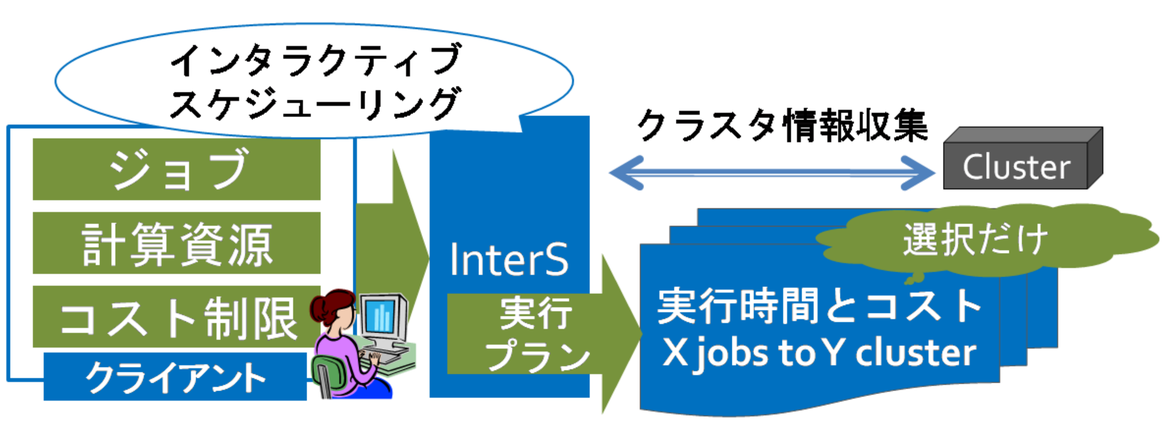
自然言語処理アプリケーションのローカル計算機とパブリッククラウド資源上でのハイブリッド実行手法

孫 コウ[1]　合田 憲人[1,2]
1. 東京工業大学　2. 国立情報学研究所

はじめに

近年，Googleサーチエンジンをはじめ自然言語の処理，検索技術が注目を集めている．自然言語処理（Natural Language Processing）アプリケーションは通常同種類な処理が数多くあるため，ローカル計算機クラスターを利用し並列処理することで実行時間の短縮を図ることができる．しかし，ローカル資源が共同利用されているため，**混雑時利用者のジョブがキューイングされいつ実行されるかは推測しがたい．さらに，計算機故障などの原因により利用できる資源が突然足りなくなる場合もある**．このような状況を改善する技術として，組織外部にあるパブリッククラウド上の資源を簡単に確保ができるクラウド技術が注目されている．本研究は，インターネット掲示板にある不適切な発言（誹謗中傷、個人情報など）を自動検出通知する自然言語処理（NLP）アプリケーションを研究対象とする．**上記の資源不足の問題を解決するためにローカル計算機とパブリッククラウドを組み合わせた実行方法を提案し，既存のInterSスケジューラー（参考文献[1]）をパブリッククラウドの資源を扱えるように拡張する．**

従来研究InterSの概要

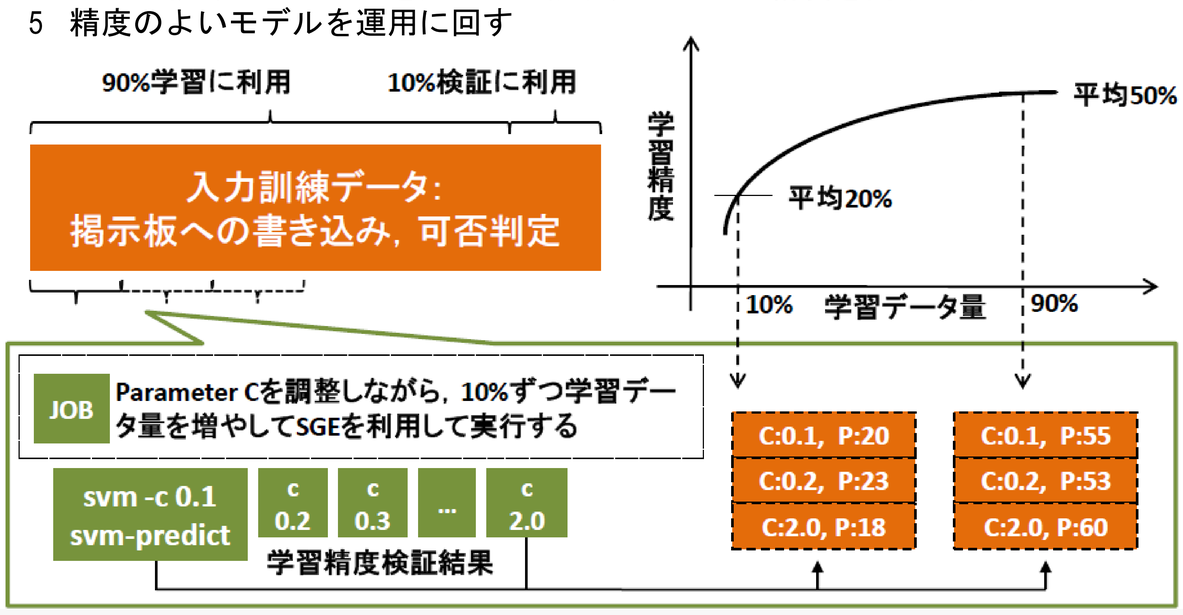


InterSはユーザのために実行プランを作成する．**ユーザはグリッドに詳しくなくても利用できる**．InterSはアプリケーションの実行性能と安定性を保つために実行失敗や計算機ノードの性能・状態変化を自動検知し解決策となる代替案をユーザに提示できる．締め切りを守れるようにコスト制限を超えるプランも提供できるため，**全自動スケジューリング方式と比べて想定外の問題をよく対処できる**．

研究対象アプリケーションの概要

パラメータサーベィ型アプリケーション

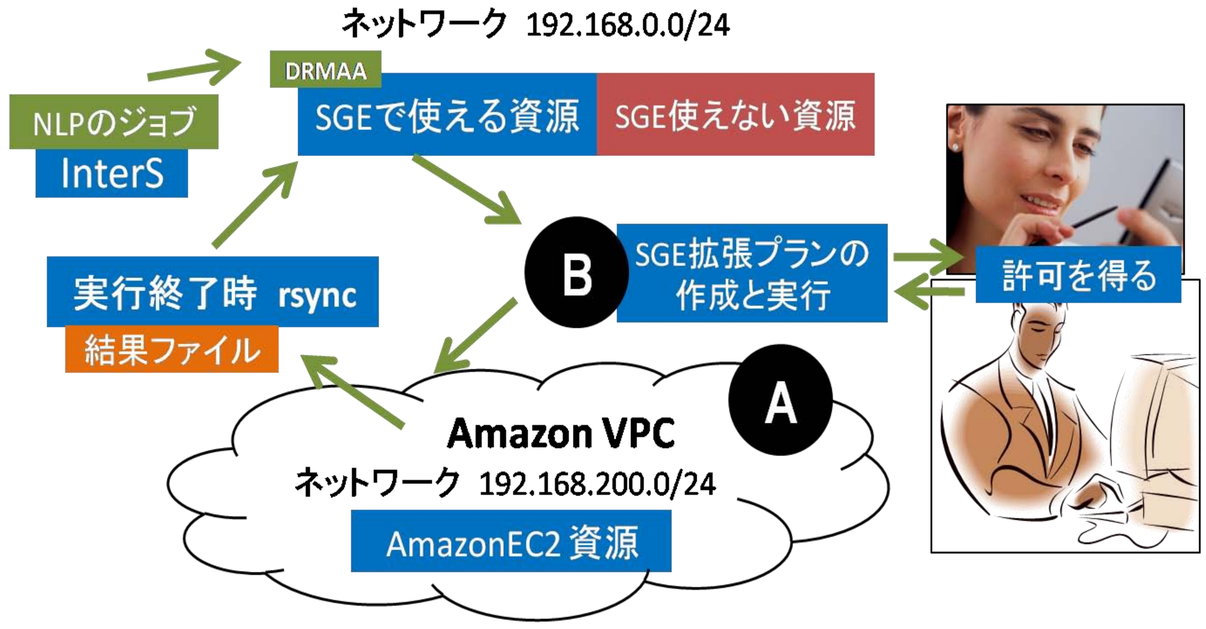
1　学習飽和度合いの検証に学習データを10%刻みにして増やしつつ学習する
2　学習の精度を影響する要素に関してパラメータサーベィ学習を行う
3　学習結果として判定モデルが出力される
4　残りの教師つきデータを用いて上記判定モデルの精度を割り出す
5　精度のよいモデルを運用に回す



要求A　セキュアな実行環境が必要
掲示板のデータには個人情報が含まれているためセキュアな環境で上記アプリケーションを実行する必要がある．

要求B　アプリケーションへの変更がないこと
NLPアプリケーションがすでにSGEに対応しているため，アプリケーションへの変更を避けるように動的にSGEの資源を拡張することで設計する必要がある．

今回の提案と実装



提案A.　VPN(AmazonVPC) を利用しセキュアな環境を構築する
Amazon EC2を利用しセキュアな環境を構築するにはIPSec-VPN技術をベースとするAmazon Virtual Private Cloud（Amazon VPC） を利用することが推奨されている．本研究ではオープンソースのIPSec接続ソフトウェアracoonとBGP対応したルータquaggaを利用しVPC環境を構築した．

提案B.　SGEを用いた動的な資源拡張
B1　計算ノードのAMIイメージ準備
Amazon VPCを通してSGEの実行キューを動的に拡張できる条件として計算ノードのsge_execdとsge_qmasterが通信できることである．今回はクラスターゲートウェイがsge_qmasterであり，計算ノードを準備するだけでよい．そのためsge_execdをインストール済みのイメージを準備しInterSからインスタンス化するだけでSGEの計算ノードとして利用できるように実装した．
B2　計算ノードの追加
InterSを用いて1分間隔でキューイング中ジョブを取得し，その数に合わせて必要なEC2インスタンスの数を計算しSGEの拡張を行う．例えば，コスト制限内ならば20個ジョブが溜っている場合は2コアのAmazonインスタンスを20jobs/2cores = 10個追加することになる．コスト制限超過した場合はユーザから許可をもらう．
B3　計算ノードの削除
AmazonEC2は1時間単位で課金している．実行ジョブがSGEのスケジューリングにより分散されているため，InterSは5分間隔で計算ノードを監視し，実行ジョブがないものを課金される5分前に削除要求する．
B4　SGEジョブの実行
対象のNLPアプリケーションのSGEジョブ(Bashファイル)を分析し，内部でInterSのジョブとして作成する．標準化されているDRMAA APIを利用しSGEへジョブ投入しジョブの実行状態を管理する．

おわりに
本研究では対象となるNLPアプリケーションをセキュアな環境で性能よく実行できるような仕組みを考案した．今後，性能測定のもとで既存研究[2][3]との比較を行いたい．

参考文献

[1] Hao Sun and Kento Aida ,“Interactive Application Scheduling with GridRPC, ”IPSJ Transactions on Advanced Computing Systems, Vol3, No1, pp.88-100, Mar.2010.
[2] Condor Workers on Amazon EC2: http://www-rcf.usc.edu/~juve/condor-ec2/
[3] SDM Cloud Service Adapter の概要: http://wikis.sun.com/pages/viewpage.action?pageId=184750309

TEMPLATE DESIGN © 2008
www.PosterPresentations.com

Market-based Resource Allocation for Distributed Computing

Ikki FUJIWARA[†], Kento AIDA[†‡] and Isao ONO[§]

[†] The Graduate University for Advanced Studies (SOKENDAI)
[‡] National Institute of Informatics [§] Tokyo Institute of Technology

Abstract

Market-based resource allocation is expected to be an effective mechanism to allocate resources in a cloud computing environment, where the resources are virtualized and delivered to users as services. In this paper we propose a market mechanism to efficiently allocate multiple computation/storage services among multiple participants. The proposed mechanism enables the users to (1) order a combination of arbitrary services with a co-allocation or a workflow manner, and (2) receive future/current services at the forward/spot market.

The Market Model

Assumptions

- ✓ The amount of a service can be measured in throughput (e.g. *MIPS* for a computation service or *GB/h* for a data processing service)
- ✓ A provider's resource can be divided into arbitrary fraction
- ✓ A task can be divided into sub-tasks and executed on multiple resources
- ✓ A task can be suspended, resumed and/or migrated during the runtime

Forward Market

- ✓ Deals with advance reservations per pre-defined timeslots
- ✓ Makes contracts periodically (clearinghouse auctions)

Bidding Language

- ✓ Allows users to combine arbitrary resources (allocated all or nothing)

for Users

- Valuation
  - Resource Type
  - Quantity
  - Earliest Timeslot
  - Latest Timeslot
  - Total Length

for Providers

- Valuation
  - Resource Type
  - Quantity
  - Earliest Timeslot
  - Latest Timeslot

Allocation Scheme

- ✓ Maximizes total welfare = Σ(buyer's valuation – seller's valuation)
- ✓ Formulated as a mixed integer program

$$\text{maximize} \quad w = \sum_{j=1}^{|N|} v_j u_j - \sum_{i=1}^{|N|}\sum_{j=1}^{|M|}\sum_{k=1}^{|G|}\sum_{t=1}^{T} v_i y_{i,j,k,t} \qquad (1)$$

s.t.

$$\sum_{k=1}^{|G|} x_{j,k} - |G|\, u_j = 0 \qquad 1 \le j \le |N| \qquad (2)$$

$$\sum_{t=1}^{T} z_{j,k,t} - l_{j,k}\, x_{j,k} = 0 \qquad 1 \le j \le |N|, 1 \le k \le |G| \qquad (3)$$

$$\sum_{j=1}^{|N|} y_{i,j,k,t} \le 1 \qquad 1 \le i \le |M|, 1 \le k \le |G|, 1 \le t \le T \qquad (4)$$

$$q_{j,k} z_{j,k,t} - \sum_{i=1}^{|M|} q_{i,k} y_{i,j,k,t} = 0 \qquad 1 \le j \le |N|, 1 \le k \le |G|, 1 \le t \le T \qquad (5)$$

$$(a_{j,k} - t) z_{j,k,t} \le 0 \qquad 1 \le j \le |N|, 1 \le k \le |G|, 1 \le t \le T \qquad (6)$$

$$(t - d_{j,k}) z_{j,k,t} \le 0 \qquad 1 \le j \le |N|, 1 \le k \le |G|, 1 \le t \le T \qquad (7)$$

$$(a_{i,k} - t) \sum_{j=1}^{|N|} y_{i,j,k,t} \le 0 \qquad 1 \le i \le |M|, 1 \le k \le |G|, 1 \le t \le T \qquad (8)$$

$$(t - d_{i,k}) \sum_{j=1}^{|N|} y_{i,j,k,t} \le 0 \qquad 1 \le i \le |M|, 1 \le k \le |G|, 1 \le t \le T \qquad (9)$$

$$u_i \in \{0,1\} \qquad 1 \le j \le |N| \qquad (10)$$

$$x_{j,k} \in \{0,1\} \qquad 1 \le j \le |N|, 1 \le k \le |G| \qquad (11)$$

$$z_{j,k,t} \in \{0,1\} \qquad 1 \le j \le |N|, 1 \le k \le |G|, 1 \le t \le T \qquad (13)$$

$$0 \le y_{i,j,k,t} \le 1 \qquad 1 \le i \le |M|, 1 \le j \le |N|, 1 \le k \le |G|, 1 \le t \le T \qquad (14)$$

where

- $M = \{m_1, \ldots, m_{|M|}\}$, $m_i = \{v_i, S_i\}$: selling orders
- $N = \{n_1, \ldots, n_{|N|}\}$, $n_j = \{v_j, S_j\}$: buying orders
- $G = \{g_1, \ldots, g_{|G|}\}$: services
- $1 \le t \le T$: timeslots
- v_i and v_j : valuation
- u_j : the number of services with $q_{j,k} > 0$
- $q_{j,k}$: quantity of service g_k
- $a_{j,k}$: arrival time, $d_{j,k}$: deadline, $l_{j,k}$: total length — for buying order
- $e_{j,k}$: timeslot — for selling order
- $S_j = \{(g_k, q_{j,k}, a_{j,k}, d_{j,k}, l_{j,k}) \mid 1 \le k \le |G|\}$: a buying order
- $S_i = (g_k, q_{i,k}, e_{i,k})$; $1 \le k \le |G|$: a selling order

Pricing Scheme

- ✓ Calculates the price which the participants actually pays/earns
- ✓ K-Pricing: Price = (buyer's valuation + seller's valuation) / 2

Spot Market

- ✓ Deals with immediate reservation up to the next timeslot begins
- ✓ Makes contracts continuously (continuous auction)
- ✓ Bidding language, allocation scheme and pricing scheme are similar to the forward market except that they have only one timeslot

Simulator

- ✓ Consists of a centralized exchange and autonomous agents

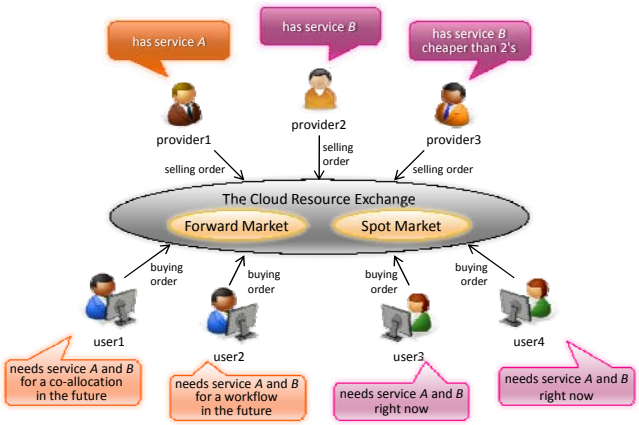
Markets

- ✓ Constructed based on MACE [1]
- ✓ Uses CPLEX or lp_solve as backend solver

Agents

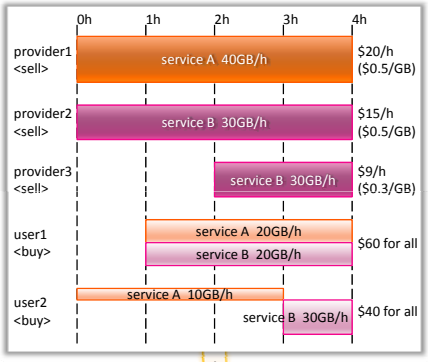
- ✓ Developed to be compatible with U-Mart [2]
- ✓ Can be either software or real human

Overview

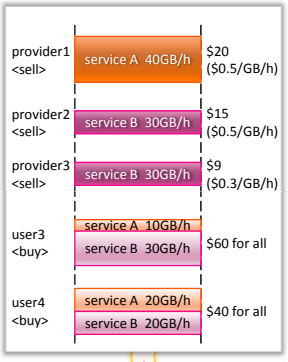


Example

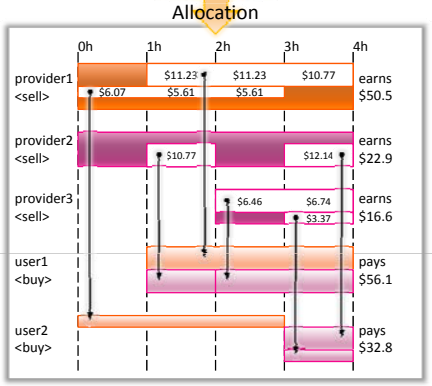
Forward Market



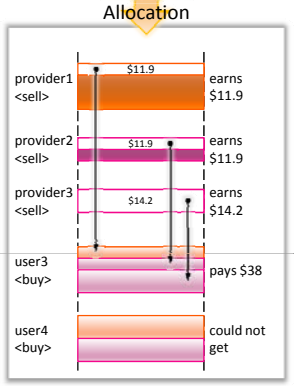
Spot Market



Orders → Forward Market → Allocation



Orders → Spot Market → Allocation



Conclusions & Future Work

- ✓ We proposed a market mechanism to allocate resources in a cloud computing environment
- ✓ Experiment shows that the market mechanism works properly
- ✓ We anticipate that the forward price will serve as a forecast of the spot price; intelligent agents will autonomously avoid the high-priced period which means a tight supply-demand situation
- ✓ We are going to investigate the market behavior and evaluate the performance of the proposed mechanism

References

[1] B. Schnizler, D. Neumann, D. Veit and C. Weinhardt, "Trading grid services – a multi-attribute combinatorial approach," *European Journal of Operational Research*, Volume 187, Issue 3, pp. 943-961, 2008.

[2] H. Sato, Y. Koyama, K. Kurumatani, Y. Shiozawa and H. Deguchi, "U-Mart: A Test Bed for Interdisciplinary Research in Agent Based Artificial Market," in *Evolutionary Controversies in Economics*, pp. 179-190, 2001.