Response Time Aware Coordination in Multi Agent Filtering Framework

Sahin Albayrak, Dragan Milosevic
DAI-Labor, Technical University Berlin
{sahin,dragan}@dai-lab.de

Abstract

In nowadays commercial and information reach society, filtering strategies have to be combined respecting the availabilities of resources, and additionally the guaranties regarding the response time should be given. The essence of the presented solution is both in the encapsulation of many known searching algorithms inside separate filtering agents, and in the integration of response time aware coordination mechanisms into one manager agent. Experimental results show that the guaranty of always proving results within 100 seconds can be given without sacrificing a user satisfaction.

Introduction

A real environment, being the assumed playing ground for the coordination among available filtering strategies [2], creates a challenge that is contained in the highly changeable availability of needed system resources. It is far away from being truth that the load of system resources, such as CPU, database and memory, can be assumed to be static. On the other side, the existed filtering strategies usually differ a lot concerning their requirements towards needed resources. The selection of strategies, for which not enough resources are available, is probably the bad move that will hardly provide good results in a reasonable amount of time.

In the case where strict guaranties about response time are additionally requested, it is not enough to only promise the reasonable duration of filtering. One has to be able to say in advance after how many seconds the results will be ready, or what is even harder, to always produce results in the predefined amount of time. There are hopefully many algorithms, such as hill climbing, and simulated annealing [2], which can be stopped at any time. These any time algorithms will then return as recommendations the best results that are found before the stopping has occurred. A coordination mechanism should ensure that at least one such any time strategy is selected, and then it can easily guaranty the provisioning of results in the given time slot.

Approach

While requirements towards different resources are modelled through CPU (F_{CPU}), DB (F_{DB}) and memory (F_M) fitness values as in [1], the response time awareness is represented through any time (F_{AT}) fitness value as:

Def. 1. *Any time fitness* F_{AT} corresponds to the ability of a strategy to deliver results whenever it is stopped. It takes only $\{0,1\}$ values, where value $F_{AT} = 1$ says that the corresponding strategy can be stopped at any time, and afterwards be asked for results. When a strategy cannot provide any response time guaranties, it should be set $F_{AT} = 0$.

System architecture is given on Fig. 1. *Manager agent* (**M**) is the cornerstone that fulfils all coordination activities and ensures the satisfied quality of filtering services. It is the entity that first performs resource estimation in order to be able to select strategies that will be asked to perform filtering. As soon as activated filtering agents have produced results, **M** will adapt the knowledge that it possesses about them based on response time measurements. In the case of receiving any feedback from the *User agent* (**U**) about result relevance, **M** will perform further adaptation.

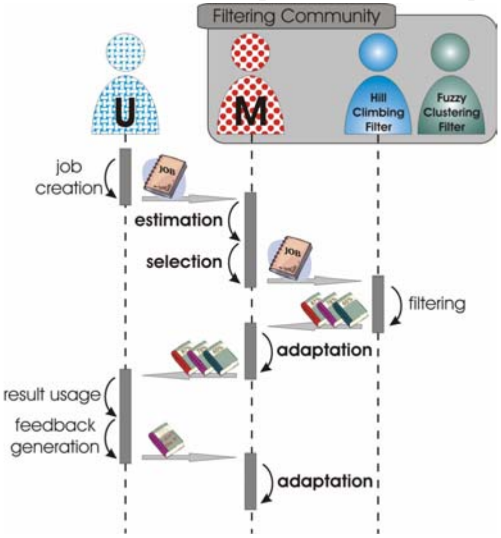

Fig. 1. System architecture illustrating agent communication

Estimation

After computing ω_{CPU}, ω_{DB} and ω_M values as in [1], so-called *main resource fitness* ($F_{mr}^{(i)}$), corresponding to filtering strategy i, can be computed as follows:

$$F_{mr}^{(i)} = \frac{\omega_{CPU} F_{CPU}^{(i)} + \omega_{DB} F_{DB}^{(i)} + \omega_M F_M^{(i)}}{\omega_{CPU} + \omega_{DB} + \omega_M}$$

The difference between main resource fitness $F_{mr}^{(i)}$ and similarly defined total fitness $F_t^{(i)}$ from [1] lies in the exclusion of user quality requirements from $F_{mr}^{(i)}$. This is possible because the filtering framework with the response time aware coordination mechanisms simply assumes that user will assign larger time slot whenever better results are needed and whenever longer response time can be tolerated.

Selection

A selection simulates the evolutionary process of a competition among available strategies, which are fighting for getting as many jobs as possible. The selection mechanism should establish not only a fair fight among them, but also should ensure that at least one any time strategy will get a filtering job. That any time strategy should serve as a guarantee that results will be ready on time, being the essential property of this response time aware coordination. The fair fight is realised through the application of proportional selection [2], which specifies that the selection probability $P_y^{(i)}$ for strategy i is proportional to its fitness value, i.e.

$$P_y^{(i)} = F_y^{(i)}\left(\sum_{j=1}^{n} F_y^{(j)}\right)^{-1}$$

As a fitness value $F_y^{(i)}$, the already introduced $F_{mr}^{(i)}$ is used in the main selection step. In the case where selected strategy j has $F_{AT}^{(j)} = 1$, selection is finished because this any time strategy can give necessary response time guaranties. Otherwise, when it holds $F_{AT}^{(j)} = 0$, it is needed additionally to find one any time strategy that can hopefully work well together with the already selected strategy j. This is achieved by defining *alternative resource fitness* value $F_{ar}^{(i,j)}$ for all strategies i ($i \neq j$) relative to j, as:

$$F_{ar}^{(i,j)} = F_{AT}^{(i)} r_p^{(i,j)} F_{mr}^{(i)}$$

After $F_{ar}^{(i,j)}$ is computed, it is used as $F_y^{(i)}$ in the alternative selection step, which will always select one any time strategy. All not any time strategies will have $F_{ar}^{(i,j)} = 0$, and therefore they will be without chances to be selected.

Not each and every two strategies can successfully work together. Maybe they both depend on the same resource, which cannot effectively support both of them at the same time. The effect of being able to successfully collaborate is modelled through *strategy pair reliability* $r_p^{(i,j)}$, being included in $F_{ar}^{(i,j)}$ computation. Value $r_p^{(i,j)}$ shows how particular strategy pair (i,j) was successful in the past in delivering results that users like. It is assumed that when two strategies collaborate well, the available resources will be successfully exploited, and hopefully good results will be found. Such two strategies should have large $r_p^{(i,j)}$ value, which will facilitate their selection in the future. The low pair successfulness in the past will result in a small $r_p^{(i,j)}$, which will reduce $F_{ar}^{(i,j)}$ and accordingly will diminish chances that the same pair will again collaborate.

Adaptation

While the adaptation of $F_{CPU}^{(i)}$, $F_{DB}^{(i)}$ and $F_M^{(i)}$ values has been discussed in [1], $r_p^{(i,j)}$ value is adapted based on the received actual feedback q_a as:

$${}^{(new)}r_p^{(i,j)} = {}^{(old)}r_p^{(i,j)}\left(\frac{q_a}{\varepsilon}\right)^{l(t)}$$

Value q_a corresponds to the quality of a result that was found when strategies i and j worked together, ε is a tolerance that defines how good feedback should be in order to get a reward, and $l(t) = l_0 e^{-\beta t}$ is a learning rate.

Experimental Results

As a test environment, PIA system is used because it actively helps to 26 DAI Labor workers in their information retrieval activities. Starting from 27th of September 2004, the following 4 coordination schemes were tested. The first 3 days PIA was working without resource aware coordination (PIA I), the next 3 days pure resource aware coordination was plugged in PIA (PIA II), and the last two, 3 day long, PIA configurations are based on a response time aware coordination, where one assigns 100s (PIA III) and the other 20s (PIA IV) as the time slot. It is noticed that not taking care about resources while doing coordination will produce long-lasting filtering jobs [1]. A resource aware coordination is managing to eliminate these long-lasting jobs, but without giving the strict response time guaranties.

The obtained user feedback values are given in Table 1. While the feedback values are comparable for PIA I, PIA II and PIA III systems, a significant decrease is noticed when a response time aware coordination with 20s time slot is used (PIA IV). Setting a realistically big time slot, such as 100s, will not reduce a feedback value (PIA III), which proves that the response time aware coordination can be used without sacrificing a user satisfaction.

Received feedback	PIA I	PIA II	PIA III	PIA IV
Very good	7	13	11	2
Good	19	15	16	12
Bad	4	6	4	15
Very bad	4	3	5	9

Table 1. Received feedback values for different PIA systems

Users are not ready to pay unreasonably high price for the reduction of a filtering time, but they are willing to wait little longer in order to get better recommendations. The advantage of the response time aware coordination mechanisms is contained in the fact that users can known in advance how long at most they have to wait for the results.

Conclusion

The presented solution is trying not only to eliminate long-lasting filtering jobs, but also to give guaranties regarding the response time. Even though the realised coordination mechanisms are generic, which treat filtering jobs as black boxes, future work will be concentrated on taking into account also the properties of jobs while doing estimation.

References

[1] Albayrak, S; Milosevic, D. 2004. Self Improving Coordination in Multi Agent Filtering Framework. *IEEE/WIC/ACM Conference on IAT and WI*, China.

[2] Michalewicz, Z.; Fogel, D. 2000. *How to Solve It: Modern Heuristics*. Springer-Verlag New York, Inc., NY.