The Process Reassignment Design and Implementation for Business Continuity Operations

Pakorn Sermsuk
The Information Processing Inst. for Education and Development
Thammasat University
Bangkok, Thailand
Email: pramoon@tu.ac.th

Kasiphan Masakul
Faculty of Informatics
Sripatum University
Bangkok
Thailand
Email: masakul@spu.ac.th

Suchai Thanawastien
Faculty of Informatics
Sripatum University
Bangkok
Thailand
Email: stw@spu.ac.th

Abstract

In this paper, a new design methodology for business continuity is presented. The methodology is based on the process reassignment in a multi-processor system in which disjoint process groups are running on different processors. The reassignment process is modeled as the process coupling graph and matrix in which the coupling weight between different process groups is defined as the number of data accesses as defined in the CRUD matrix. The scenario considered is that of multiple single failures of processors that can occur over a time period. The reassignment vectors for each state of failures are computed on-the-fly.

Key Words: Business Continuity, Survivability, Process Reassignment, CRUD Matrix

1. Introduction

In the era of decreasing hardware cost with increasingly powerful processors, the failure modes become a complex combination of software errors and hardware glitches. The cost of having a total system failure is unacceptable for any enterprises. Moreover, the pressure of a business continuity paradigm is imperative in most of modern corporations. Hence, the design of a highly survivable system is now one of the underlined goals of many organizations.

Survivability is the ability of a system to recover predefined service levels in a timely manner after the occurrence of system failure or simply the failure of processors. The system failure can be the result of complex interaction of hardware and software or due to the consequences of a disaster. A desirable system should have some degree of fault tolerance or some mechanism to degrade the system gracefully while trying to return to normal service within an acceptable time even when a disaster has occurred.

Jacques Botha and Rossouw Von Solms [1] presented a cyclic model for business continuity planning for small organizations to be able to recover after a disaster. The implementation and approach proposed is incremental in nature with specified goal for each iteration. Benjamin B.M. Shao [2] presented the optimal redundancy allocation for information technology disaster recovery.

A solution procedure based on probabilistic dynamic programming is presented along with two examples. Lucia Cloth and Boudewijn R. Haverkort [3] proposed a model based on continuous scholastic logic (CSL) to assess survivability. The model was illustrated by evaluating the survivability of the Google file system using stochastic Petri nets. Most of the literature in this area either provides a conceptual framework for business continuity [4-7] or quantitative analysis after the occurrence of disasters [8-11]. In general, the problem of designing system with business continuity mechanisms built in has not been addressed. In this paper, we will examine the business process design with a built-in business continuity mechanism based on process reassignment. The concept of migrating or reassigning processes from a failed processor to one of the good processors in the system is well-accepted as a means to increase availability; however no research has been reported on how to incorporate this concept in the design phase for the purpose of increasing the chance of business continuity in spite of critical processor failure.

2. The Process Reassignment Problem

In designing the highly available distributed information system, M groups of highly coupled processes occur on M processors, after assigning a group of processors to a process. Then during the execution of the processes, there is communication between processes with other processes residing on other processors. If there is a processor failure or total communication link failure of a processor, the strategy to survive is to reassign the process group of the failed processor to one of the existing good processors. The question is which processor should be the target processor. The criteria might be the processor with the lowest load, or the nearest processor, or the processor that would result in some minimum cost criteria.

In this section, we will define the coupling cost of the potential remote data access between processes residing in different processors. The ideal case is to design the process groups such that there are no reads/writes between groups. However, in designing an enterprise-wide integrated application, this is not possible as certain processes and data will be used by many process groups. In the extreme case, we can put the entire process group on one big processor which is highly risky unless proper redundancy technique is deployed. The single processor case, provides none of graceful degradation necessary in operating an enterprise information systems. If each group of processes is assigned to a processor and there are no couplings between any pair of the processes in the processors, then a processor failure would affect only the processes on that failed processor, while other processors can still operate correctly. Consequently the processes in the failed processor can be assigned to any one of the good processors with some degradation in performance. For the practical case where processes interact across processors, it is desirable to have minimum coupling between processors.

3. The Process Reassignment Model

Let the CRUD matrix with a business processes *b* data classes be represented by the matrix $M(b_i,d_j)$ when b_i is the business process i, i = 1 to *a*, and d_j is the data class j, j = 1 to *b*. Let us consider 2 macro processes labeled F and G shown in Figure 1.

Next, we identify two sub matrices F and G as F $(b_e{:}b_f\ ,\ d_g{:}d_h)$, and sub matrix $G(b_g{:}b_h\ ,\ d_p{:}d_q)$. From these two matrices, when projecting the coordinates along the vertical and horizontal axes, the result would be two matrices $M(b_e{:}b_f\ ,\ d_p{:}d_q)$ and $M(b_g{:}b_h\ ,\ d_m{:}d_n)$ shown as shaded area in Figure 1.

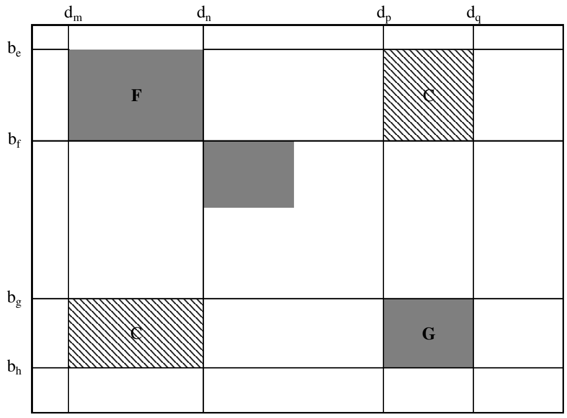

Figure 1 Macro processes in CRUD matrix

Next, we will define the Process Coupling Graph (PCG). The process coupling graph is a directed graph with the nodes representing the macro process and the edges is defined as follows. For macro processes F and G (node F and G of the PCG), if the sub matrix $M(b_e{:}b_g\ ,\ d_p{:}d_q)$ contain at least one entry with "U" then there is an edge from node G to F; if the entries of the matrix $M(b_g{:}b_h\ ,\ d_m{:}d_n)$. The degree of how tightly two macro processes are coupled can be characterized by the number of U's in the matrices R= $M(b_e{:}b_f\ ,\ d_p{:}d_q)$ and S = $M(b_g{:}b_h\ ,\ d_m{:}d_n)$. For simplicity, let each entry with "U" be denoted by 1 unit. Then, we define the coupling weight, as the number of U's in the coupling matrix. Hence

$$cw_{RS} = \sum_{y=p}^{q}\sum_{x=e}^{f}[M(b_x,b_y)],\ cw_{RS} \geq 0$$

$$cw_{SR} = \sum_{y=p}^{q}\sum_{x=m}^{n}[M(b_x,b_y)],\ cw_{SR} \geq 0$$

The total coupling weight (tcw) between two macro processes are defined as $cw_{RS}+cw_{SR}$.

Process Coupling Matrix (PCM) is the matrix representation of PCG. The entries of the matrix are the coupling weights. Let tcw(x, y, N) be the total coupling weight of the PCG with N nodes, hence

$$tcw_{(N)} = \sum_{y=1}^{N}\sum_{x=1}^{N}[PCM(x,y)]$$

x, y = 0, if the initial configuration is the index of the node (failed node), and y is the index of the target node (up node).

3.1 The Reassignment Matrix

Let R(x,y) be the reassignment matrix in which node x is merged with node y, the entry R(x,y) is tcw(x,y,N-k), and k is the number of failed nodes.

Let R() be the reassignment matrix constructed from the Coupling graph. For the entry q(i,j) , the value is computed from the sum of all the coupling weights minus the coupling weights of the entry m(i,j) and m(j,i). For the processor i, the optimum reassignment is identified as the entry of the row i with minimum cost value

$$J = \min Q(i, j), \quad j = 1 \ldots m$$

3.2 Process Reassignment

In the PCG if a node i is merged with node j, then the reduced PCG with N-1 nodes, since there are N processors, will have a total of N(N-1) weight. Assuming that process x is merged with node y (meaning that process x has failed), then the tcw (x,y,N-1) of the reduced PCG is

$$tcw(x,y,N-1) = tcw(N) - cw_{xy} - cw_{yx}$$

3.3The Optimum Reassignment Trajectory (ORT)

The optimum reassignment trajectory (ORT) if node x fails is the target node that would result in minimum total coupling weight.

$$ORT(x) = \min [\, [R(x,y) \, , \, y = 1 \text{ to } N-1] \,]$$

Hence, from the R(x,y), we can construct a reassignment vector (RV) that is a list of the target nodes.

$$RV = (\, ORT(1), ORT(2), \ldots, ORT(N) \,)$$

For each ORT(x), there will be a corresponding PCG. Also, the set of all RV's is the Process Reassignment Map.

4. Example

4.1 Develop the CRUD matrix

By performing a system analysis to identify all the business processes and data classes in an enterprise, the CRUD matrix is formed with the processor on one axis and the data group on another axis. Each entry of the matrix then would be either "U" if the business process uses some data in that data class, "C" if the business process creates some data in that data class, or " " if the business process does not use any data in that data class. Figure 2 shows an example of the CRUD matrix.

Data Classes / Process	Planning	Policy/Regulation/Law/Legal	Man Power	Mission and Function	Personnel	Products	Financial Status	Procurement and Contracts	Work in Progress	Training	Security and Safety	Reference Information	Agency Agreements	Fixed Assets and Expenses	ADP Information
Implement Policy Guidance from OCE	C	U	U	C		U		U	U		U	U	C	U	
Manage Research and Development Programs	U	U	U	U	U	C	U	U	C	U	U	U	U	U	U
Manage Reimbursable and Small Problem Program	U	U	U	U	U	C	U	U	C	U	U	U	U	U	U
Provide Strategic Direction	U	U		C					U			U	C		
Develop Improved O&M Procedures		U				C	U		C						
manage Financial Resources	U	U	U				C	U	U				U	U	
Manage Procurement and Contract	U		U				U	C	U			U		U	U
Manage Manpower	U	U	C	U	C		U	U	U	C	U		U	U	U
Manage Organizational Effectiveness Programs	U	U	C		U	U	U		U	C				U	
Manage Public Affairs Program	U	U	U	U	U	U			U			C	U	U	U
Manage Personnel Training	U	U	U	U	U		U	U		C	U	U			U
Develop Improved Planning, Design & Construction Procedures		U				C	U		C						
Manage Laboratory Support Functions	U	U	U	U		U	U		U		U	C	U	C	U
Provide Staff Review and Approval	U	U	U	U	C	U	U	U	U	U	U	U	U	U	
Investigate and Solve Personnel Problems		U			C	U	U		U	U	U				
Manage Safety and Security Activities		U			U					U	C				U
Formulative Army RDT&E Programs	C	C	U	U	U	U	U		U	U	U	U	U	U	U
Manage Automation and Information	U	U				U	U	U	U	U	U			U	C

Figure 2 Example of the CRUD matrix (Adapted from IBM's Business System Planning)

4.2 Perform the Macro Process Analysis

From the CRUD matrix, we perform the affinity analysis to cluster the "C" entries so as to align along the northwest to southeast diagonal; the grouping is done forming adjoining sets with mutually exclusive groups of business processes and data classes as shown in Figure 3. Here, we have 6 Business areas or 6 macro processes denoted by A, B, C, D, E, and F.

Process \ Data Classes	Policy/Regulation/Law/Legal	Mission and Function	Agency Agreements	Planning	Work in Progress	Products	Financial Status	Procurement and Contracts	Man Power	Training	Personnel	Security and Safety	Reference Information	Fixed Assets and Expenses	ADP Information
Implement Policy Guidance from OCE	U	C	C	C	U	U		U	U			U	U	U	
Provide Strategic Direction	U	C	C	U	U								U		
Formulative Army RDT&E Programs	C	U	U	C	U	U	U		U	U	U	U	U	U	U
Manage Research and Development Programs	U	U	U	U	C	C	U	U	U	U	U	U	U	U	U
Manage Reimbursable and Small Problem Program	U	U	U	U	C	C	U	U	U	U	U	U	U	U	U
Develop Improved Planning, Design & Construction Procedures	U				C	C	U								
Develop Improved O&M Procedures	U				C	C	U								
manage Financial Resources	U		U	U	U		C	U	U					U	
Manage Procurement and Contract				U	U		U	C	U				U	U	U
Manage Manpower	U	U	U	U	U		U	U	C	C	C	U		U	U
Manage Organizational Effectiveness Programs	U			U	U	U	U		C	C	U			U	
Manage Personnel Training	U	U		U			U	U	U	C	U	U	U		U
Provide Staff Review and Approval	U	U	U	U	U	U	U	U	U	U	C	U	U	U	
Investigate and Solve Personnel Problems	U				U	U	U			U	C	U			
Manage Safety and Security Activities	U									U	U	C			U
Manage Public Affairs Program	U	U	U	U	U	U			U		U		C	U	U
Manage Laboratory Support Functions	U	U	U	U	U	U	U		U			U	C	C	U
Manage Automation and Information	U			U	U	U	U	U		U		U		U	C

Figure 3 Six macro processes after the affinity analysis.

4.3 Assign the Macro Processes to Processors

Each Macro process will be assigned to a processor. Hence in our example, we have six processors, each having a macro process A, B,…, F respectively.

	A	B	C	D	E	F
A		10	4	14	9	2
B	5		2	7	4	2
C	2	6		8	1	2
D	4	6	2		5	1
E	7	6	3	9		2
F	1	2	1	2	3	

Figure 4 The process coupling matrix (PCM)

4.4 Construct the Macro Process Flow Diagram with Weighted Edges

From the CRUD matrix, a Process Coupling Graph can be constructed. The nodes are the macro process and the edges are the coupling between. There is an edge with coupling weight $cw_{RS} + cw_{SR}$. Shown in Figure 4 is the Process Coupling Matrix of the Process Groups shown in Figure 3. The corresponding Process Coupling Graph is shown in Figure 5.

4.5 Find Process Reassignment Trajectory

Here, we compute the total coupling of the coupling graph assuming a processor failure where the processes are assigned to one of the processors resulting in a new coupling graph with one less node and a new total coupling weight. From the example in Figure 4, the reassignment matrix is shown in Figure 6. The entry with highlighted numbers is the optimum reassignment. It is apparent that the process reassignment trajectory for multiple single failures is basically a tree with N levels, with the node at level i would having N-i+1 child.

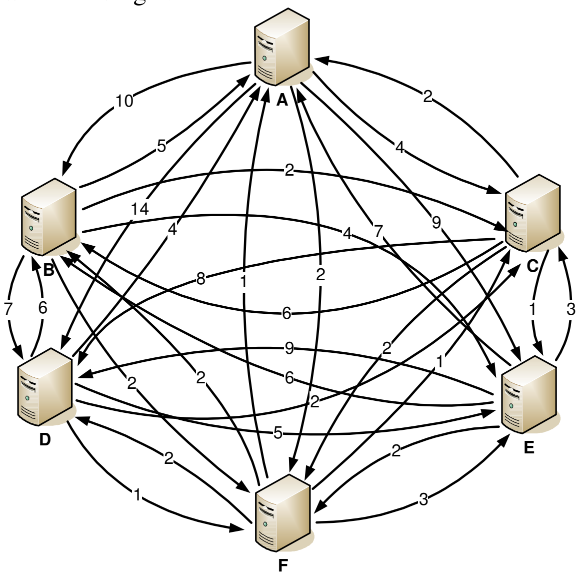

Figure 5 The coupling graph of the process groups

4.6 Design the Process Reassignment Map (PRM)

For the reassignment to work as desired, it is thus necessary to have a way to generate the reassignment vector efficiently. For example, for Figure 6, the reassignment vector is [D, A, B, A, AE] for any single processor failure. From this configuration, each processor can then assume the next single processor failure event, and generate the reassignment matrix and the reassignment vector.

	A	B	C	D	E	F
A		117	126	114	116	130
B	117		120	119	122	128
C	126	120		122	127	129
D	114	119	122		118	129
E	116	122	127	118		127
F	130	128	129	129	127	

Figure 6 The reassignment matrix (assuming single processor failure

4.7 Implement the PRM

The process reassignment map can be generated at the design phase after the identification of macro processes and the process coupling matrix as shown in Figure 7. Alternatively, the PRM is generated stage by stage as dictated by the number of processor failures up to the current state. Some of the global information is needed to initialize and synchronize the reassignment of processes in order to provide continued operations in spite of multiple single processor failures. This can be implemented using current process communication techniques.

5. Conclusion

A new process design technique for designing a highly available and gracefully degrading system is presented. The techniques is based on generating the process coupling graph from a CRUD matrix that partitions the processes and data classes of the entire enterprise into a set of disjoint communicating macro processes. In contrast, past techniques do not provide a systematic approach to designing enterprise systems with high availability in mind.

Future work will be in the area of modeling the performance of such a system using the CTMC (Continuous Time Markov Chain) model. Also, a detailed design of the underlining mechanism to provide either a centralized or distributed process reassignment mechanism will lead to practical use of such a design methodology.

6. References

[1] J. Botha, and R.V. Solms, “A cyclic approach to business continuity planning”, Information Management & Computer Security, Emerald Group Publishing Limited, Vol.12, No.4, 2004, pp. 328-337.

[2] Benjamin B.M. Shao, “Optimal Redundancy Allocation for Information Technology Disaster Recovery in the Network Economy”, IEEE Transactions on Dependable and Secure Computing, Vol.2, No.3, July-September 2005, pp. 262-267.

[3] L. Cloth, and B.R. Haverkort, “Model Checking for Survivability”, Proceedings of the Second International Conference on the Quantitative Evaluation of Systems, 2005.

[4] Dudenhoeffer, D. Donald, and M. Manic, “CIMS: A framework for infrastructure interdependency modeling and analysis”, Proceedings of the 2006 Winter Simulation Conference.

[5] R. W.Helms, S. van Oorschot, J. Herweijer, and M. Plas, “An integral IT continuity framework for undisrupted business operations”, Proceedings of the First International Conference on Availability, Reliability and Security.

[6] Keeton, Kimberly, and Merchant, “A framework for evaluating storage system dependability”, Proceedings of the 2004 International Conference on Dependable Systems and Networks.

[7] C. Castillo, “Disaster preparedness and Business Continuity Planning at Boeing : An integral model”, Journal of Facilities Management, Vol. 3, Number 1, 2004, pp.-8-26

[8] Y. Liu and K. S. Trivedi, “A general framework for network survivability quantification”, 12th GI/ITG Conference on Measuring, Modeling and Evaluation of Computer and Communication Systems together with 3rd Polish-German Teletraffic Symposium (MMB & PGTS 2004). VDE Verlag, 2004, pp. 369–378.

[9] B. Ellison, D. A. Fisher, R. C. Linger, H. F. Lipson, T. Longstaff, and N. R. Mead, "Survivable network systems: An emerging discipline," Carnegie Mellon Software Engineering Institute, Tech. Rep. CMU/SEI-97-TR-013, 1997.

[10] D. Pradhan, Fault-tolerant and dependable computer system design, 2nd ed. Prentice Hall, 2003.

[11] S. C. Liew and K. W. Lu, "A framework for characterizing disaster-based network survivability," IEEE J. Selected Areas in Communications, Vol. 12, No. 1, pp. 52–58, 1994.

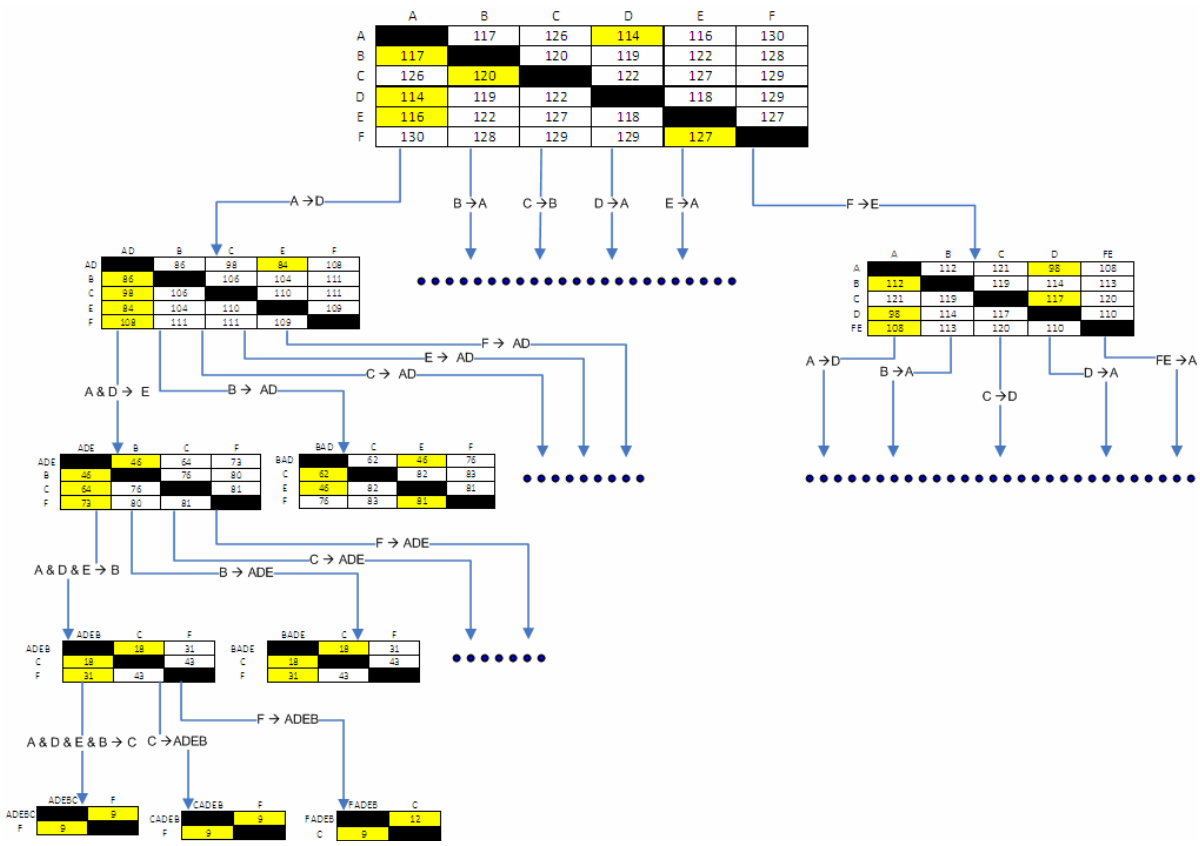

Figure 7 The process reassignment map