A Reconfigurable Architecture via Grain Perception Operator

Fan Yang
School of Information and Engineering
Huangshan University
Huangshan, Anhui, China

Laixin Shen
School of Information and Engineering
Huangshan University
Huangshan, Anhui, China

Sheng Li
College of Science
Anhui University of Science & Technology
Huainan, Anhui, China

Run Wang
School of Information and Engineering
Huangshan University
Huangshan, Anhui, China

Abstract—**Application tasks are complex and diverse, which need HPC (High Performance Computing) to solve them. The computing efficiency is hugely different when they run in different architectures, because different application tasks have different computing features. The PMC (Processor-Memory-Communication) resource requirements of application tasks are perceived to get PMC operating grain, and the best matched architecture is assigned to the application. Hypergraph is used to describe the application structure and the computing architecture, and its isomorphic principle is utilized to build and demonstrate the super-mixed heterogeneous architecture. Experiments show that reconfigurable architecture based on operator grain perception has characteristic high computing and low energy consumption.**

Keywords—High performance; architecture; perception; PMC grain operator; Hypergraph

I. INTRODUCTION

HPC (High Performance Computing) application problem is becoming more and more complex and diverse, involving many different calculating types, which include atmospheric science, molecular dynamic, material science, fluid mechanic, quantum chemistry, signal processing, bioinformatics, and other disciplines, such as linear algebra, Fourier transform, search sorting [1]. HPC system is mainly divided into two categories: 1) In a MPP (Massive Parallel Processing) architecture based on homogeneous system, the energy consumption is very serious and the actual running rate is low. 2) The advanced CPUs with accelerator processor, such as the Cell, GPU, FPGA, DSP are composed of heterogeneous special systems and their components can accelerate heterogeneous architecture, which greatly raise the application performance and effectively reduce power consumption. The heterogeneous architecture becoming a HPC is one of the important developing trends. Different dedicated processors have significant differences in term of performance and application field, such as image and floating point operations in GPU, whose efficiency is 50~100 times than CPU, and the combination of general processor and dedicated processor is the key to high performance in heterogeneous system [2].

Different calculating tasks are with different characteristics that need to compute their PMC by resource requirements varying. There are some resource model including the common intensive computing, stream intensive, data intensive computing, I/O intensive tasks, internal mixed parallelism. The same application running in different structures has operational performance differences. The time complexity of matrix multiplication on a serial processor is $O(n^3)$, and is $O(3n)$ in a parallel structure Mesh, and is $O(n)$ in the Torus structure. The single architecture is unable to meet many complex application tasks. Reconfigurable computing has been applied in many HPC, and its architecture is used to design a variable structure to adapt different application tasks [3]. Reconfigurable computing makes HPC feasible and further improves the computing power, low power consumption, low cost, and low development cycle, which provides an effective way to solve the technical wall problems.

In this paper, according to some basic algorithm research of commonly used, such as linear algebraic, combinatorial optimization, graph algorithms, and image processing, different computing characteristics need different complicated algorithm and running structure. At the same time, according to the calculation characteristics of various types of processing components, the analysis and calculation of application tasks in the algorithm of particle size of each sub-algorithm are important, and the PMC of each sub-algorithm is particle calculated and the current state of existing components are also computed by the perceptron algorithm. The sub-structure is designed reasonably, and the sub-algorithm is assigned to the sub-structure in order to suit the calculation characteristic to get a computing model for different system structures according to the application of tasks. So the sub-algorithm can realize the highest computing performance, and achieve sub-parts on a high utilization rate.

II. RELATED WORKS

Reconfigurable computing and HPC are hot research at home and abroad. [2] pointed out that whether the integrated circuits develop, HPC or the Internet and memory, about in 2020, there will be encountering insurmountable information technology wall. The classification model of computer architecture was given in [3], which was proposed for heterogeneous system structure in chip level, node level and system level according to the interconnection model of architecture based on different hardware granularity. [4] proposed an flow model based on the service execution and

system designing object. [5] presented a practical parallel computation models named the LogGP synchronization model, which is based on non-exclusive heterogeneous and reflected the impact of heterogeneity and non-exclusive computing environment for concurrent algorithm design and analysis from the system level. A reconfigurable architecture [6] was analyzed and gave the design method for general purpose and special purpose in energy consumption level multi-processor in special applications with 500 times performance improvement, and 70% energy saving. [7] pointed out that a reconfigurable architecture can make the hardware resources behavior to adapt to the special computing requirements on hardware resources level, which provides an interactive mechanism to maximize the use of logic resources.

[8] proposed the system level granularity architecture, which put forward reconfigurable computing applications with hybrid interconnected manner in static and dynamic science. [9] worked in a large-scale multiprocessor network, and used optical interconnection equipment to complete the heterogeneous architecture design. [10] proposed that using multi-FPGAs system structure, the fine-grained and coarse-grained partition problem behavior, whose memory space was effectively used. [11] putted forward the calculation of the HPC necessity. The above thesis was presented respectively in the level of hardware, operating system, scheduling algorithm, such as the interconnection of the reconfigurable architecture design, but only single architecture is used to solve different application tasks with different complexities and diversities. There is lack of viewing the application task angle, where system structure may be suitable for its computing features and the calculation of the performance and energy consumption should be taken into account. [12] proposed the viewpoint of "Application deicide its structure, structure decide its effectiveness". According to different applications, revealing the application characteristics of the reality problems, the suitable different variable system structure model can be designed to make the structure suitable for the application. Thus, the optimal target including high computing performance and energy consumption can obtain.

[13] determined the minimum number of hyper edges in a hypergraph and characterized the hyper edges of a k-partition-connected hypergraph. [14] used hypergraph to assemble all local link structures, and employed HMETIS for hypergraph partitioning. [15] proposed a novel algorithm called hypergraph regularized non-negative matrix factorization which captured intrinsic geometrical structure by constructing a hypergraph instead of a simple graph. [16] presented a low-rank matrix factorization method, which incorporated multiple hypergraph manifold regularization. The hypergraph is introduced to model the local structure of the intrinsic manifold. [17] modeled the haplotype assembly problem using hypergraph partitioning formulations and proposed a novel hypergraph-based haplotype assembly method. [18] introduced the class of cored hypergraph and power hypergraph, and investigated the properties of their Laplacian Eigenvalues.

III. PMC GRAIN ANALYSIS

For many applications in the field of HPC, PMC parts can be analyzed to suit for some basic algorithm. In the same way, the PMC grains of components or sub-structures are also calculated. Perception algorithm is used to calculate grain characteristics of components and application. The matching component or sub-structure is assigned to different applications and reach the targets of high performance and low energy consumption.

A. The basic concept of grain calculating

Grain calculating is applied to perceive the characteristics of tasks and obtain a pattern. Components and sub-structures are also need to calculate their grain features, including numerical algorithms, combinatorial optimization algorithm, fast Fourier transform, image processing. The matrix multiplication can be regarded as a basic grain.

Grain calculating has many characteristics, such as independence, diversity, universality, and intensive features including computation intensive, data intensive, communication intensive, storage intensive. Granularity characteristics involve instruction level fine-grained, process function level granularity, program process coarse granularity, operation service level granularity. Structure characteristics include a branch, loop, order, and pattern feature, which includes cell task pools, stream/serial, and task/data parallel. Some algorithm grains are shown in Table 1.

TABLE I.   GRAIN CALCULATING DESCRIPTION OF COMMONLY USED ALGORITHM (PART)

Commonly used algorithm	grain calculating
Support vector machine	dense matrix
SuperLU/OSKI/SpMV	Sparse matrix
FFT/DFT/SFT/IFFT	Spectral matrix
Hash/CRC/RSA	Combinational logic
Bayesian Markov/HMM	Graph model
BP/ANN	Dynamic program
Monte Carlo/similar	Graphs
Fast multi-stage	N-Body
Video compression/group	Finite automata

B. PMC grain calculating model of algorithms

P is used to express calculating grain (calculation, including computing intensive), M express memory grain (storage, including I/O intensive and data-intensive), C express communicating grain (communication, including communication intensive, service intensive). The implementation algorithm of application tasks (application algorithm) is performed by multiple sub-algorithms together.

We define the PMC grain calculating model as a six group: $OG = (O, G, P, M, C, U)$. $O = \{o_1, o_2.. o_n\}$ express calculating grain; $G= \{e_1, e_2...e_n\}$ is an edge set, which express the relation of data dependencies between these calculating grain. $P = \{p_1, p_2,... , p_n\}$ express the processor grain; $M = \{m_1, m_2,...m_n\}$ express the grain of the memory capacity required; $C=\{c_1, c_2,...c_k\}$ express each edge of the communication; $U = \{u_1, u_2,... ,u_n\}$ express the type of grain; $T_i \in \{CPU, GPU, FPGA, Cell.. \}$.

Next, the grain perception algorithm is given as follows:

a) For a basic computing components (FPGA, GPU, DSP, Cell, etc.): Its calculation feature is analysized and tested to master the characteristics of computing components and the best computational performance for specific applications.

b) The basic application of high performance computing analysis: The commonly used basic algorithms such as linear algebra, matrix operations, image processing, graph algorithm are studied to get their computing resource requirements, and all kinds of basic PMC calculate grain.

c) The PMC is the relationship between grain: such as the relations between serial, parallel, calculating grain class relationship, and combination relationship, which consider calculating grain of the combination of PMC constraints.

d) For a given application: the relationship between perception and PMC grain calculating are assigned to match components or sub-structure. The types of components use the interconnection between parts and established the relationship with reference to the relationship between sub-algorithms.

e) In order to effectively represent sub-algorithm relationship: the relationship between components and the corresponding matching states are described in this paper by using hypergraph application and architecture algorithm.

IV. HYPERGRAPH DESCRIPTION OF THE APPLICATION ALGORITHM AND ARCHITECTURE

Collaborative application tasks are considered as many sub-algorithms, architecture, and its interconnection structure composed of multiple parts. We give hypergraph and system structure definition.

Definition1: Application Hypergraph (*AH*) is defined as a 4-tuples group. $AH = (A, E, W, D)$. $A = \{a_1, a_2, ... a_n\}$ is a node set. $E = \{e_1, e_2, .., e_m\}$ is an edge set. $W = \{w_1, w_2, .. w_n\}$ is a node attribute set. $D = \{d_1, d_2, ..., d_m\}$ is an edge attribute set.

AH can calculate grain by the PMC model using $O-G$ equivalent transformation.

The transformation rule: the node set is same, that is, $A = O$. The edge set is the same according $E = G$. The node properties are corresponding to the combination, $W = M \cup P$, named $w_i = (p_i, m_i)$, $i \in \{1,2,...,n\}$. The properties are corresponding to the same edge, $D = C$, named $d_i = c_i$, $i \in \{1,2,...,k\}$.

Definition2: Architecture hypergraph (GH): a tuple $GH = (G, P, M, N)$, where $G = \{g_1, g_2...g_n\}$ is a part set. P is an edge set, named the interconnection between components. $M = \{m_1, m_2... m_n\}$ is an attribute set for processing parts, $N = \{d_1, d_2...d_k\}$ is a set of communication part properties. Processor part properties set represents the amount of calculation and the combination of traffic attribute. The communication part properties generally describe the relationships and the size of the traffic.

Definition3: Hypergraph Isomorphism (IS): For the 2 given hypergraph, $AH= (A, E, W, D)$ and $GH= (G, P, M, N)$. If there is a 1-1 mapping function f, which satisfies (1):

$$\forall a(a \in A \to f(a) \in G) \cap$$
$$\forall a \forall b(a \in A \wedge b \in A \wedge (a,b) \in E \to \quad (1)$$
$$(f(a), f(b)) \in P)$$

The attributes are equal to corresponding nodes and edges (or similar), i.e.:

$$\forall a(a \in A \to W(a) \cong M(f(a)))$$
$$\forall e(e \in E \to D(e) \cong N(f(e))) \quad (2)$$

For a given PMC model, we can get the application algorithm hypergraph through the combination of attributes transform. Then we use the algorithm hypergraph and system structural hypergraph to reach isomorphism.

In this paper, we use a practical application to illustrate the realization process of matching algorithm with its architecture based on the granularity computing of the application task.

In a cloud computing task, there is an application, sub-algorithm $a_1 \sim a_8$ work cooperatively. Different granularities compute each seed algorithm. A call exists in each other or the communication relationship between them. The relationship between the sub algorithms is used hypergraph to describe, as shown in Fig. 1.
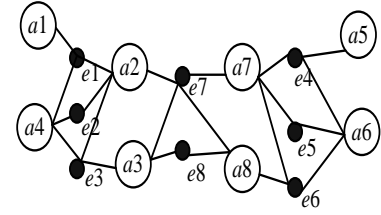

Fig. 1 Application of the hypergraph algorithm

In Fig. 1, the white circles represent some sub-algorithms. The black points express edges, which are the relationship of these sub-algorithms. Each application algorithm realized by a plurality of sub-algorithm cooperation can be made the appropriate division and combination according to the correlation pair algorithm. So we can use several sub-cluster algorithms to form a layered hypergraph.

The calculation of size perception is related to the sub-algorithm. The sub-algorithm is assigned to the processing unit for the calculation of particle size on the execution. The cluster algorithm is applied in system structure of neutron structure. Based on Hypergraph isomorphism, the properties are layered architecture hypergraph, which are shown in Fig. 2.
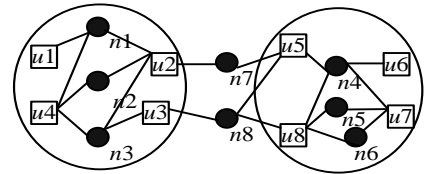

Fig. 2 The architecture hypergraph

In Fig. 2, $u_1 \sim u_8$ can be a processing component or sub-structure, which are used to calculate the corresponding $a_1 \sim a_8$ sub-algorithms. The processing parts are GPU, Cell, FPGA. The sub-structure is parallel structure of Torus and Mesh. A

particular sub algorithm can be assigned to the processing unit for the calculation of particle size or sub-structure executive. A sub-structure is corresponding to a sub-algorithm cluster. The relationship of sub-algorithm cluster within each sub-structure can refer to the corresponding sub-structure within the complete interconnection. The interconnection between sub-algorithm clusters is used to complete the interconnection, which produce a layered architecture hypergraph. At this point, the system structure can be a plurality of sub-structures, which can exist in distributed heterogeneous environment of different geographical position.

V. A RECONFIGURABLE ARCHITECTURE BASED ON PERCEPTION OF GRAIN

Through the application calculating, Internet applications and clouds on HPC application can be composed of a variety of algorithms to achieve. At the same time, each algorithm may include several independent functional sub-algorithms.

In order to efficiently implement the application tasks, the choice of algorithm and sub algorithm must consider the application features including task function size, type of service, the resources status, processing unit connected topology and transmission bandwidth. In order to effectively manage multiple sub algorithms, a sub algorithm cluster of sub algorithms are often formed by combined strong correlation.

For a sub-algorithm, based on the perception algorithm, its computational granularity is got. At the same time, the state of related processing part is perceived. The decision algorithm makes the sub-structure suitable for the calculation of particle size. For a single processor component, we can also assign a parallel structure of Mesh and Torus. In a plurality of multi mapping algorithm and the sub-structure, the algorithm is looking for the highest efficiency of mapping relations to obtain the system structure that best fits the application algorithm. The mapping structure is shown in Fig. 3.
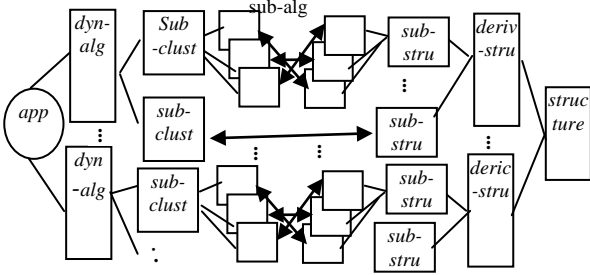

Fig. 3 The application and architecture mapping structure

In Fig. 3, *app* is an application task. *dyn-alg* express a dynamic algorithm. *sub-clust* express a sub-algorithm cluster. *sub-stru* is a sub-structure which runs the sub-algorithm. *deriv-stru* is the derived structure. Architecture mapping transformation model and application algorithm are given below, which are suitable for the calculation of particle size.

Definition4: Calculating grain perception transformation model TSM: a 7-tuples *TSM= (AH, GH, CM, OG, LD, CE, DS)* is described as follows:

a) *AH=(A,E,W,D)* is an application of algorithm hypergraph. $A=\{a_1, a_2,..A_n\}$ is an algorithm set. E is the edge set, W and D are corresponding to the attribute collection.

b) *GH=(G,P,M,N)* is a system of structural hypergraph. $G=\{g_1, g_2,..g_n\}$ is sub-structure set, and its initial edge set is empty. P,M and N are corresponding to the attribute collection.

c) *CM=(CPU,GPU,Cell,FPGA..)* is a processing unit set, including part number attribute, which is composed of elements of sub structure G.

d) *OG=\{o_1,o_2,..,o_n\}* is a grain set. o_i is the PMC granularity attribute of sub-algorithm i.

e) *LD=\{d_1, d_2,..d_n\}* is the load information of neutron structure. d_i is set to {0,1}, and 0 is free, 1 is busy.

f) *CE* is the perceptron algorithm, which express sub-algorithm of PMC size o, or load information of component d.

g) *DS* is the decision algorithm. According to the PMC size o and part load d, the sub-algorithm is assigned to the sub-structure appropriate g treatment.

Transform model execution process is listed as follows:

a) *Initialization:* The appropriate clustering algorithm is based on the sub-scale application of algorithms, parts information to get the sub-algorithm clusters hypergraph.

b) *Perception phases:* The intelligent sensing algorithm is used to calculate the particle size characteristics and component characteristics and load.

c) *The decision-making stage:* The sub-algorithm is P size, M size and component characteristics. The handling algorithm matches its size mode and assign to a sub-structure.

d) *Interconnected stages:* The interconnection between components and sub-structure approach is based on PMC middle C granularity attributes and sub-cluster relationships. The sub-structure is composed of interconnected cluster.

For a practical application task, its dynamic algorithm decomposition (several sub-algorithms) and cluster (sub-cluster algorithm) give the number of sub-clusters algorithm using a cluster-aware algorithm and sub-sub-algorithm PMC grain analysis. The child of PMC algorithm is assigned to match the sub-sub-structure algorithm, and the number of sub-structures is according to the principle of super-isomorphism between sub-structures. The relationship between sub-algorithms is according to the sub-cluster algorithm, which obtain the interconnection between the sub-structures derived from the structure. In the same manner, the derived structure of the interconnect architecture of the algorithm is obtained the relationship between the sub-clusters, and finally we can get a hybrid architecture.

For a given application, if the application uses different algorithms, and the different particle of sub-algorithms, the respective assigned sub-structure should be different in different architectures formed, and the operation efficiency will be different. Saturated optimum computing without constraints and with different constraints build multiple hybrid models to guide hybrid reconfigurable architecture to complete the application of high-performance implementation.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

To illustrate the efficiency of heterogeneous reconfigurable architecture model, simulation experiments are designed.

Example1: A given application algorithm comprises the number of the sub-algorithm clusters ranging from 10 to 200, respectively. Each sub-cluster contains three kinds of arithmetic operation, namely KNN (K Nearest Neighbor) algorithm, keyword matching (Cmatch) and Bayesian algorithms. The available components include a general purpose CPU, GPU and other special-purpose processor

Table 2 shows the comparison of the complexity and executing time of the three algorithms on a single processor, multi-processor and GPU. n is the input data dimension. m is the training sample data. p is the number of processors..

TABLE II. COMPARISON OF THREE KINDS OF TIME COMPLEXITY OF THE ALGORITHM

algor	*1 个CPU*	*p 个CPU*	*GPU*
KNN	$O(mn)$	$O(mn/p+mnO(logp))$	15
Cmatch	$O(n)$	$O(n/p)$	30
Bayes	$O(mn+n)$	$O((mn/p)+nlog(p))$	18

Example2: A travel service in the cloud has three services (Web Service), mapping service, travel services and weather service. Each Service contains several operations respectively. $op_1 \sim op_{10}$ is the relationship between the solid line for their callings. The dotted line is a dependency. Travel services application architecture is shown in Fig. 4.
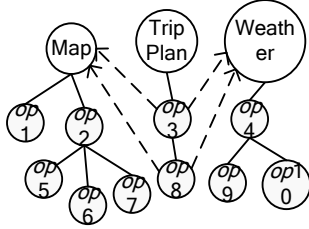

Fig. 4 Travel Services Application

When the number of sub-algorithms is designed for 1, 3,4,10 cluster structures in the serial, parallel Mesh, the super hybrid structure TSM (TSM1 is assigned a Mesh substructure, TSM4 is assigned 4 Mesh substructures and TSM10 is assigned 10 Mesh substructures) is comparable shown in Fig. 5.
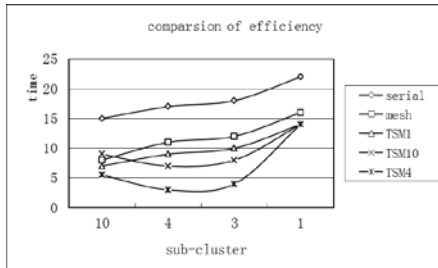

Fig. 5 Comparison of execution time in different structure

In Fig. 5, TSM1 Mesh structure is comparable with the execution time of TSM10 execution time when the number of clusters is 10. When the number of clusters is 1, the execution time of TSM1, TSM4 and TSM10 are considerably as a sub-algorithm cluster assigned to only one sub-structure. When the number of clusters is 4, we can allocate 4 sub-structures to deal it. At this time, the amount of computation and communication integrated is optimal and execution time is the shortest.

When the algorithm is decomposed into sub-cluster algorithm, the calculation needs to consider the traffic amount of sub-algorithms. The particle size of sub-divisions is too large or too small, the overall computing performance is affect ed. The optimal clustering scheme assigns the appropriate sub-structure to perform the corresponding sub-algorithm. At this time, the local performance achieves optimal, and each sub-structure has basically the same workload. The relationships between sub-algorithms and sub-structures are matched with the appropriate implementation structure of communication components to build reconfigurable system, which enables these applications achieve the overall performance of algorithms. Meanwhile, the work of the various components of the reconfigurable architecture is load balancing, high utilization and low power consumption.

VII. CONCLUSION

Based on the characteristics of the application computing tasks, variable architecture can be designed to match them. The basic algorithm uses grain operators and grain PMC analysis. The application-aware algorithm operators obtain task characteristics and features of computing components and sub-computing structures, adaptable algorithms are assigned to match the characteristics of the sub-structures, which can establish a reconfigurable heterogeneous architecture.

In the future we'll delve into the basic algorithm analysis and the extraction of grain pattern, and improve PMC automatic identification and classification of grain to improve the perception and intelligence of decision-making algorithm.

REFERENCES

[1] G.L. Chen. Parallel Computing Structure algorithm programming Peking: Higher Education Press, 2003.

[2] G.J. Li. The Long term trend of information science and technology and our strategic alignment.Chinese Science: Information Science, 2010, 40 (1): 128-138.

[3] X.B. Shen, F.C. Zhang. The classification model of computer architecture. Chinese Journal of Computers, 2005, 28(11): 1759-1766.

[4] L. Joe, J. Qi, Y.C. Gong. A support and implementation of operating system design and reconstruction of hybrid system. Chinese Journal of computers, 2009, 32(5): 1046-1054.

[5] Y.H. Ji, W.Q. Ding, G.L. Chen. A parallel computing model. Journal of computer utility, 2001, 24(4): 437-441.

[6] Todman T.J, Constantine's G.A, Wilton S.J.E. Reconfigurable Computing: Architectures and Design Methods. IEEE Proceedings Computers and Digital Techniques, 2005, 152(2): 193-207.

[7] Bondalapati K, Prasanna V.K. Reconfigurable Computing: Architectures, Models and Algorithms. Current Science, Special Section on Computational Science, 2000, 78(7): 828-837.

[8] Shoaib K, Ali P, Daniel G, et al. Reconfigurable Hybrid Interconnection for Static and Dynamic Scientific Applications. Proc.of ACM International Conference on Computing Frontiers. ACM Press, 2007.

[9] Artundo I, Heirman W, Debaes C, et al. Design of a Reconfigurable Optical Interconnect for Large-scale Multiprocessor Networks [C]//Proc. of SPIC'08. Strasbourg, France: [s. n.], 2008.

[10] Srinivasam V, Govindarajan S, Vemuri R. Fine-grained and coarse-grained behavioral partitioning with effective utilization of memory and design space exploration for multi-FPGA architectures. IEEE Transactions on Very Large Scale Integration System, 2001, 9(1): 140-158.

[11] Z.N. Chen. From the high performance computing to the high performance computing. Computer Education, 2004, (6): 26-28.

[12] Wu Kong Hing. Cloud computing efficient road to [EB/OL]. 2010.11.

[13] X.F. Gu, H.J. Lai. Augmenting and preserving partition connectivity of a hyper graph. Journal of Combinatorics, 2014, 5(3): 271-289

[14] H.C. Tao, Z. Wu, J. Shi, et al. Overlapping community extraction: a link hypergraph partitioning based method. 2014 IEEE International Conference on Services Computing, 2014(25): 123-130.

[15] K. Zeng, J. Yu, C.H Li, et al. Image clustering by hypergraph regularized non-negative matrix factorization. Neurocomputing, 128 (2014): 209-217.

[16] T.S. Jin, J. Yu, J. You, et al. Low-rank matrix factorization with multiple hypergraph regularize. Pattern recognition, 48(2015): 1011-1022

[17] X. Chen, Q.K. Peng, L.B. Han, et al. An effective haplotype assembly algorithm based on hypergraph partitioning. Journal of theoretical biology, 358(2014): 85-92

[18] S.L. Hu, L.Q. Qi, J.Y. Shao. Cored hypergraphs, power hypergraphs and their Laplacian Hedgenvalues. Linear Algebra and its Applications, 439(2013): 2980-2998

Authors

Yang fan received the M.S. degree from Hefei University of Technology, Hefei, China, in 2007.

She is currently an instructor with the school of information and engineering, HuangShan University, Anhui, China.

Her research interests include data mining, massive data process, and architecture. Her email is fanyang80@163.com.

Shen Lai-xin, corresponding author, received the M.S. degree from Jinan University, Guangzhou, China, in 2005. He is currently working towards the Ph.D. degree at the Department of Computer Science and Technology, Tongji University, Shanghai, China.

He is also an associate professor in HuangShan University, Anhui, China. His research interests include machine learning, society computing and big data. His email is slx965@163.com.

Sheng Li, is currently an instructor with the college of science, Anhui University of Science & Technology.