



## UvA-DARE (Digital Academic Repository)

### Hierarchical resource management in grid computing

Korkhov, V.V.

**Publication date**

2009

**Document Version**

Final published version

[Link to publication](#)

**Citation for published version (APA):**

Korkhov, V. V. (2009). *Hierarchical resource management in grid computing*.

**General rights**

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

**Disclaimer/Complaints regulations**

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

# Samenvatting

Het in dit proefschrift beschreven onderzoek behandelt het probleem van resource management in complexe en meervoudig gelaagde gedistribueerde applicaties. Het voornaamste doel is de bestudering van parallelisme op alle niveaus in dergelijke applicaties, en het definiëren van nieuwe methoden voor resource management en load balancing (het evenredig verdelen van de werklast) die niet afhankelijk zijn van specifieke gedistribueerde rekentechnologieën, en die toegepast kunnen worden op alle niveaus in de applicatiehiërarchie.

Na een inleidende verhandeling over technologie-invariante methoden voor resource management in complexe parallele applicaties die draaien in heterogene gedistribueerde systemen, introduceren wij onze methode van adaptieve workload balancing (AWLB), die toegepast kan worden in een grote verscheidenheid aan computeromgevingen. De AWLb methode is afhankelijk van de dynamische karakteristieken van zowel de applicatie zelf als van de omgeving waarin deze applicatie draait. Traditioneel bestaan er twee verschillende load balancing strategieën voor parallele applicaties: (1) bepaal vooraf grondig de verdeling van de werklast, rekening houdend met de specifieke aspecten van zowel de applicatie als het onderliggende systeem, en (2) verspreid de werklast op een simpele, voor de hand liggende manier, hooguit rekening houdend met de relatieve rekenkracht van elke deelnemende computer. De eerstgenoemde methode is tijdrovend, en vereist expertkennis van zowel de complete applicatiestructuur, alsook van de specifieke details van de toegepaste algoritmieken. De tweede methode is veel sneller, maar niet efficiënt voor wat betreft de behaalde rekensnelheid. Wij stellen daarom een middenweg voor die is gebaseerd op een snelle eerste benadering, gevolgd door een iteratieve verfijning in de richting van het optimum.

Gegeven de complexiteit van applicaties, en de vele manieren om complexe applicaties op te delen in een gelaagde hiërarchie, beschouwen we een structuur voor parallelisme en distributie die bestaat uit drie niveaus: (1) het taakniveau, d.w.z. een enkele parallele applicatie, (2) het jobniveau, d.w.z. een verzameling taken, bijvoorbeeld een parameter sweep, en (3) het workflowniveau, d.w.z. een functionele decompositie van de applicatie. Op elk van deze niveaus moet aan specifieke eisen worden voldaan om de rekensnelheid te maximaliseren, en om schaalbaarheid te garanderen. Bijgevolg moet het probleem van load balancing en resource management in beschouwing worden genomen tijdens het creëren van een applicatie als geheel.

Hoofdstuk 3 behandelt de onderste laag in deze complexe applicatiehiërarchie: een enkele parallele applicatie draaiend op een heteroogeen systeem. Eén van de grootste uitdagingen in het overbrengen van een parallele applicatie van een homogene clusteromgeving naar een heterogene verzameling van computers is het verkrijgen van een vergelijkbaar niveau van parallele efficiëntie. Voor de oplossing van dit probleem hebben we een theoretisch raamwerk ontwikkeld, tezamen met een generieke workload balancing techniek die rekening houdt met de specifieke parameters van zowel de applicatie als het onderliggende systeem. We introduceren hier de AWLb methode, en passen het toe op één van de sturende applicaties van dit onderzoek: een parallele implementatie van de Virtuele Reactor.

Het geval van een enkele parallele applicatie die wordt geëxecuteerd op een verzameling van parallele computers wordt beschreven in Hoofdstuk 4. De beschikbaarheid van een groot aantal rekensystemen roept de vraag op of, en wanneer, het gebruik van meerdere parallele computers voor een enkele parallele applicatie een versnelling kan betekenen. Een theoretische methode tot het schatten van de mogelijke versnelling van een parallele applicatie in een homogene Grid omgeving is gegeven door Hoekstra en Sloot in [49]. Hier valideren we

onze aanpak op basis van een bestaande applicatie: een Lattice Boltzmann Equation (LBM) solver. We gebruiken een eenvoudig model van deze applicatie voor de schatting van de snelheidswinst die mogelijk te behalen is op een multicluster systeem. In vergelijking met onze oplossing voor parallele applicaties draaiend op een heterogeen systeem (in Hoofdstuk 3) is de aanpak die in dit hoofdstuk wordt gepresenteerd toepasbaar op homogene Grids. De aanpak is vrij accuraat in het voorspellen van de snelheidswinst van applicaties, op basis van een simpel applicatiemodel en een eenvoudige verzameling van systeemkarakteristieken.

Het volgende niveau in de applicatiehiërarchie wordt behandeld in Hoofdstuk 5. In applicaties die uit een aantal gedistribueerde taken bestaan (ofwel: multi-job applicaties) is het rekenwerk georganiseerd als een verzameling aparte componenten die alleen informatie uitwisselen aan het begin en aan het eind van de executie. In dit hoofdstuk wordt de AWLB methode toegepast op multi-job applicaties waarvan de werklast verder op te splitsen is. We presenteren een hybride resource management omgeving, werkend zowel op applicatieniveau als op systeemniveau, die de executietijd van dergelijke applicaties in heterogene Grid systemen minimaliseert. Deze omgeving bestaat uit op applicatieniveau toegepaste AWLB die wordt geïntegreerd met zogenaamde User-Level Scheduling (ULS). Deze laatste component dicht het gat tussen de applicatie zelf en de Grid resource managers: het is een door de gebruiker aan te passen middleware systeem voor het scheduleren van applicaties. Het onderliggende systeem wordt dynamisch gemeten en applicatiekarakteristieken worden bij voortduring geschat ter optimalisatie van het gebruik van de door ULS gecontroleerde dynamische verzameling van Grid computersystemen. Ter validatie hebben we experimenten uitgevoerd met een synthetische en configureerbare applicatie, gebruik makend van het EGEE Grid systeem, de AWLB methode, en de DIANE user-level scheduler. Op basis van onze resultaten bespreken we verschillende manieren voor het managen van de werklast van splitsbare multi-job applicaties op het Grid. Bovendien vergelijken we verschillende methoden voor werklastdistributie, en illustreren we het gebruik van een dynamische set van computers onder dynamische variatie in het applicatiegedrag door middel van adaptieve selectie van computerresources.

Hoofdstuk 6 behandelt de bovenste laag van onze applicatiehiërarchie: de Grid workflow. Onze voornaamste aandacht gaat uit naar workflows die worden gestuurd door een dataflow, en we evalueren verschillende resource management strategieën voor dit type dataflow. Voor het executeren van een gedistribueerde workflow op het Grid is een workflow management systeem (WMS) benodigd dat de controle heeft over de executie. In dit hoofdstuk presenteren we VLAM-G en het belangrijkste kerncomponent, het Run-Time Systeem (RTS), als een implementatie van een data-gestuurd WMS. Het RTS maakt gebruik van Grid resources, terwijl het de inherente complexiteit van het Grid voor gebruikers verbergt. We behandelen de architectuur, de verschillende onderdelen van het RTS, en de speciale mogelijkheden van VLAM-G. Speciale aandacht wordt besteed aan het concept van dataflow, en het direct laten vloeien van data tussen verschillende gedistribueerde workflow componenten. Als bovenste niveau in de applicatiehiërarchie kan een workflow de management van de verschillende onderliggende niveaus verzorgen, alsook de coördinatie van de executie van elk component in samenhang met alle andere. Deze mogelijkheden van het VLAM-G WMS wordt geïllustreerd aan de hand van een workflow voor de Virtuele Reactor applicatie, die relevante onderdelen bevat op alle niveaus van onze applicatiehiërarchie.

Het in dit proefschrift gepresenteerde werk behandelt de gedistribueerde executie van meervoudig gelaagde applicaties die bestaan uit meerdere complexe componenten. Voor alle niveaus in de applicatiehiërarchie, van laag tot hoog, beschrijven we de meest geschikte methoden voor resource management. We presenteren onze implementatie alsook een experimentele evaluatie van de voorgestelde methodieken.