

Система за отдалечено изпълнение на приложения и услуги в йерархичен грид (резюме)

Настоящата дипломна работа представлява част от проекта GrOSD за разработка на лек грид мидълуеър. Изследванията и разработката на грид платформата GrOSD (Grid-aware Open Service Directory) се извършват по проекта СУГрид на Факултета по Математика и Информатика към СУ "Свети Климент Охридски", както и в партньорство с европейският изследователски проект CoreGRID (European Network of Excellence CoreGRID). Основната цел е изграждането на лека грид инфраструктура, като приоритет са опростената архитектура и реализация.

Тема на дипломния проект е разработката на системен модул за отдалеченото изпълнение на приложения и услуги, които в контекста на проекта GrOSD се установяват и работят във възлите на посочената грид-система. С помощта на този модул се оползотворява предоставения за нуждите на грида хардуерен ресурс за реализирането на разпределени изчисления. Основна функционална черта на модула е да осигури инфраструктурата за прехвърлянето до възлите, изпълнението и връщането на резултата за, практически, произволни приложения, написани от клиентите на грида, без да им бъдат поставяни специални изисквания за формата на кода, както е в редица други грид-системи. Акцент в разработката са два типа на обслужване – задания с еднократно извикване и такива, които имат по-продължително присъствие на възлите и към които се адресират многократни обръщения с различни входни данни. Същевременно, модулът се характеризира с гъвкава архитектура, която, от една страна, позволява неговото използване в рамките на различни системи, а от друга страна, е предпоставка за неговото разширяване в посока изпълнението на разнообразни и по-специализирани типове задания, които да посрещнат нови специфични нужди. За този модул се използва наименованието Node Service (или съкратено NS) в съответствие с приетата терминология на платформата GrOSD.

Първата глава от изложението акцентира върху основната приложна област на Node Service, а именно грид архитектурата и конкретни имплементации на грид системи. Тази глава разглежда по-подробно леките грид-системи, като се спира на основните им характеристики и технологиите за тяхното изграждане.

Във втора глава моделът на грид се конкретизира в контекста на проекта GrOSD, като се прави обстоен анализ на структурата на GrOSD и детайлно разглеждане на съставлящите го модули и обзор на техните функции и взаимодействия. Въвежда се нуждата от реализацията на услугата Node Service като функционално ядро за изпълнението на изчислителните услуги, които предлага GrOSD за своите клиенти. Посочват се общите изисквания

към изпълнителния модул – разширяемост на модула към изпълнението на нови типове услуги, адаптируемост към промени в мрежовата система, отказоустойчивост на грешки при изпълнението, пълноценно използване на предоставените ресурси на възела и ясно дефиниран канал за взаимодействие и комуникация със зависимите от модула централни сервиси в системата и други негови директни клиенти. Node Service е сравнен с аналогични изпълнителни модули в някои други мрежови системи, като във фокуса на сравнителния анализ се поставя специализираният формат на приложенията, предназначени за тези системи.

Трета глава е посветена на детайлен и обстоен анализ на трите компонента, съставляващи Node Service – *приложният програмен интерфейс (API)* за формулирането и изпращането на заявките до възлите на мрежата; *изпълнителното ядро*, което се разполага в самите възли и е отговорно за стартирането на получените задания и събирането и обратното изпращане на резултатите; *платформената подсистема за задания (task framework)*, която играе ролята на свързващо звено между другите два компонента, като дефинира общите термини, с които те боравят и се разбират.

Платформеният модул представлява съвкупност от интерфейси. Чрез тях се очертава концептуалният и структурен модел на заявките за изпълнение, които се конструират с конкретни имплементации в API-то откъм клиентска страна, и се поемат и обслужват от изпълнителното ядро от страна на възлите. Също така, в този модул се намират централният интерфейс на Node Service, който декларира кръга от предлагани услуги за директно ползване от API-то, както и интерфейсите на мониторинговите обекти и събитията от изпълнението, за които клиентите на API-то могат да бъдат уведомени. Уведомяването им става асинхронно спрямо подаването на заявката за изпълнение, което означава, че клиентите не са блокирани до получаването на резултата.

Ролята на Node Service API-то е да служи като фасада за услугите, осигурени от изпълнителното ядро. Основните му функции са две – 1) да предостави конкретни реализации на платформените интерфейси, както и инструментирани класове за съставянето на елементите на заявката и на самата заявка; 2) да осигури прозрачен достъп за клиента на услугата до функциите и ресурсите на Node Service, работещ на конкретен възел, базирайки се на възможностите на технологията JavaRMI.

Изпълнителното ядро на Node Service е натоварен с логиката да обработи получените метаданни на заявката, формулирани според спецификациите на платформената подсистема, за да състави реалното задание, след което да го извика и да събере и изпрати резултатите. Основно се разчита на възможностите на Java Reflection API. Особености в имплементацията на ядрото представляват: 1) *специализираните процесорни модули*, към които вътрешно се разпределят отделните типове заявки; така архитектурата на ядрото става разширяема за обслужването на нови видове задания, което е възможно чрез простото добавяне на съответния процесор за тях; 2) изпълнението на заданията в собствени *нишки* – по този начин ядрото може да обслужва няколко заявки едновременно; 3) централизиран контрол над изпълнителните нишки чрез специално организиран *регистър* на заданията и услугите; 4) *персистентен модел* на клиентските задания за многократно стартиране на възела с различни подадени входни данни; персистентността

се постига чрез запазване на метаданните на заданията в отделни файлове, от които могат да се възстановят след рестартиране на ядрото.

С четвърта глава се илюстрира приложението на услугата в два сценария – 1) в опростен вариант, извън контекста на работа в грид – при него клиентската програма директно използва възможностите на Node Service API-то за създаването на заявки и тяхното зареждане за изпълнение; 2) детайлите по работата на Node Service в интегриран вид с услугите в GrOSD, с фокус върху взаимодействията с тях – в този случай клиентите на Node Service и техните функции се конкретизират в контекста на GrOSD.

Петата глава дава кратки указания по инсталирането на Node Service от страна на възела и на клиента, което е изключително опростено поради единственото изискване за неговата работа – JDK версия 5.0 или по-висока. С това се гарантира максимално широк кръг от потенциални донори-възли за работата на услугата.

Шестата глава дава детайли за използваните технологии, за които основно изискване е да са от некомерсиално естество. Набляга се върху възможностите на Java и JavaRMI, които осигуряват мобилност на кода и прозрачна комуникация, върху Java Reflection API, чрез който става възможно обработката на произволен компилиран клас, и върху шаблоните за проектиране, които дават ефективни решения за проблемите, възникнали по време на разработка на модулите на Node Service.

Последната глава от изложението е специално посветена на разширяемостта на услугите, предлагани от Node Service, посредством гъвкавостта на нейната архитектура, капсулация на конкретните имплементации на елементите на заявката в API-то и специализация на модулите за обработка в изпълнителното ядро. Очертават се широките перспективи за развитие на Node Service за нуждите на GrOSD и се прави прогноза за възможностите му за модификация за самостоятелна работа в peer-to-peer мрежи, при която се поемат част от функциите на някои централни услуги в GrOSD.

В приложението е поместен изходният код на клиентска програма, която илюстрира нагледно начинът на работа с Node Service API-то по отношение на конструиране на заявка, изпращането ѝ към определен възел и начина за получаване на резултата.