*, *RMI/IIOP*.

Відправник, і отримувач *SOAP*-повідомлення повинен мати доступ до

одного і того ж опису сервісу. Відправник потребує опису сервісу, щоб знати, як правильно формувати повідомлення, а приймач – для правильної його інтерпретації. Файл *WSDL* формується у форматі *XML*-документа самим відправником або провайдером сервісу і містить всю необхідну для потенційних споживачів сервісу інформацію у формалізованому вигляді. Іншими словами, знання мережної адреси *WSDL*-файлу в Інтернеті, є достатньою основою для розуміння і правильної побудови запитів від клієнта-споживача сервісу.

Опис сервісів являється *XML*-документом, що складається з декількох елементів, у тому числі з опису простору імен (*namespace*), опису типів і елементів повідомлень, порту, а також можливих операцій (для *RPC*) – запитів і відповідей. *WSDL*-файл, що містить опис сервісів, є достатньо складним документом, тому для його створіння і аналізу, як правило, користуються автоматичними генераторами, включеними до складу засобів розробки.

Сервіс реєстрації – репозитарій використовується для публікації провайдером сервісу в тому або іншому форматі (вільному або формалізованому вигляді) інформації про послуги. Клієнт-споживач послуг за різними критеріями може знаходити і досліджувати їх якість і кількість і, після виконання процедури з'єднання (*binding*), безпосередньо звертатися до постачальника для виконання декларованого сервісу. Для пошуку і аналізу сервісу необов'язково мати вищеописану службу. Вся необхідна інформація для підключення до сервісу може передаватися провайдером споживачу конфіденційним шляхом. Консорціум *W3C* не займається виробленням рекомендацій (стандартів) для інформаційного обслуговування постачальників (провайдерів) сервісів. Для цього існують в електронному бізнесі системи типу *ebXML* або *UDDI*.

Відзначу декілька реалізацій реєстраційних служб:

UDDI (Universal Description, Discovery and Integration) є засобом реєстрації обчислювальних або комерційних послуг у формалізованому XML-форматі. *UDDI* можна розглядати як логічну структуру, загальноприйнятую стандартну класифікацію, що має:

Жовті Сторінки, в яких реєструються під різними категоріями ділові пропозиції і послуги;

Білі Сторінки, які містять контактну інформацію;

Зелені Сторінки, що містять в основному технічні деталі і використовуються для звернення до сервісу.

UDDI надає свої послуги також у форматі *Web-сервісів*, і до нього можна звертатися за отриманням сервісу стандартними засобами, наприклад *SOAP*-повідомленнями.

ebXML (Electronic Business using eXtensible Markup Language) призначений для електронного бізнесу. Він надає засоби реєстрації, пошуку і аналізу сервісних послуг у форматах *CPP (Collaboration-Protocol Profile)* і *CPA (Collaboration-Protocol Agreement)* в XML-представленнях. На відміну від *UDDI*, де інформація точно структурована і формалізована, тобто містить тільки метадані про сервіс, *ebXML* допускає включення в реєстрацію окрім метаданих додаткову інформацію довільної структури. Простий приклад: файл *WSDL*, такий необхідний для клієнта-споживача сервісу, не може включатися безпосередньо в *UDDI*-описи (тут можливе лише посилання на цей файл на стороні провайдера послуг), *ebXML* допускає включення в реєстрацію будь-яких технічних деталей, що стосуються декларованого сервісу.

JAXR (Java API for XML Registries) – є складовою частиною вільно поширюваного пакету *JWSDP* компанії *Sun*, в якому реалізована підтримка *Web-сервісів*. Треба підкреслити особливості цієї служби реєстрації:

– користувач може самостійно встановлювати сервер реєстрації *JAXR* і відповідну базу даних;

– інформаційна модель *JAXR* заснована на *RIM*-моделі реєструючої системи *ebXML (RIM, Registry Information Model)* з розширеною підтримкою *UDDI*.

Таким чином, *JAXR* є сумісним з обома моделями *UDDI* і *ebXML*.

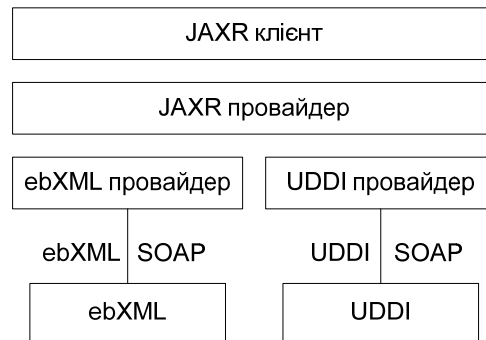


Рис. 2. Архітектура верхнього рівня *JAXR*

Обмін даними між клієнтом-споживачем і провайдером-постачальником послуг – це передача *SOAP*-повідомлень. Протокол *SOAP* пропонує спосіб формування повідомлень в XML-форматі, заснований на об'єктній моделі XML-документа *DOM (Document Object Model)*. *SOAP*-повідомлення – це ієрархічна структура вкладених XML-елементів (або вузлів): *SOAPMessage*, *SOAPPart*, *SOAPEnvelope*, *SOAPHeader*, *SOAPBody* і вміст передаваного повідомлення, також у форматі XML (*XML content*). На рис. 3 представлена ієрархічна структура *SOAP*-повідомлення.

Розробники серверної і клієнтської частин *Web-сервісів* можуть використовувати два типи надання і споживання обчислювальних послуг:

– передача даних (запит і відповідь) у форматі *SOAP*-повідомлень. Причому, формування XML-структури повідомлення, його кодування і обробка результату виконуються “вручну” прикладними програмами з використанням об'єктної моделі *DOM* і *SOAP*-протоколу;

– використання механізму виклику віддалених процедур *RPC (Remote Procedure Call)* для передачі даних і отримання результатів в зверненні

до підпрограм серверного об'єкту. Тут важливу роль відіграє вміння користуватися описами інтерфейсів віддаленого об'єкту в *WSDL*.

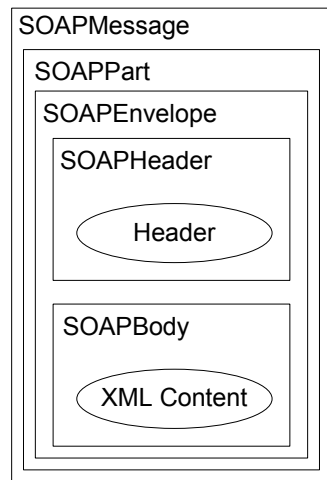


Рис. 3. Ієрархічна структура SOAP-повідомлення

Повідомлення можуть передаватися в синхронному або асинхронному режимі. В синхронному режимі (*request-response messaging*) процес передачі блокується на час очікування відповіді, яка також повинна відповідати *SOAP*-стандарту. В асинхронному режимі (*one-way messaging*) для передачі повідомлень використовуються посередники (провайдери). На час обміну процес передачі не блокується, але час повернення результату – непередбачуваний. Керування типами повідомлень виконується вибором з'єднання клієнта і провайдера (об'єктів *SOAPConnection* або *ProviderConnection*). Для передачі синхронних повідомлень використовується метод *call()* об'єкту *SOAPConnection*, параметром якого є структура, що представляє об'єктну модель *SOAP*-повідомлення, а результатом виконання запиту – також *SOAP*-повідомлення у форматі *DOM*. Для передачі асинхронних запитів використовується об'єкт *ProviderConnection* і його метод *send()*.

Режим *RPC* багато в чому нагадує технології обміну даними, що використовуються в об'єктних розподілених системах типу *RMI* і *OMG CORBA*. Основні моменти розгортання *RPC*:

розробка інтерфейсу віддаленого сервісного об'єкту (формальний опис методів об'єкту, типу параметрів і результату), що реалізовує обчислювальний сервіс (*Web Service*);

кодування методів сервісного об'єкту (*implementation*);

створення опису методів об'єкту в *WSDL*-документі;

декларація *WSDL*-документа;

обробка (пошук і аналіз) клієнтською програмою *WSDL*-документа;

з'єднання (*binding*) – підключення до серверу (провайдера сервісу);

формування об'єктного посилання для методів сервісного об'єкту;

звернення до методів сервісу і отримання результатів обчислень.

Треба відзначити два істотні моменти використання *RPC*:

оскільки єдиним форматом передачі даних в технології *Web*-служб є *SOAP*-повідомлення, тому серйозною проблемою є конвертація всіх викликів процедур і їх даних в стандартний формат, прийнятий в *SOAP*, тобто в *XML*-структуру;

взаємна конвертація *SOAP*-повідомлень і *RPC*-запитів виконують спеціальні програми – представники протилежної сторони. На стороні клієнта вони називаються стабами (*stubs*), а в серверній програмі для *Web*-служб вони називаються *tie*. Ці програми-посередники генеруються автоматично при підготовці і розгортанні клієнтських і серверних компонентів *Web*-служб.

Веб-служби забезпечують взаємодію програмних систем незалежно від платформи;

Веб-служби базуються на відкритих стандартах та протоколах. Завдяки використанню *XML* досягається простота розробки та відладки *Web*-сервісів;

Використання інтернет-протокола *HTTP* забезпечує взаємодію програмних систем через міжмережевий екран;

Більш низька продуктивність у порівнянні з технологіями *CORBA*

(CORBA (англ. *Common Object Request Broker Architecture* – загальна архітектура брокера об'єктних запитів) – це запропонований консорціумом OMG технологічний стандарт розробки розподілених застосувань), DCOM (Об'єктна модель розподілених компонент (англ. *Distributed Component Object Model*) – власна технологія Microsoft для організації взаємодії між компонентами програмного забезпечення, розподіленого між комп'ютерами в мережі) за рахунок використання текстових XML повідомлень.

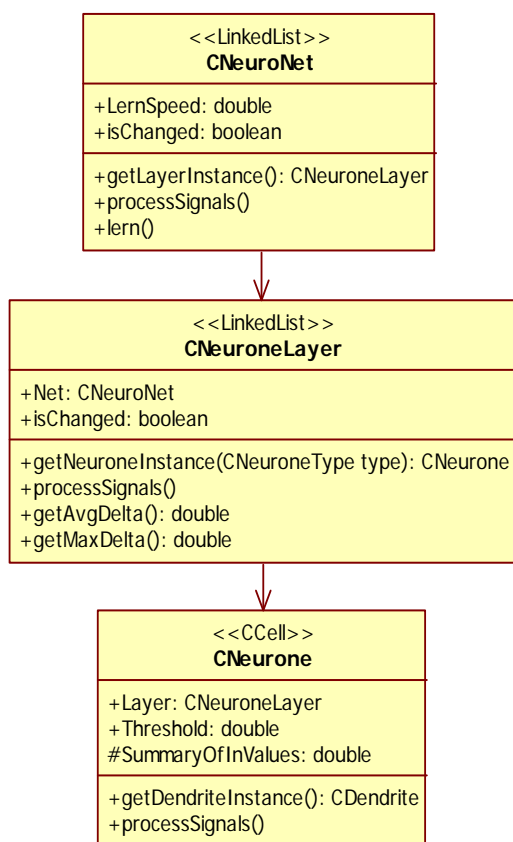


Рис. 4. UML діаграма структури базових класів

Взаємодія між кластерами нейронної мережі організовано наступним чином. Нейронна мережа будується з застосування наступних базових класів рис. 4. *CNeuroNet* – цей клас зберігає структуру всієї мережі. До нього входять класи шарів мережі *CNeuroneLayer*. Клас *CNeuroneLayer* –

включає до себе всі нейрони визначеного шару нейронної мережі *CNeurone*. Клас *CNeurone* – це базови клас для усіх типів нейронів у мережі.

До класу *CNeuroNet* можуть входити як об'єкти *CNeuroneLayer* поточного обчислювального так і віртуальні об'єкти. Взаємодія з віртуальними об'єктами *CNeuroneLayer* організована з використанням технології *Web*-сервісів.

Кожен слой нейронної мережі може бути реалізован як нейронна мережа на іншому обчислювальному вузлі. Але для поточної нейронної мережі вона буде виглядати як слой нейронів зі своїми входами та виходами.

Висновки

Використання *Web*-сервісів для побудови динамічної кластерної нейронної мережі дозволить легко та не відходячи від об'єктної моделі організувати різноманітні нейронні мережі. Також *Web*-сервіси дозволять зберігати створений код при зміні структури мережі. Потрібно буде лише змінити відповідні файли конфігурації.

Сама ідея *Web*-сервісів дозволить динамічно змінювати структуру нейронної мережі з використанням розроблених методів та алгоритмів.

Список літератури

1. Саймон Хайкин Нейронные сети: полный курс. – М.: Издательский дом «Вильямс», 2006. – 1105 с.
2. Генри Бекет, Мераж Куннумпурат, Шон Роді, Андре Тост Java. Основы Web-служб. – М.: Издательство «КУДИЦ-Образ», 2004. – 464 с.
3. Генри Бекет Java SOAP для профессионалов – М.: Издательство «Лори», 2004. – 458 с.