

інформацію між двома будь-якими сенсорами мережі, причому з необхідною швидкістю передачі (смугою пропускання);

- сенсори повинні споживати малу кількість енергії, щоб працювати впродовж декількох років без заміни батарей;
- сенсори повинні дуже швидко реагувати;
- сенсори повинні бути непомітними і високонадійними в експлуатації;
- сенсори повинні мати низьку вартість.

ВИСНОВКИ:

- 1) Досліджено особливості побудови сенсорних мереж та визначено її основні переваги: повна відсутність кабелів – електричних, комунікаційних і т.д.
- 2) Шляхом аналізу особливостей формування топології сенсорних мереж запропоновано можливість компактного розміщення або навіть інтеграції сенсорів в об'єкти навколишнього середовища;
- 3) Досліджені питання надійності як окремих елементів так і всієї системи сенсорної мережі в цілому: у ряді випадків мережа може функціонувати при справності тільки 10-20 % сенсорів;

Література

1. Борщ В.І., Коршун Є. І., Туманов Ю. Г., Чумак М. О. Сигналізація і синхронізація в телекомунікаційних системах. – К.: Наукова думка, 2004.
2. Вишневский В.М., Ляхов А.И., Портной С.Л., Шахнович И.В. Широкополосные беспроводные сети передачи информации. – М.: Техносфера, 2005. – 591 с.
3. Пантелейчук А. Построение беспроводной сети датчиков // Электронные компоненты. – 2008. – № 4.
4. Мішан В.В. Синхронизовані автогенераторні масиви з низькодобротними коливальними системами / В. В. Мішан // Радиоэлектроника и молодежь в XXI веке: материалы 11 юбилейного международного молодежного форума 10 – 12 апреля 2008 г. – Харьков: ХНУР, 2008. – С. 185.

Надійшла 16.2.2010 р.

УДК 004.891.3

Д.М. ДРАГУН, Д.Ю. ЧАЙКОВСЬКИЙ
Хмельницький національний університет

ДОСЛІДЖЕННЯ МОЖЛИВОСТІ ВИКОРИСТАННЯ АГЕНТНОГО ПІДХОДУ ДЛЯ ДІАГНОСТУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ

Проаналізовано агентні технології як засіб діагностування комп'ютерних систем. Досліджено ефективність формальних моделей агентів та діагностичних агентних систем для діагностування спеціалізованих комп'ютерних систем, проаналізовано їх недоліки. Розглянуто можливість розподілення етапів і задач діагностування між агентами у кластерних системах.

Analyzed agent technology as a means of diagnosing computer systems. Investigated the effectiveness of formal models of agents and agent diagnostic system for diagnosing cpetsializovanyh computer systems, analyze their shortcomings. The opportunity of distribution of phases and tasks between agents in the diagnosis of cluster systems.

Ключові слова: діагностування, комп'ютерні системи, агентні технології.

Вступ

Зростання вимог до комп'ютерних систем (КС) призводить до звуження їх спеціалізації та збільшення вимог до рівня їх надійності. Для забезпечення необхідного рівня надійності, КС потребують ряду комплексних засобів, серед яких програмно-апаратні засоби діагностування є важливим компонентом. Ускладнення сучасних КС потребує розвитку та вдосконалення методів і засобів їх діагностування. Однією із актуальних задач є діагностування спеціалізованих КС на етапі експлуатації.

Сучасні КС як об'єкти діагностування мають ряд особливостей [1-4], котрі впливають на динаміку розвитку засобів діагностування і ефективність процесу діагностування, а саме:

- 1) низький рівень контролепридатності сучасних КС та їх складових;
- 2) відсутність або надзвичайно висока вартість спеціалізованих ефективних діагностичних апаратних засобів та програм діагностування КС;
- 3) динаміка нарощення можливостей КС призводить до зростання спектру апаратних складових та їх постійної модернізації, що робить нерентабельною закупівлю вузькоспеціалізованого діагностичного обладнання;
- 4) етап експлуатації КС недостатньо забезпечений документацією. У той же час, на цьому етапі необхідно забезпечити діагностування надзвичайно широкого спектру КС, тому успішність етапу

експлуатації фактично повністю залежить від рівня кваліфікації спеціалістів, що використовують та обслуговують комп'ютерну техніку.

Отже, розвиток методів і засобів діагностування КС на етапі експлуатації є актуальною задачею. Зокрема, на сьогодні не приділяється достатньо уваги методам інтелектуального діагностування КС з використанням агентних систем.

Постановка задачі

Отже, на етапі експлуатації КС актуальною є задача підвищення ефективності діагностування комп'ютерних систем шляхом використання інтелектуальних методів та засобів діагностування. Для вирішення цієї задачі необхідно дослідити агентні технології, як засоби діагностування, формальні моделі агентів та агентних систем діагностування.

Метою статті є дослідження можливості використання агентного підходу для діагностування комп'ютерних систем.

Аналіз агентних технологій як засобу діагностування КС

Перевагами використання агентного підходу в технічній діагностиці є: автономність функціонування системи діагностування, розподіл та децентралізація задач діагностування, можливість паралельного опрацювання та аналізу діагностичної інформації.

Поняття “агент”, “мультиагентна система”, “агентно-орієнтоване програмування” не є чітко визначеними поняттями не тільки у контексті діагностування, а й у методології штучного інтелекту взагалі, оскільки на даний момент цей напрямок лише розвивається. Існує багато дослідницьких груп, що використовують ці поняття і інтерпретують їх визначення у своїх цілях [5-7].

До простих агентів відносять агентів, які є подальшим розвитком об'єкта, проте простий агент є автономним; приймає інформацію з оточуючого світу та взаємодіє з іншими агентами. Простий діагностичний агент є „мікрофоном”, що спостерігає за станом об'єкта діагностування і надсилає дані іншим агентам або користувачу.

Розумні агенти мають можливість використовувати предметні знання. Тобто, вони здатні не тільки передавати дані іншим агентам, а й формувати свої висновки чи рішення та передавати знання іншим. Розумний агент-діагност має змогу пересилати дані про стан пристрою, робити свої висновки щодо якості та коректності його функціонування.

Інтелектуальний агент повинен вміти не просто робити висновки на основі вхідних даних, слідкувати за оточенням та взаємодіяти з іншими агентами, а й навчатись з оточуючого середовища. Навчання з середовища передбачає не тільки поповнення бази знань, а й адаптації до оточення, а також переформування як своєї бази знань, так і метазнань. Бажано, щоб при створенні інтелектуального агента без наявності чіткої цілі, агент самостійно міг її сформувати зі списку найбільш підходящих чи доступних цілей. Діагностичний інтелектуальний агент, основна мета якого – визначення стану об'єктів діагностування, при відсутності мети повинен сформувати список пристроїв, які необхідно контролювати та, у разі необхідності, розпочати діагностування залежно від стану пристроїв.

Отже, інтелектуальним агентом є агент, що містить знання про навколишній світ і здатен приймати певні рішення на основі цих знань. Інтелектуальний агент-діагност – це агент, який містить знання про об'єкт діагностування (ОД) і середовище діагностування, здатен самостійно визначати стан ОД на основі наявних в нього даних і знаннях, а у разі, коли знань недостатньо, взаємодіяти з іншими агентами для уточнення стану ОД.

Системи, у яких передбачена взаємодія агентів, називаються мультиагентними. Відповідно, системи, у яких взаємодіють агенти-діагности, є мультиагентними системами діагностування. Мультиагентні системи, що складаються з взаємодіючих автономних агентів надають можливість для побудови комплексних розподілених додатків. Парадигма “мультиагентна система” представила нові проблеми у проектуванні і побудові систем діагностування, які мають бути сумісними з традиційними підходами до проектування. Відповідно, виникла необхідність у розробленні як нових принципів і методологій побудови, так і нових засобів для ефективного вирішення поставлених задач, враховуючи особливості не тільки ОД, а й особливості і можливості середовища, у якому здійснюється діагностування.

Існує ряд агентних методологій, серед яких найвідомішими є [8-15]:

- BDI, “намір-бажання-віра” людського практичного міркування. Концепція BDI була реалізована рядом проектів, а саме: PRS, IRMA, UM-PRS, dMARS, AgentSpeak, JAM, JACK, JADEX, Jason, SPARK, ZAPL. Найвідомішими є JACK та JADEX. Недоліком концепції BDI є нагромадження філософських і психологічних понять у агентній термінології, недостатній зв'язок між абстрактними поняттями і формальним описом, що ускладнює її застосування для діагностування комп'ютерних систем;

- Tropos. Мова Tropos призначена для концептуального моделювання і формалізована у метамодель, що описана множиною діаграм-класів UML. Використання UML для формалізації моделей агентів є поширеним підходом у агентних методологіях. Недоліком методології Tropos є те, що вона охоплює лише початкові стадії проектування, тобто вона придатна для опису предметної галузі, розроблення моделей комп'ютерних систем, планування процесу діагностування, але вона не призначена для створення систем технічного діагностування;

- MaSE є методологією загального призначення для побудови гетерогенних мультиагентних систем. MaSE використовує графічно побудовані моделі для опису системних цілей, поведінки, агентних типів

і інтерфейс комунікації агентів. MaSE також забезпечує шлях для визначення архітектурно-незалежного детального визначення внутрішнього проекту агента, що надає можливість створення узагальненого визначення поняття агента-діагноста. Недолком методології MaSE є узагальнене представлення моделей агентних систем, що не враховує специфіки прикладної області, в зв'язку з чим знижується ефективність використання даної методології при розробці мультиагентних систем діагностування;

- Foundation for Intelligent Physical Agents (FIPA). Специфікації FIPA представляють колекцію стандартів, які забезпечують просування взаємодії гетерогенних агентів і сервісів, які вони можуть надавати. Такий підхід надає можливість взаємодії розподілених агентних систем діагностування для обміну досвідом і підвищення достовірності діагностування. Основою мультиагентної моделі специфікацій FIPA є агентна взаємодія, яка відбувається на мові ACL (Agent Communication Language). Недоліком FIPA є те, що на детальному рівні стандартизовано лише передачу повідомлень, моделі агентів і агентних систем представлені на абстрактному рівні. Тобто, при розробці агентних систем діагностування існує необхідність розроблення на деталізованому рівні як моделі передчі повідомлень, моделі агентів, так і моделі самої агентної системи;

- Agentcities визначає множину додаткових протоколів/механізмів для забезпечення мережних вузлів можливістю відкриття інших вузлів і для наявності інформаційних мережних сервісів для забезпечення детальнішої картинки стану мережі. Цілями створення Agentcities є: простий механізм обміну даними між платформами Agentcities, а також простий механізм для створення, збирання і візуалізації мереж, що працюють на платформах Agentcities. Agentcities побудовано згідно стандартів FIPA щодо обміну інформацією між агентами. Це дозволяє використовувати розробки даного проекту для обміну діагностичною інформацією між агентами-діагностами, зменшуючи час діагностування шляхом використання досвіду інших агентів-діагностів;

- Gaia призначена для розробки мультиагентних систем. Мультиагентна система може бути представлена і спроектована, як обчислювальна система і визначає відповідні абстракції, які є основними у аналізі та проектуванні цих систем. Недоліком Gaia є недостатнє врахування якісних параметрів проєктованих систем, які відіграють важливу роль у системах технічного діагностування.

На даний момент існує ряд середовищ для побудови мультиагентних систем, розроблених згідно методологій, описаних вище:

- JASK є середовищем для створення, виконання й інтегрування комерційно-орієнтованих мультиагентних систем, що використовують компонентний підхід. Агентна мова JASK – мова програмування, що розширює мову Java агентно-орієнтованими концепціями, такими як: агенти; здатності; події; плани; бази знань агента; управління паралелізмом;

- AgentBuilder є інтегрованим програмним засобом для побудови інтелектуальних програмних агентів. AgentBuilder складається із двох головних компонентів: інструментарію розробки і системи виконання (Run-Time System). Інструментарій включає в себе засоби для керування процесом розробки агентно-орієнтованої системи, аналіз можливих операцій агента, проектування й розробки мереж сполучених за допомогою агентів та ін. У середовищі AgentBuilder агенти взаємодіють на мові KQML (Knowledge Query and Manipulation Language). Для специфікації поведінки агента в ній використовується спеціальна мова, що базується на конструкціях виду WHEN-IF-THEN. Усі компоненти інструментарію AgentBuilder і системи виконання розроблені на мові Java. Це означає, що створювані додатки можуть виконуватись на будь якій платформі, що має віртуальну машину Java;

- AgentTool. Це середовище побудоване згідно принципів методології MaSE. Проектувальник графічно створює високорівневу агентну модель, створює структуру агентів, а потім забезпечує детальну поведінкову специфікацію для кожного компонента в архітектурі агента. Перевагою AgentTool у порівнянні з іншими мультиагентними середовищами є вбудована бібліотека компонентів штучного інтелекту;

- середовище Bee-agent повністю “агентизоване”, тобто всі дії, що виконуються у системі, виконуються агентами, а всі взаємодії виконуються між агентами. Дане середовище дозволяє створювати гнучкі розподілені агентні системи діагностування, що будуть взаємодіяти з існуючими програмними засобами. Специфікація поведінки здійснюється на основі протоколів взаємодії агентів. Такі протоколи представляються спеціальними графами, основними поняттями яких є стани й переходи. Для опису протоколу використовується XML.

Найпоширенішим середовищем є JASK. Наявність великої кількості програмного забезпечення, розробленого сторонніми виробниками, робить його придатним для створення інтелектуальних систем діагностування. Проте, недоліком JASK та відомих агентних середовищ, більшість з яких розроблені на мові JAVA, є їх ускладнена взаємодія із засобами діагностування (діагностичними програмами; давачами, які надають діагностичну інформацію), що вимагає розроблення додаткових засобів взаємодії.

Використання мультиагентних систем у діагностуванні є перспективним напрямком досліджень, що надає наступні переваги у порівнянні з класичними підходами діагностування: децентралізований збір інформації; мінімальні витрати ресурсів на діагностування; існування у системі, що досліджується декількох агентів-контролерів; формування результату багатьма суб'єктами, кожен з яких має свою думку; розподілені обчислення. Однак, різна інтерпретація агентних технологій ускладнює їх застосування у діагностуванні.

Дослідження відомих моделей агентів та агентних систем діагностування

Розглянемо множину формальних моделей агентів з метою визначення їх придатності для

вирішення задачі підвищення ефективності діагностування компонентів КС [16 – 18].

У роботі [16] агент представляється сутністю, що має пам'ять і містить параметри як самого агента, так і параметри, за допомогою яких агент взаємодіє з середовищем [16]:

$A_g = \{a_1, \dots, a_n\}$: множина ситуаційних агентів;

$y_i \in Y$: індивідуальність a_i з $Y = \{y_1 \dots y_n\}$ множиною унікальних індивідуальностей, по одній на кожного агента системи;

$Id : A_g \rightarrow Y$ є функцією, що повертає індивідуальність агента, тобто $Id(a_i) = y_i$;

$\Psi = 2^Y$ множина всіх можливих підмножин індивідуальностей агентів;

$p_i \in P_i$: сприйняття агентом a_i навколишнього середовища, P_i – множина всіх можливих сприйнять агента a_i ;

C : множина можливих витрат (ресурсних, часових та ін.);

$c_i \in C_i$: витрати агента a_i , $C_i \subseteq C$ - множина всіх можливих витрат, які можуть бути витрачені агентом a_i ;

$s_i \in S_i$: внутрішній стан a_i , S_i – всі можливі стани a_i ;

$\psi_i \in \Psi$: множина індивідуальностей агентів.

У роботі [17] представлена формальна модель “подія-умова-дія” колективного і реактивного діагностичного агента, що базується на логічному програмуванні і описана наступними предикатами:

- send (From, To, Msg) призначений для взаємодії між агентами;
- do (Action) призначений для виконання дій;
- remember (Notice), forget (Notice) для маніпуляції станом агента. Послідовність дій поєднуються оператором &.
- recv (From, To, Msg) є реакцією на подію.

Мультиагентна система діагностування з семантично розподіленими знаннями, представлена у роботі [18] задається наступними параметрами:

$$S = \{C, M, Id, Sd, Ctx, Obs\},$$

де C – множина компонентів;

$M = \{M_c \mid c \in C\}$ – можливі несправності кожного компонента;

Id – множина ідентифікаторів р точок з'єднання між компонентами;

Sd – опис системи;

Ctx – множина вхідних значень системи, яка визначена ззовні системи середовищем;

Obs – множина визначених значень системи.

Недоліками відомих моделей агентів та агентних діагностичних систем є:

- 1) відсутня класифікація агентів за виконуваними задачами, що ускладнює розподілення задач діагностування;
- 2) задачі діагностування не є розподіленими між агентами, що не дозволяє проводити їх розпаралелювання;
- 3) недостатньо описана взаємодія користувача з системою діагностування і агентів з об'єктом діагностування;
- 4) недостатня формалізованість моделей, що ускладнює розроблення на їх основі методів діагностування.

Кластерні системи та засоби їх діагностування

Одним із видів спеціалізованих КС є кластерні системи. Впровадження таких кластерних рішень, як The High Availability Linux Project, Linux Virtual Server, Beowulf, WCCS та ін. для вирішення задач, що потребують високопродуктивних обчислень, на сьогодні визначається їх дешевизною у порівнянні з аналогічними рішенням, побудованими на базі суперкомп'ютерів [19 – 23].

Високопродуктивні кластери, що працюють під керуванням операційних систем Linux і Windows, забезпечують можливість вирішення широкого спектру нагальних обчислювальних задач, серед яких: інформаційні сховища, системи прийняття рішень, додатки з онлайнною обробкою транзакцій (OLTP) та ін.

Складність технічного обслуговування кластерних систем набагато вища, ніж складність обслуговування комп'ютерних систем. Це сприяло появі широкого спектру апаратного і програмного забезпечення адміністрування, контролю і діагностування кластерів під Linux, таких як OpenPBS, Ganglia, MRTG, C3, Torque та ін. Існує ряд діагностичного програмного забезпечення для діагностування хостів кластерних систем.

Для оцінювання ефективності відомих засобів діагностування було проведено ряд досліджень. З метою виявлення залежностей значень параметрів робочих станцій кластерів від умов їх функціонування і конфігурацій було проведено дослідження температурних режимів жорстких дисків, компонентів

комп'ютерних і кластерних систем (процесора, відеокарти, елементів системної плати, кулерів), класифіковано програмні додатки за рівнем їх навантаження на компоненти (рис. 1).

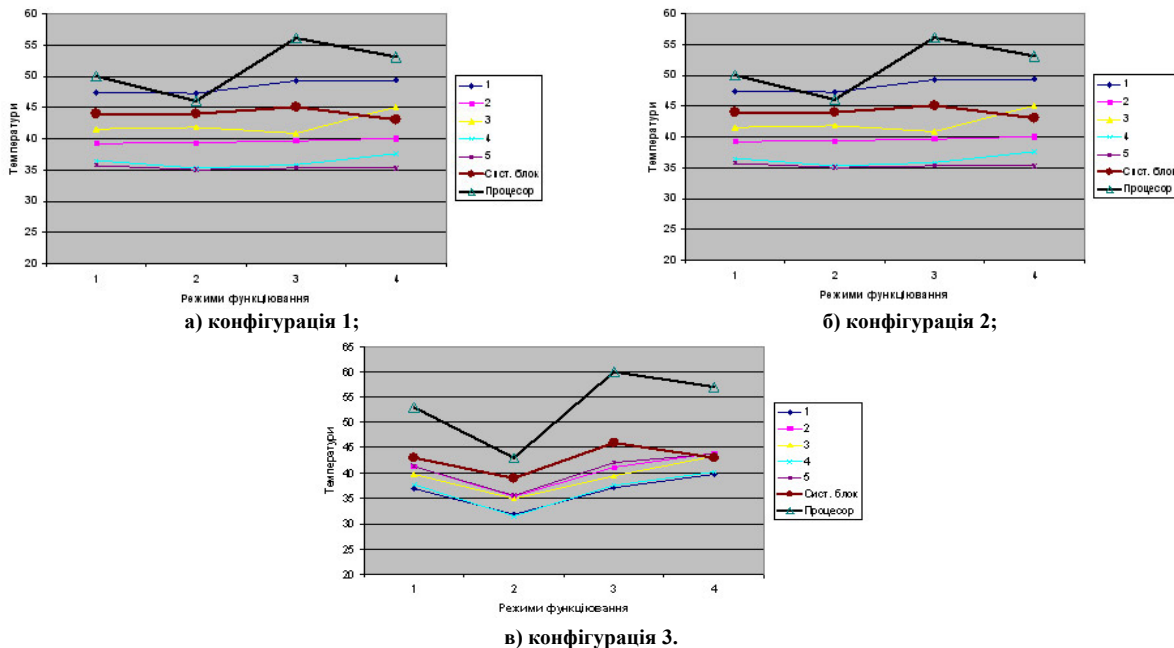


Рис. 1. Результати досліджень залежностей температурних режимів компонентів КС від умов функціонування

Місця встановлень температурних датчиків:

1. на мікросхему North Bridge;
2. на мікросхему South Bridge;
3. на тильну сторону відеокарти у місці встановлення процесора відеокарти;
4. на USB 2.0 controller у конфігурації 1 та на транзистор Mosfet біля центрального процесора у конфігураціях 2, 3;
5. на Super I/O Controller у конфігурації 1 та на котушку індуктивності біля центрального процесора у конфігураціях 2, 3.

Конфігурації, розглянуті на рис. 1, що були досліджені на предмет зміни температур і напруг:

- конфігурація 1: процесор – Celeron-2000, системна плата – ASUS P4XP-X, оперативна пам'ять – 256 Мб;
- конфігурація 2: процесор – Athlon64 XP 3000+, системна плата – ASUS M2NPV-MX, оперативна пам'ять – 512 Мб;
- конфігурація 3: процесор – Sempron 2500+, системна плата – Abit KV8 Pro, оперативна пам'ять – 512 Мб.

Експериментальні дослідження при різних умовах функціонування КС:

1) нормальні умови. Системний блок закритий. Центральний процесор функціонує на робочій частоті і коефіцієнтах множення, які визначені відповідною специфікацією і вважаються нормальними. Рух повітряних потоків є вільним. Мета – визначення температур компонентів КС за нормальних умов, які в подальшому будуть використовуватись для порівняння;

2) відкритий системний блок. З системного блоку зняті металеві кришки. Рух повітряних потоків є вільним, частоти – нормальними. Мета – дослідження охолодження компонентів КС при зміні руху повітряних потоків;

3) розгін або оверклокінг, що передбачає штучне форсування режимів функціонування деяких компонентів КС з метою збільшення продуктивності комп'ютера в цілому. Найпоширенішими видами розгону є розгін процесора і відеокарти. Розгін виконується шляхом підвищення частот, на яких працює компонент і/або збільшення коефіцієнту множення. Функціонування компонентів КС у режимі розгону призводить до підвищення рівня споживаної ними потужності, що веде до збільшення температури компонентів, які підлягають розгону. При даному режимі центральний процесор функціонує на підвищеній тактовій частоті. Інші умови функціонування є аналогічними до нормальних умов. Мета – визначення впливу зміни частот на температуру компонентів;

4) ускладнений повітряний хід. У системному блоці встановлено додаткові кабелі, шлейфи та елементи, що погіршують повітряний хід. Системний блок закритий, частоти нормальні, повітряний хід ускладнений. Мета – визначення температури компонентів всередині системному блоці при погіршенні охолодження повітряними потоками.

Дослідження динаміки зміни температурних режимів, навантажень, робочих частот компонентів комп'ютерних систем виявило неоптимальні режими функціонування таких компонентів як North Bridge, South Bridge, деяких моделей відеокарт у зв'язку з відсутністю засобів контролю режимів функціонування і

діагностичних засобів.

У процесі експериментальних досліджень температурних режимів жорстких дисків різних виробників виявлено причинно-наслідкові зв'язки умов функціонування та виконуваних програмних додатків. Дослідження показало, що вбудовані температурні датчики жорстких дисків не завжди відображають реальну температуру елементів, на яких вони встановлені, створюючи ситуації, при якій температура фіксується датчиком, як та, що знаходиться у межах допусків, а в той час реальна температура деяких елементів є неоптимальною [24-26].

Одним із шляхів вирішення вищевказаних проблем є розроблення методів та засобів інтелектуального діагностування кластерів, котрі задовольнятимуть ряд вимог, а саме: забезпечать автономність функціонування діагностичних модулів, розподіленість збирання та опрацювання діагностичної інформації, можливість своєчасної постановки діагнозу в умовах неповноти діагностичної інформації, підвищуватимуть живучість кластерів.

Отже, на етапі експлуатації кластерних систем (КлС) актуальною є задача забезпечення ефективності діагностування робочих станцій та серверів кластера шляхом використання різнотипних настроєваних діагностичних засобів.

Етапи діагностування компонентів КлС умовно можна розділити на:

- 1) основні (контроль, локалізація причин зміни, прогнозування стану компонентів КлС);
- 2) допоміжні (збір діагностичної інформації, аналіз діагностичної інформації, обмін діагностичною інформацією).

Для комп'ютерних систем послідовне виконання етапів діагностування є прийнятним, а для кластерних систем це є суттєвим недоліком. При діагностуванні множини компонентів КлС ефективність діагностування знижується у зв'язку з послідовним діагностуванням кожного компонента КлС.

Складність розпаралелювання процесу діагностування компонентів КлС пов'язана зі складністю організації та динамічної реорганізації агентного середовища системи діагностування.

Таким чином, для підвищення ефективності діагностування компонентів КлС слід організувати їх паралельне діагностування без нарощення ресурсоемності системи діагностування, дослідити можливості розподілення етапів і задач діагностування між агентами у кластерних системах.

5. Розподілення етапів і задач діагностування між агентами у кластерних системах

Розподілення процесу діагностування полягає у розподіленні його етапів (збір, аналіз, діагностування, контроль, локалізація, прогнозування) між агентами, що виконують покладені на них задачі і обмінюються результатами своєї роботи одночасно [24-26]. Розподілимо функції і етапи діагностування між агентами:

1) агент збору даних. Реалізується за допомогою програми-агента, який зчитує навантаження, температуру компонентів КС; частоти обертання кулерів та інші параметри кожні T сек.;

2) агент збору локальних даних. Накопичує дані при різних умовах функціонування компонентів КС. Зчитує дані про конфігурацію;

3) агент-генератор знань. Формує знання з отриманих даних, які надходять від агенту збору тимчасових знань і агенту збору локальних знань;

4) інтерфейсний агент. Забезпечує зв'язок між системою діагностування і користувачем. Отримані від користувача запити (можливість зниження температури, можливість запуску додатків на вказаній конфігурації та ін.) передає до агента-діагноста. Після аналізу запиту агентом-діагностом інтерфейсний агент повертає результат користувачу. У разі можливості задоволення запиту користувачу або можливості оптимізації температурних режимів формує послідовності дій по усуненню причин зростання температурних режимів компонентів КС за допомогою автоматичного усунення (зниження робочих частот, зниження навантаження, підвищення обертів кулерів) або рекомендацій користувачу по реконфігурації системи;

5) агент-діагност. Проводить аналіз потреби та можливості зміни конфігурації і умов функціонування для забезпечення оптимальних температурних режимів при мінімальних витратах (фінансових, продуктивності) шляхом аналізу тимчасових та локальних знань і, при необхідності, знань розподілених агентних систем діагностування;

6) агент-комунікатор. Забезпечує обмін діагностичними знаннями між агентними системами діагностування.

Особливістю розподілення процесу діагностування у кластерній системі є розподілення агентів між робочими станціями (хостами), що дозволяє розподілити навантаження між ними. На рис. 2 представлено розподіл агентів між робочими станціями кластерної системи, де:

- 1 – агент-збирач діагностичної інформації;
- 2 – агент-генератор знань;
- 3 – інтерфейсний агент;
- 4 – агент-комунікатор;
- 5 – агент-діагност.

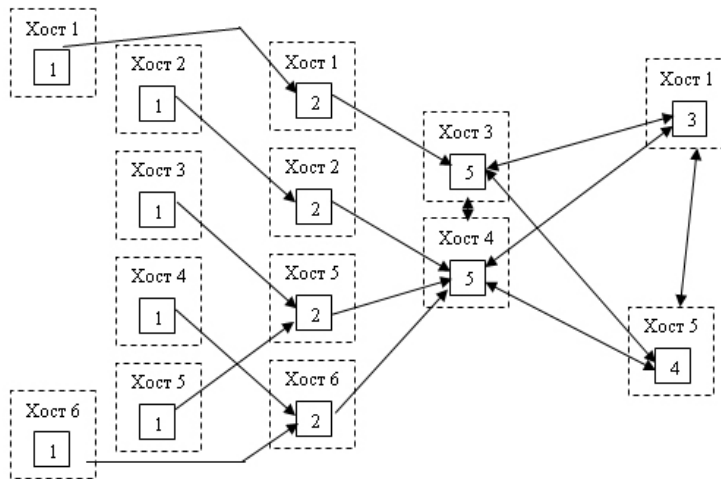


Рис. 2. Приклад обміну діагностичною інформацією між агентами у кластерній системі

Агенти-збирачі існують на всіх хостах, агенти-генератори знань функціують на хостах 1, 2, 5, 6. Хост 5 опрацьовує дані від хостів 3 і 5, хост 6 – від хостів 4 і 6. Хости 3 і 4 на основі аналізу діагностичної інформації, знань про конфігурацію, що діагностується проводять діагностування компонентів. Хост 1 і хост 5 виконують інтерфейсну і комунікаційну функції.

Розподілений процес діагностування у порівнянні з послідовним при діагностуванні комп'ютерних і кластерних систем надає такі переваги: можливість отримання попередніх результатів діагностування вже на ранніх етапах; отримання результатів діагностування за рахунок відсутності надлишкового використання ресурсів, як у випадку послідовного процесу діагностування; можливість підвищення ефективності діагностування спеціалізованих комп'ютерних систем шляхом перерозподілу агентів між робочими станціями (хостами).

Висновки

Використання агентного підходу надає можливість підвищити ефективність діагностування компонентів КС шляхом розподілу та децентралізації задач діагностування між агентами; паралельного виконання збору, опрацьовання та аналізу діагностичної інформації; збільшення автономності функціонування системи діагностування.

Проте, сучасний рівень розвитку агентних технологій ускладнює їх застосування у діагностуванні, що призводить до необхідності розробки таких методів і засобів для агентних систем діагностування, які б ефективно вирішували проблеми процесу діагностування, а саме: розподілення даних, контроль, експертиза та ресурси, що важко реалізуються (або взагалі не реалізуються) традиційними методами діагностування.

Відомі системи діагностування кластерів призводять до нерегульованого дисбалансу завантаження вузлів та комутаційних каналів, що є наслідком жорсткості архітектур систем діагностування. Розподілення етапів процесу діагностування та діагностування компонентів спеціалізованих КС надає можливість опрацьовання більших об'ємів діагностичної інформації у реальному часі у порівнянні з процесом послідовного діагностування за рахунок децентралізованого перерозподілу функцій, розподілу агентів між робочими станціями, автономності функціонування агентів.

Література

1. Локазюк В.М. Проблеми та методологія контролю і діагностування сучасних мікропроцесорних пристроїв та систем. / Локазюк В.М // Вимірювальна та обчислювальна техніка в технологічних процесах. – № 2. 2000. – С. 10 – 17.
2. Локазюк В. М. Інтелектуальне діагностування мікропроцесорних пристроїв та систем: [навч. посібник для вузів] / В. М. Локазюк, О. В. Поморова, А. О. Домінов. – К.: "Такі справи", 2001. – 286 с.
3. Поморова О. В. Теоретичні основи, методи та засоби інтелектуального діагностування комп'ютерних систем: Монографія. – Хмельницький: ТОВ „Триада-М”, 2007. – 253 с.
4. Надійність комп'ютерних систем / [Тарасенко В. П, Маламан А. Ю., Черніченко Ю. П., Корнійчук В. І.]. – К.: «Корнійчук», 2007. – 256 с.
5. Рассел С. Искусственный интеллект: современный подход. / С. Рассел, П. Норвиг; пер. с англ. – 2-е изд. – М.: Издательский дом "Вильямс", 2006. – 1408 с.
6. Искусственный интеллект: стратегии и методы решения сложных проблем. – Люгер Д [4-е изд]. – М.: Издательский дом «Вильямс», 2003. – 864 с.
7. DeLoach S. A. Developing Multiagent Systems with agentTool / S. A. DeLoach, M. F. Wood // In Intelligent Agents VII. Agent Theories Architectures and Languages, 7th International Workshop. – ATAL 2000, Boston, MA, USA, July 7-9, 2000. – P. 121 – 125.
8. Weiss G. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. – New-York:

The MIT Press. – 2001. – 648 p.

9. Wooldridge, Michael and Nicholas R. Jennings. Agent Theories, Architectures, and Languages: a Survey, in Wooldridge and Jennings Eds // Intelligent Agents. – Berlin: Springer-Verlag. – 1995. – P. 1 – 22.

10. Smith D. C. KidSim: Programming Agents Without a Programming Language / D. C. Smith, A. Cypher, J. Spohrer // Communications of the ACM. – 37, 7. – 1994. – P. 55 – 67

11. Wooldridge M. Theory and Practice / M. Wooldridge, N. Jennings // Intelligent Agents. – Knowledge Engineering Review. – № 10. – 1995.

12. Brustoloni, Jose C. Autonomous Agents: Characterization and Requirements // Carnegie Mellon Technical Report CMU-CS-91-204, Pittsburgh: Carnegie Mellon University. – 1999. – P. 122 – 221.

13. Guinchiglia F. The Tropos Software Development Methodology: Processes, Models and Diagrams / F. Guinchiglia, J. Mylopoulos, A. Perini // The Third International Workshop on Agent-Oriented Software Engineering (AOSE-2002). – Bologna, Italy. – July 2002.

14. Bauer B. Agent UML: A formalism for specifying multiagent software systems / B. Bauer, J. P. Muller, J. Odell // Int. J. Softw. Eng. Knowl. – Eng. 11, 3 (Apr.). – 2001. – P. 207 – 230.

15. Milojicic D. Mobil Agent Applications [Електронний ресурс]. – Режим доступу: <http://www.computer.org/concurrency/pd1999/pdf/p3080.pdf>.

16. Maes, Pattie. Artificial Life Meets Entertainment: Life like Autonomous Agents // Communications of the ACM. – 38. – № 11. – 1995. – P. 108 – 114.

17. Nwana H. A Perspective on Software Agents Research: An Overview / H. Nwana, D. T. Ndumu [Електронний ресурс]. – Режим доступу: <http://www.cs.umbc.edu/awclassic/introduction/hndn-ker99.html>

18. Weyns D. A Formal Model for Situated Multi-Agent Systems / D. Weyns, T. Holvoet // Fundamenta Informaticae IOS Press. – № 63. – 2004. – P.1– 34 [Електронний ресурс]. – Режим доступу: <https://lirias.kuleuven.be/bitstream/123456789/124991/1/2004FI.pdf>

19. Schroeder M. A Deliberative and Reactive Diagnosis Agent based on Logic Programming Rechnergestützte Wissensverarbeitung Universität Hannover Lange Laube 3, 30159 Hannover, Germany [Електронний ресурс]. – Режим доступу: <http://www.di.uevora.pt/~ica/research/fullpaper/mora-ictai96.pdf>.

20. Roos N. Multi-Agent Diagnosis with semantically distributed knowledge Universiteit Maastricht, IKAT, P.O.Box 616, 6200 MD Maastricht / N. Roos, A. Teije, C. Witteveen [Електронний ресурс]. – Режим доступу: http://www.st.ewi.tudelft.nl/~renze/pubs/ RoosTeijeWitteveen_BNAIC2003.pdf.

21. Portable Batch System [Електронний ресурс]. – Режим доступу: <http://www.pbspro.com/>.

22. Ganglia Monitoring System [Електронний ресурс]. – Режим доступу: <http://www.ganglia.info/>.

23. Multi Router Traffic Grapher [Електронний ресурс]. – Режим доступу: www.usenix.org/event/lisa98/full_papers/oetiker/oetiker.pdf.

24. Cluster Command Control [Електронний ресурс]. – Режим доступу: <http://www.csm.ornl.gov/torc/C3/>.

25. Terascale Open-source Resource and Queue Manager [Електронний ресурс]. – Режим доступу: <http://supercluster.org/torque/>.

26. Поморова О. В. Розподілена мультиагентна система діагностування комп'ютерних пристроїв / О. В. Поморова, Д. Ю. Чайковський // Оптиелектроніка. – 2005. – № 2 (10). – С. 114 – 118.

27. Поморова О. В. Агентний метод розпаралелювання процесу діагностування кластерних систем / О. В. Поморова, Д. Ю. Чайковський // Радіоелектронні і комп'ютерні системи. – Харків: Національний аерокосмічний університет ім. М. Є. Жуковського „Харківський авіаційний інститут”. – 2009. – № . 7 (41). – С. 164 – 170.

28. Поморова О. В. Агентне діагностування комп'ютерних систем / О. В. Поморова, Д. Ю. Чайковський // Науковий вісник Чернівецького університету: збірник наукових праць. Вип. 423: Фізика. Електроніка: Тематичний випуск «Комп'ютерні системи та компоненти»: Частина I. – Чернівці: ЧНУ, 2008. – С. 102 – 109.

Надійшла 15.2.2010 р.