

Arhitectura pentru managementul calitatii serviciilor (qos) în sisteme distribuite

Asist. Cosmina IVAN

Catedra de Calculatoare, Universitatea "Babes Bolyai"

Lect.dr. Lucia RUSU

Catedra de Informatica Economica, Universitatea "Babes Bolyai"

Lucrarea își propune descrierea problemelor managementului calitatii serviciului (Quality of Service - QoS) în sistemele distribuite și este propusa o arhitectura pentru monitorizare automata și control operational al unui asemenea sistem. Modelul mapării taskurilor la resurse permite construirea de clusteri de resurse și taskuri. Arhitectura propusa și metodologia adoptata a fost testata pe un sistem WEB.

Cuvinte cheie: managementul calitatii serviciilor (Quality of Service), arhitectura pentru management, monitorizare și control operational, managementul, aplicatiilor distribuite, managementul sistemelor WEB, clusteri, AD (aplicatii distribuite).

Introducere.

Un mediu distribuit cuprinde calculatoare, LAN-uri și WAN-uri, periferice, diverse sisteme de operare, middleware și aplicatii, unde de regula resursele sunt partajate. De ace- ea, managementul lor este o sarcina deloc usoara. În ultimii ani s-au adus contributiile remarcabile la managementul AD în domeniul analizei QoS pentru aplicatii multimediale. Lucrarea propune o arhitectura pentru managementul AD (aplicatiilor distribuite) incluzând managementul, administrarea, sistemele de masura și control interconectate pentru a genera monitorizarea și managementul fazelor ciclului de viata al unui AD.

Lucrarea definește în acest scop un set de modele de procesare a cererilor utilizatorilor și modele de resurse pentru a putea construi clusterii de resurse și taskuri.

Modelare

Utilizatorii de aplicatii gândesc în termeni de tranzactii de nivel înalt și nu în termeni de swichuri, computere, rețele și încarcarile lor. Userul percepe QoS cu care sistemul îi returnează un raspuns la solicitarea sa. În

exemplul din figura 1 cerintele de QoS ale utilizatorilor se refera la timpul de raspuns pentru diverse cereri adresate serverelor remote.

Timpul de raspuns contine:

- Timp consumat la client pentru procesare cerere și pregătire pentru transmitere
- Întârzierea rețelei la livrarea cererii;
- Procesarea cererii și generarea raspunsului la server;
- Întârziere la livrare raspuns;
- Procesarea raspunsului la client.

Constrângerile asupra timpului de raspuns introduc constrângeri asupra parametrilor diferitelor subsisteme ce procesează cererea (Ex. Pentru rețea – performanța rata de erori, pentru server- viteza CPU, memoria).

Problema eliminării de către administrator a cauzelor degradării QoS este deosebit de complexă datorită complexității AD și a sistemelor care pot fi eterogene, mai mult chiar pot opera în configurații specifice, ce nu pot fi apriori evaluate.

Mai mult decât atât, resursele sunt în, general, partajate între aplicatii și useri astfel încât nu

este suficient sa se analizeze o singura aplicatie sau un user al unei aplicatii.

Arhitectura pentru management

Elementele unei arhitecturi pentru managementul AD sunt redat e  n figura 2. Pentru oricare din taskurile generale (instalare, monitorizare QoS, diagnoza), se pot detecta de-

pendente  ntre parametrii QoS pe divese niveluri ale arhitecturii, alaturi de proprietatile lor interpretabile specific pentru un anumit task de management.

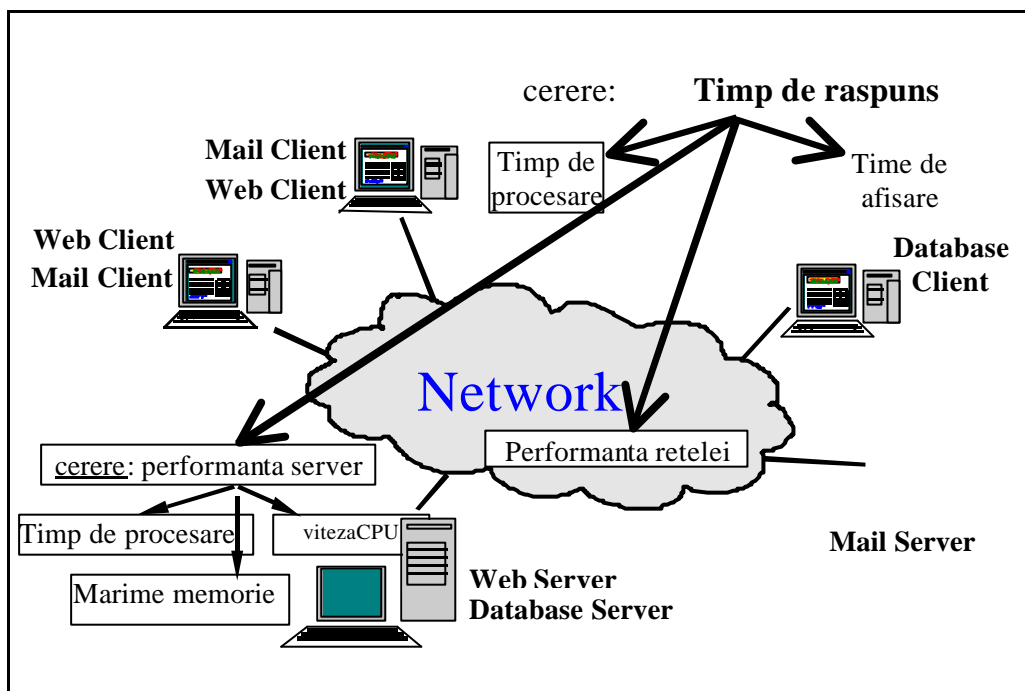


Fig. 1. Cerinte QoS si partajare resurse

O sarcina precisa transmisa de *sistemul administrativ* va construi o schema de monitorizare, astfel ca solicitarea exprimata  n parametrii de nivel  nalt va putea evaluata de *sistemul de masura*. *Sistemul de management* este responsabil pentru decizia de alege a parametrilor de masurat, c t si de informarea userului. Maparea (trasformarea din parametrii masurabili  n parametrii de nivel  nalt) o realizeaza *model builderul*. Informatiile astfel construite reprezinta suport de decizie pentru sistemul administrativ. Acesta va transmite corespunzator comenzi catre sistemul de management, care va construi o schema operationala de control (*Analizorul* detecteaza care parametri cauzeaza degradarea QoS si folisind *model builderul*  i mapeaza la cei

controlabili gener nd setul de valori necesare). Sistemul de control este cel care realizeaza ajustari prin primitivele sale, regl nd executia aplicatiei prin implementarea de mecanisme de mentinere dinamica ( n timp real) a QoS. Monitorizarea QoS presupune trasarea nivelurilor QoS (parametrilor) pentru o aplicatie si compararea lor cu cele cerute. Schema presupune o ierarhizare a modelarii sistemului. S-a implementat astfel un mecanism pentru construirea unei scheme de monitorizare. Controlul operational rezolva problema detectarii acelor parametri controlabili a caror variatie permite schimbarea parametrilor monitorizati care pot fi astfel readusi la valorile dorite. Analizorul genereaza o schema de monitorizare, select nd parametrii controlabili

necesari mentinerii QoS. Daca nu se poate atinge scopul propus de sistemul de management, va apela model builderul spre a genera un nou model cu o alta decompozitie a

parametrilor. A fost cre-at un algoritm de control operational bazat pe modelarea descrisa si utilizat de sistemul de management.

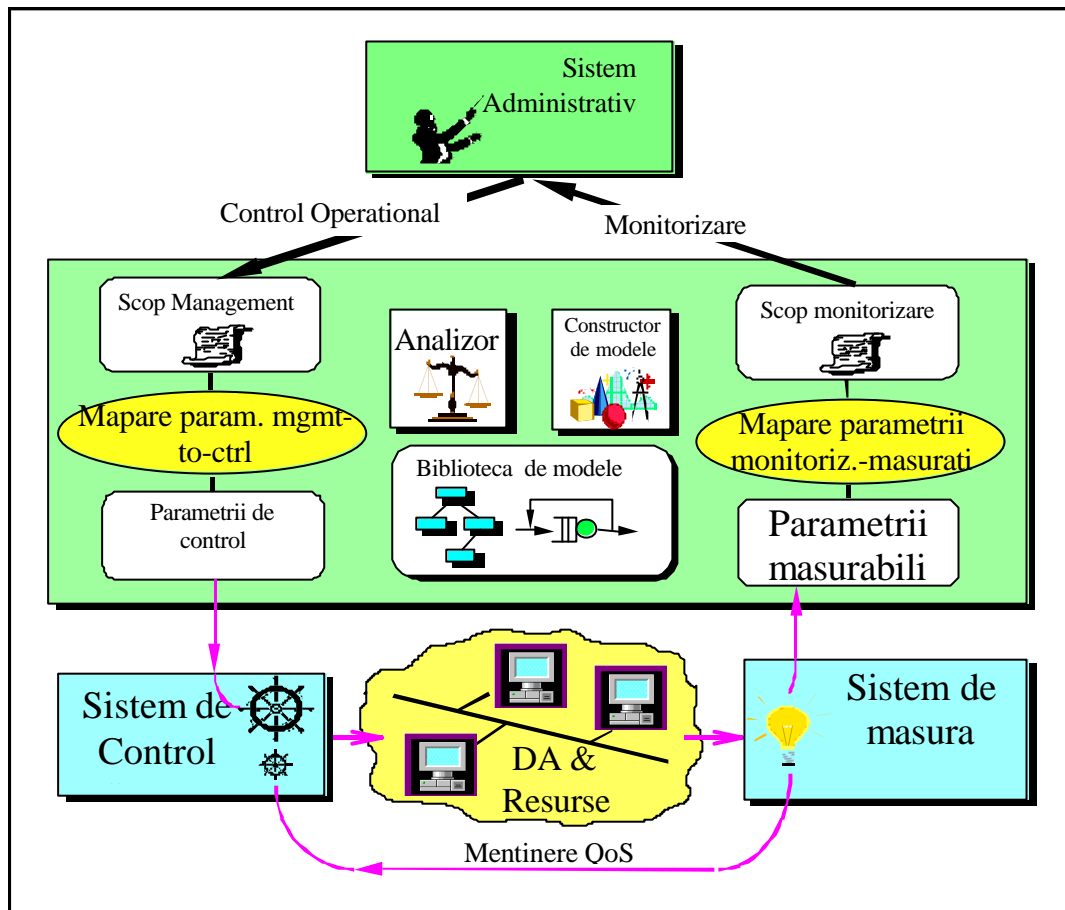


Fig. 2. Arhitectura pentru managementul AD

Model builder

Modelul user request processing (T-model) folosește notatia grafului orientat pentru a descrie interconexiunea taskurilor și secvența de procesare a cererii utilizator. Modelul este ierarhic (figura 3).

Modelul resurselor (R-model) conține componente client, componente server, sistemul de transport, CPU, disk, memorie. Fiecare task are alocată o resursă la execuție.

Semnificația notațiilor este următoarea: Cl-Rq - taskul ce pregătește cererea de sistemul client; Cl-Dsp - taskul afisării răspunsului de către sistemul client; Tr-S - taskul de transmisie; N-Rq - task de transmisie cerere client de către

rețea; N-Rs - taskul de ras-puns a serverului la transmisia prin rețea; S-Rq - taskul cererii client procesată de server; C-Rq - taskul recepționării cererii client de către server; C-Rs - taskul răspunsului la ce-rere trimis de server; Client - sistemul client; Net-S - sistemul serviciu (rețeaua și siste-mul server); Server - resursele calculatorului server; CPU - resursa CPU la server; Disk -resursa disk la server.

Cele două modele {T-model și R-model} pot fi concepute folosind diferite niveluri de decompozitie, pentru diferite taskuri și resurse.

Modelarea mapării taskurilor la resurse.

Conceperea modelului TR-model presupune o distributie a taskurilor la resurse, generând așa numitii clusteri de resurse și procese (figura 4). Taskurile mapate la aceeași resursă aparțin aceluiași grup, apoi toate resursele la care taskurile grupului sunt mapate sunt incluse în același grup, alte resurse adăugate care au

taskuri definite și mapate lor vor fi adăugate grupului. Rezultatul acestui proces este un cluster de taskuri și resurse interdependente (un task poate utiliza mai multe resurse și o resursă și poate fi utilizată de mai multe taskuri).

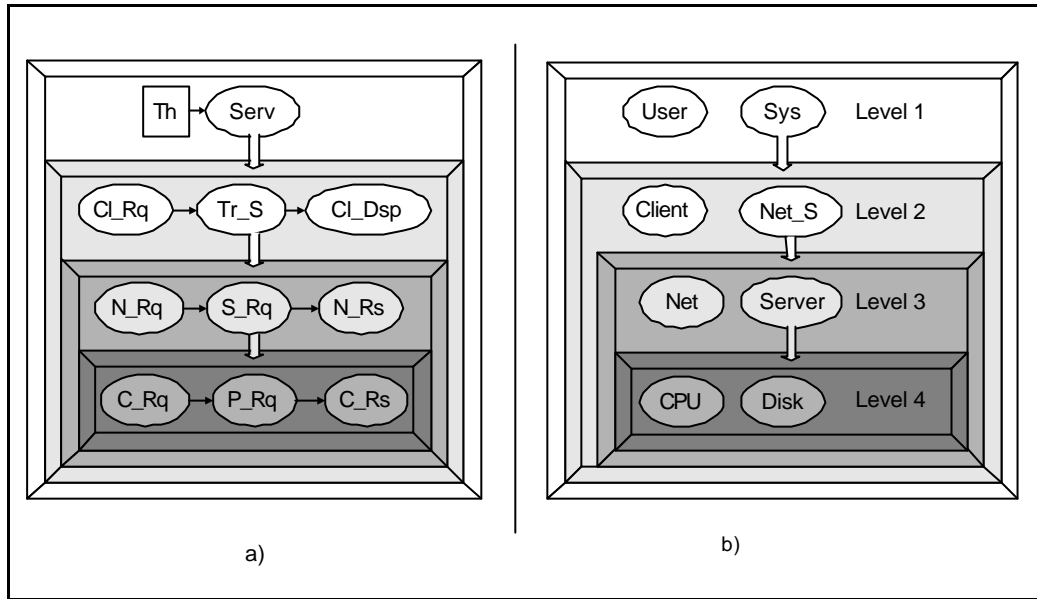


Fig. 3. Descompunere procese pentru modele a) taskuri și b) resurse

Metoda propusă aduce următoarele îmbunătățiri (față de metoda clasică a nivelurilor)

- construirea și execuția modelului de taskuri ierarhice diferă de o modelare ierarhică software;
- modelul TR permite construcția unui set de modele separate dar interdependente mo-dele

de performanță pentru sisteme AD. Se pot construi seturi de astfel de modele de mapări considerând diferite niveluri de decompunere a taskurilor și resurselor;

- Pot fi selectate din aceste seturi modele de performanță utilizabile în sistemul de management.

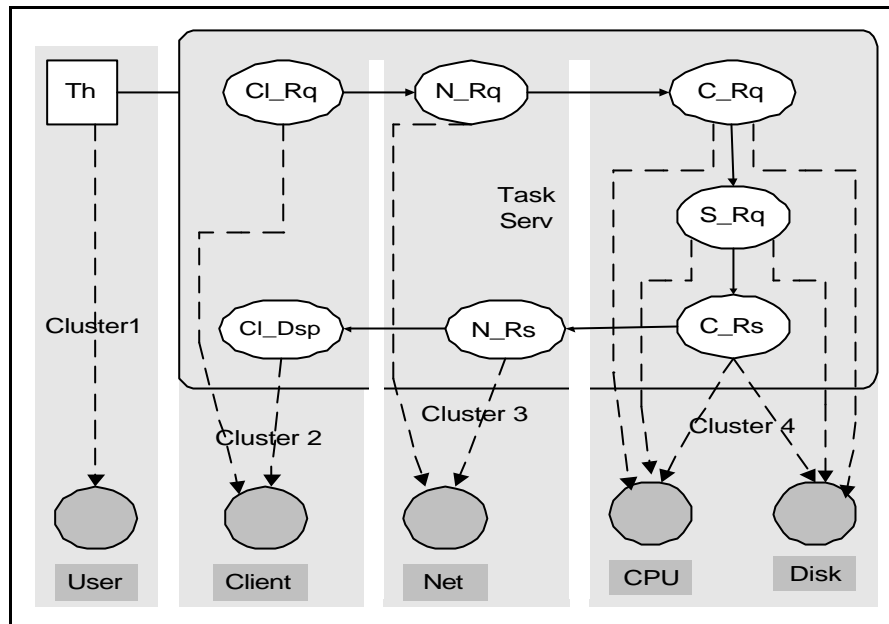


Fig. 4. Maparea taskuri-resurse

Experimentare.

Arhitectura propusa într-o structura mini-mala s-a aplicat managementului unui sistem WEB. S-a modelat astfel procesarea unei cereri utilizator de tip HTTP pentru predictia

raspunsului sistemului WEB, analizând timpul de raspuns pentru diferite latimi de banda si niveluri de concurenta (figura 5). Rezultatele experimentului de-monstreaza ca modelarea propusa poate fi folosita pentru predictia QoS.

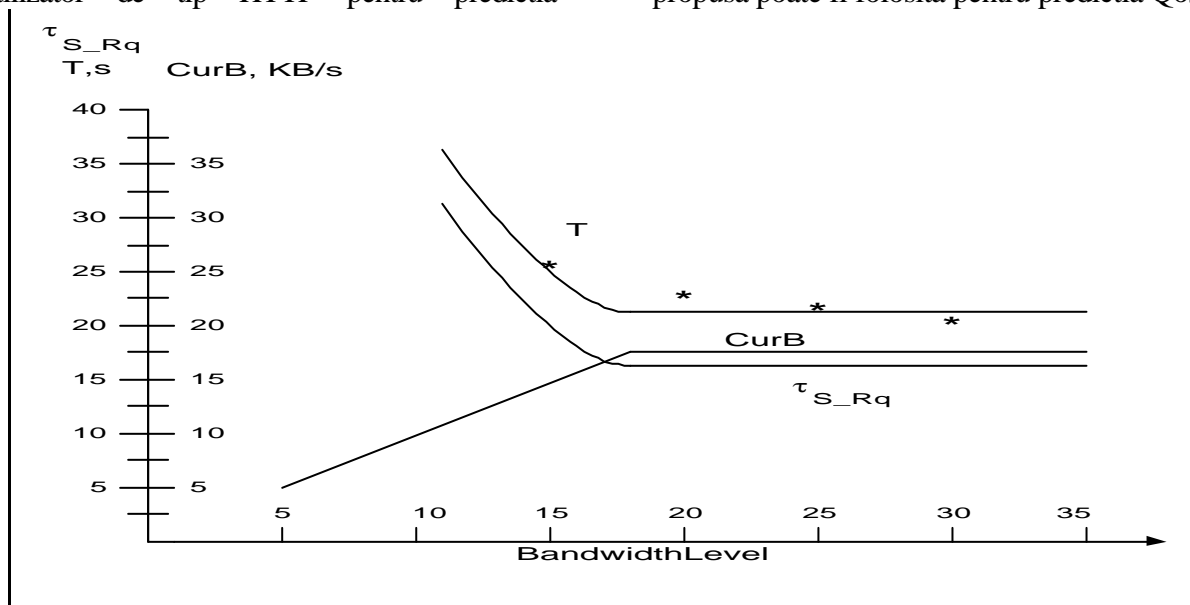


Fig. 5. Caracteristicile performantei sistemului WEB

Concluzii

Lucrarea concepe o metodologie pentru managementul QoS al sistemelor distribuite (al aplicațiilor distribuite, al caror aplicabilitate

devine). Sunt descrise elementele esențiale ale unei arhitecturi cu detalierea elementului cheie, constructorul de modele (Model Builder). Au fost implementați algoritmi pentru o schema de

control operational, si pentru monitorizare. Lucrarea reprezinta un punct de plecare pentru cercetari viitoare pri-vind dezvoltarea schemelor de monitorizare si control.

Bibliografie

[1] Campbell, A., Coulson., G. (1996). A QoS Adaptive Transport System: Design, Implementation and Experience. Proc. ACM Multimedia 96, Boston, pp. 117-127.

[2] Dermier, G., Fiederer, W., Barth, I., Rothermel, K. (1995). A Framework for Negotiable Quality of Service in Distributed Multimedia Applications. Tech. Report, Fakultätsbericht, No. 10, Universität Stuttgart, p. 19.

[3] Iqbal, M.A., Hagin, A. (1996). Partitioning and Mapping Techniques for Distributed Multimedia Applications. Tech. Report, Fakultätsbericht, No.14, Universität Stuttgart, p. 23.

[4] Hafid, A. (1995). *Hierarchical Negotiation for Distributed Multimedia Applications in a Multi-Domain Environment*. Second Workshop on Protocols for Multimedia Systems "Mozart on Multimedia Highways", Salzburg, Austria, pp. 397-409.

[5] Koch, T., Krämer, B. (1996). Rules and agents for automated management of distributed systems. Special issue of the Distributed Systems Engineering Journal on Distributed Systems Management, The Institution of Electrical Engineers and Institute of Physics Publishing.