**9. Среди за разпределена oбработка**

**Обзор на РСА –** тема 8;

**Клъстерна обработка**

Компютърните клъстери предоставят разпределен ресурс като виртуален уникомпютър; типични черти:

- хомогенна мултикомпютърна изпълнителна платформа в интранет - обикновено без апаратна специализация на възлите (евентуално функционална специализация) – симетричен мултикомпютинг

- обща администарция и защита; вътрешно отворен достъп

Обикновено за специализирани приложения важат:

- надеждна обработка (high-availability clusters) – репликиране/дублиране на приложенията

- балансирани разпределени сървери (load-balancing clusters) – най-често в комуникациите – общ портал, функционална хомогеннист – за бързо обслужване

- разпределен сървер данни

- сърверна ферма

«Фермите» (computer farms, server clusters) са къмпютърни клъстери

със специализирани сърверни функции на различни сърверни възли или модули (вкл. апаратна специализация) – напр. числова обработка (разпределена с работни станции или паралелна с минимашини), дисково поле, комуникационен сървер и др.

В един административен домейн и в обща високоскоростна опорна мрежа; За интензивна числова обработка или уеб-хостинг

**Метасистеми**

метасистемите (Legion, Condor, SNIPE) – осигуряват достъп до

хетерогенни разпределени ресурси.

Потребителят изпраща прозрачно заданието към виртуална разпределена система без да специфицира отделни компоненти – виртуален уникомпютър.

Предимно се прилагат в LAN системи отколкото WAN (проблемна scalability)

Oтсъства концепцията за потребителска услуга освен предоставяните системни функции и поради това са приложими само в професионални среди. Слаба стандартизация. Проблемна преносимост – дилемата CORBA/Java (+Jini/Jxta)

**Грид системи**

Раслоена програмна среда от за достъп до услуги, базирани върху разпределени и мрежови ресурси, които са споделени

***между различни институции.***

Кординирано споделяне на програмно-апаратни ресурси в динамична ***виртуална организация*** (9.7)фиг.

Списък характеристики (checklist) за особеностите на грид-средите:

- споделени ресурси

- без централизирана администрация

- различна административна принадлежност на потребителите

- системните протоколи и интерфейси са стандартни, отворени и с общо предназаначение (без орентация към конкретен клас приложения)

-по-високи функционални (напр. метауслуги) и нефункционални (напр.

производителност, отказоустойчивост..) изисквания към предоставяните услуги, отколкото в класическите РС

**Грид архитектура (9.8)фиг.**

универсални потребителски услуги –

- идентификация(authentication) и допуск (authorization)

- създаване на процеси (и нишки) с контролиран достъп до ресурсите - разпределено управление на свързаните процеси: хоризонтално – единно адресно пространство обща памет или MOM; вертикално – комуникиращи процеси: конвейризация, потоково управление workflow - комуникационни протоколи (принцип на пясъчния часовник върху [TCP]/IP ) системни функции:

- блокиране и освобождаване на заявени рисурси

- диспечеризация (scheduling, mapping, matchmaking, load balancing); - защита - управление на виртуалната памет (файлови системи, бази данни, планиране на активните страници в паметта)

- компютърно счетоводство и одит

**Open Grid Services Architecture**

OGSA – описателен стандарт за грид-архитектура на GGF за изграждане на сервизно-базирани гридове с

Web-обслужване развива концепцията на Web-услугите в посока на разпределено управление на ресурситена ВО в грид

прилага се от Globus Alliance

специфицира системните интерфейси и услуги основни интерфейси на OGSA ***регистрация и откриване*** на услуги (със сервизно представяне на ресурсите) – директория на услугите ***сервизни интерфейси*** – заредим код за достъп до услугите стандартен интерфейс за изграждане на ***сервизни области*** (комплексни услуги) от елементарни локални услуги интерфейс за ***дефиниране на цели*** (policies): за разпределеното и прозрачно администриране на натоварването (вкл. резервирането и алоцирането на ресурси), комуникациите (вкл. QoS), защитата

интерфейс за ***управление на данните***: разпределение, структура (БД/файлове), достъп, резервация,

транслиране, репликиране, описание и откриване, транзакции

основни системни услуги в OGSA

***мониторинг*** – наблюдение на ресурси и услуги – осигурява информация за разпределението, планирането (алокацията и ре-алокацията) на обслужващите процеси ***ресурсно управление*** (в OGSA clustering) – планиране на ресурси и услуги за оптимизация на обработката (load balancing, съгл. различни критерии), възстановяване след грешка (disaster recovery, seamless recovery), вкл. и за бартер на услуги/данни и др. ***потокова композиция*** – workflow – осигурява изпълнението на комплексни приложения като координирано изпълнение на набор от услуги – вкл. координация на съставните услуги, пренос на данните, интрфейси между

услугите ***одит*** на данни и услуги – за разервиране на данни и ресурси, за оценка (billing) на участието на

административните единици в ресурсите на ВО и за защита

***защита*** – супер-протокол, картиращ защитнте протоколи на различините административни областти в грида за постигане на прозрачна защита (принцип на най-слабото звено)

**Грид – слоест модел фиг. 9.10**

- инфраструктура (fabric) – ресурси за изчисление, запаметяващи устройства, мрежи, сензори и др.

- съобщителен слой (connectivity – “hourglass principle”) – малък брой протоколи, имплементирани за всички компоненти на инфраструктурата. Осигуряват комуникация и защита и в резултат – изпълнението на едно приложение в различните компоненти на инфраструктурата. Globus Toolkit реализира основно този слой и се състои от портоколи и APIs;

 - колективни (системни) услуги – състои се от услуги, протоколи и APIs, които осигуряват взаимодействието между ресурсите от инфраструктурата. Разширяват разнообразието от услуги чрез комбиниране на малкия брой протоколи от съобщителния слой.

Типични функции са

- директории/регистратори/браузъри на услуги и ресурси

- репликиране на данни и услуги - балансиране на изчислителния товар, - защита на достъпа и т.н.

Приложни услуги (applications) – потребителски и интерфейсни приложения (напр. портали), които се състоят от обръщения към модулите от долните слоеве.

**Globus Toolkit**

среда за изграждане на грид инфраструктура – отворен проект на Globus Alliance текуща версия 4

**Архитектура на грид приложенията – 9.13фиг.**

Традиционни грид приложения (без услуги) – развитие на C-S в

Интернет. Координиращият процес ползва грид инфраструктура – grid-aware Централизирано управление на достъпа през координиращия процес. Често централизирана БД за общия приложен контекст. Евентуално с Web интерфейси към локалните обслужващи процеси. Ограничен scalability и преносимост

**SOA**

- Бизнес и системните функции се поддържат от дискретни изолирани процеси, дефинирани съгласно платформения стандарт за услуги

- Услугите в SOA са с функционална спецификация – т.е. покриват функция или група от функции

прозрачно от инфраструктурата

грануларността им не се стандартизира (засега) – приложими са елементарни и

комплексни услуги; независими и прозрачни – принцип на черната кутия и компонентни технологии; публичност и откриваемост (advertising); проектират се със специализирани интерфейси за различини класове сървери, така че могат да се вграждат в съществуваща сърверна инфраструктура – SOA не изисква специализирана ифраструктура (освен при някои класове грид)

- интеграция на SOA-приложения:

потребителски слой (user interaction) – за [интерактивно] обслужване свързаност (connectivity) – междусервизен интерфейс интегриращ процес – междусервизно управление за “хореграфия” на потребителските сценарии интеграция на данните – федериране или пренос на данни

**SOA – реквизит**

- Web-услуги (9.16.1): използват се като устойчиви услуги в среда за

публикуване, търсене/откриване и изпращане на заявки към

съответната Web-услуга

- Web-базираните интерфейси (9.16.2): WSDL, SOAP, UDDI, XML

форми (вместо RMI, Jini, CORBA, DCOM) – за преносимост, модулност, разширимост и глобален достъп

типично върху НТТР като приложен протокол (tunneling) за универсално

разпространение в интернет (за преодоляване на защитни стени и прокси- процеси); но и други (SMTP)

- WSDL ( Web Services Definition Language) – за ***описание на услугите*** като [колекции от] точки

за достъп – т.е. мрежови портове и дефиниции на интерпретираните съобщения портовете се дефинират с мрежов адрес и типизация (абстрактна колекция от поддържаните операции от услугата, протокола на обмен и формáта на данните) съобщенията се дефинират чрез формáта на съдържанието им дефиниция на тип SOAP (Simple Object Access Protocol) – протокол за XML-спецификация на ***отдалечени заявки*** (вкл. обектни

аргументи – контекст) към обектни методи спрямо RPC/RMI

се различава по своята пълна платформена и езикова независимост въвежда по-голям комуникационен свръхтовар (поради усложнената нотация с марки (tags)) си прилича по фиксирания модел CS (не напр. P2P – поради транспорта върху НТТР)

структурира събщенията в обвивка/заглавие/тяло (envelope/header/body) клиентските имплементации се вграждат в Java, Python, Perl и MS.Net приложения, а UDDI- директориите се поддържат от повечето сървени платформи – Apache, SAP, Windows Server 2003, Oracle и т.н. - UDDI (Universal Description, Discovery and Integration Service) (9.17) – глобална платформено- независима ***директория на услугите*** (registry, service broker) с XML-записи в три дяла – бели, жълти и зелени “страници” респ. с адресна, бизнес- и техническа категоризация на

услугите – общо предназначение вкл. и за Web-услуги интерфейсът й обслужва SOAP-съобщения достъп до WSDL-описания на протоколите за свързване със съотв. услуга и на формáта на съобщенията, които се интерпретират от услугата

- XML-фомите: използват се във всички изброени стандартни протоколи, а също могат да бъдат използвани извън стандарта или в разширения като различни дескрипнивни езици (напр. за

описание на услуги като с WSDL)

**Моделно-базирани архитектури**

[MDA – Model-driven architecture]

- MDA представлява допълнение или средство за автоматизирано проектиране на приложения,

използващи обичайни архитектурни подходи като ОО, COTS или SOA. - Абстрахира се инфраструктурата (QoS) както и системните или общите функционалности – приложението се разработва на базата бизнес модел. - Детайлността на модела предопределя използването на автоматизирано проектиране

- широко се прилага формално специфициране с UML моделите - допускат прилагането на индустриални стандарти, както и въвеждането на стандартизация в моделирането - разработваните моделни стандарти (напр. от OMG – Object Management Group) всъщност формират SW-архитектура на базата на абстрактни (в смисъл на бизнес-ориентирани) модели

- спецификацията на моделните архитектури има предварителна моделна фаза – по което се различава от останалите SW-архитектури - стандартната спецификация на абстрактни бизнес модели позволява модела да съдържа изпълнима семантика и съответно прилагането на автоматизирана генерация на приложения

- моделната спецификация е платформенонезависима (PIM – platform-independent model) и е

ориентирана към пробламната област

- за генерация на изпълним код освен PIM е необходима и дефиниция на изпълнителните платформи на компонентите на модела (PDM – platform definition model: напр. CORBA, .NET, Web), при което моделът де транслира автоматично до Java, C#, PHP, Python

- автоматичната генерация на код при МБА може да се съчетае и с прилагане на различни други проектни подходи – най-често с ползване на програмни шаблони (design patterns)

**Аспектно-базирани архитектури**

[Aspect-Oriented System Architecture]

SW-архитектура, изградена в съотвествие с принципите на аспектно-базираното проектиране (AOSD – Aspect-oriented software development) “***аспекти***” (или още ***crosscuts***) на разпределеното приложение са общи функционалности, които се прилагат от няколко от специализираните (и централни) бизнес-модули напр.: дневник (log), оторизиране и защита поради това декомпозицията т.е. модулността на “аспектните” функционалности

представлява проблем с много варианти на решение основен проблем е елиминирането (или стройната организация) на разпръснати и объркани

спецификации на “аспекти” поради разделянето на тези общи функционалности между

модулите, които специфицират централните бизнес-функции

обект на разработка за изграждане на цялостен архитектурен подход са стандартизацията, модуларизацията и шаблонизацията на различни категории функционални и

нефункционални “аспекти”

за целта се използват специализирани езици (aspect oriented languages) и платформи (тъй като аспектите не могат – или не е технологично – да бъдат модуларизирани с обичайните

архитектурни абстракции – CS, p2p, OO, шаблони); тези средства подпомагат проектирането

или ре-дизайна на общата архитектура с оглед на “чисто” аспектно проектиране AOSA се фокусира на идентифицирането, локализацията (или картирането) и

спецификацията на аспектите в явен вид на етапа на проектиране на SW архитектурата

**Софтуерни агенти – 9.20фиг.**

Автономни (самоинициативни) процеси, изпълняващи задания съвместно с други [отдалечени]агенти; Свойства:

**- автономност и активност** – самоинициатива за въздействие върху дадени параметри на средата - **реактивност** – сканиране (с евентуална реакция) на средата

**- комуникативност** – интерфейс/и към потребители и други агенти – по-специално съвместни агенти

- непреходност (continuity) – продължително и многократно изпълнение на функциите, не за всички типове агенти

- мобилност – миграция между възли (може и при ниска мобилност на кода), не за всички; напр. агенти за извличане на информация от разпределени документни системи

- адаптивност – еволюция на реакциите при еднакви параметри на средата, не за всички

Изолирани класове агенти (освен общия клас Софтуерни агенти): интерфейсни – потребителски достъп до приложения и други агенти с адаптивност (самообучение); напр. брокерски агент информационни – подобно на интерфейсните – за управление на постъпваща информация

**Технология на софтуерните агенти**

среда за поддържане на агенти (агентна платформа) – предоставя

основните свойства на агентите като middleware-услуги

FIPA модел на агентна платформа (на Foundation for Intelligent

Physical Agents) – обслужва създаването и закриването на агенти, междуагентните комуникации и откриването на агенти агентната платформа за разпределена система включва модулите (фиг. 9.21.) за управление (създаване, идентификация, съединение) за директория (локална) – таблица иеднтификатори – атрибути; достъпна за отдалечени агенти

комуникационен канал – АСС (Agent Communication Channel) – обмен на

съобщения (подредени и проверени) между агнтските съединения напр. По Internet Inter-ORB Protocol (IIOP – при хетерогенни МОС)