社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－OS－108 (14)
2008／4／24

相関障害への耐性の高い広域分散データ配置の検討

阿部洋丈[†1] 梅村恭司[†1]

広域分散システムにおいて，多数のノードが単一の原因によって同時に障害に見舞われるケースがある．そのようなケースは相関障害 (correlated failure) と呼ばれている．本論文では，大きなデータを分割して複数のノードに分散配置する場合に，それぞれの配置が相関障害に対してどのような耐性を持っているかを評価する方法について論じる．各ノードが保持する複製数が高々2 の場合は，複製配置を置換群の元として解釈することで，その元の持つ互いに素な巡回置換の分割によって耐性を評価することができる．この結果より，分散システムによく見られるランダムな配置やシーケンシャルな配置よりも耐性の高い配置があり得ることを示す．

On wide-area distributed data placements with higher tolerance to correlated failures

Hirotake Abe[†1] and Kyouji Umemura[†1]

There are cases where a single cause brings simultaneous failures on many nodes in a wide-area distributed system. Such cases are called correlated failures. In this paper, we describe our ongoing work for developing a method that can evaluate impact that a data placement used in a distributed storage system against tolerance to correlated failures. In cases that the number of data fragments that each node can hold is at most 2, a data placement can be interpreted as a member of a permutation group. We developed a method to calculate the tolerance of a data placement based on the structure of coprime circular permutations in the equivalent permutation of the placement. Using the method, we show that there can be better placements than random placement or sequential placement, which are commonly used in existing wide-area distributed systems.

1. はじめに

近年，Peer-to-Peer システムなどの広域分散システムにおける相関障害 (Correlated Failure) への対策が注目を集めつつある．相関障害とは，単一の原因によって分散システム中の多数の多数のノードが同時に障害を起こす現象である．Nath らの観測によれば，PlanetLab[6] 等の環境において，最大で約 20 %のノードが同時に通信不可能になる場合があったことが報告されている[5]．

ここでは，複数のノードにまたがってデータを分割保存しておくような広域分散システムを考える．データを保存するというタスクは，それ自体が目的である場合[2)–4)] もあれば，何らかのサービスを実現するために必要な機能である場合もある．いずれにせよ，分散システムにとって代表的なタスクの一つである．もし，そのようなシステム上で相関障害が起きると，システムが障害から十分に回復するまでは一部のデータにしかアクセスできないという事態が生じる恐れがある．

本論文では，データを分割保存する広域分散システム上において，相関障害への耐性を高めるためにはどのような配置方法が適しているかを検討する．本論文では，分散システムを互いに対等なノードの集合として抽象化し，その上で実現可能なすべてのデータ配置を，相関障害への耐性の観点から等価なグループに集約するというアプローチをとる．

各ノードが持つ複製数が高々2 である場合，分散システム上で実現し得るデータ配置のグループは，ノード数を n とすると，n の整数分割と 1 対 1 に対応づけることができる．本論文では，ある整数分割に代表されるデータ配置が，すべてのデータを揃えるために最低限必要なノードの数を求める方法を示す．その結果，これまでの広域分散システムでよく用いられているランダムな配置やシーケンシャルな配置は，相関障害への耐性という観点からは必ずしも優れていないことが示唆される．

†1 豊橋技術科学大学 情報工学系
Department of Informantion and Computer Sciences, Toyohashi University of Technology

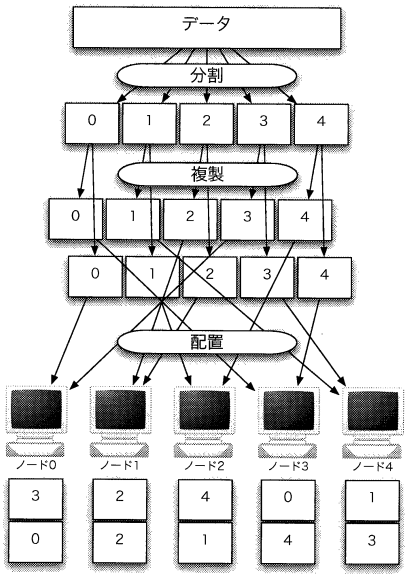

図 1　想定するシステムのモデル.

2. 想定するシステム

ここでは，本論文が想定する広域分散システムを規定する.

広域分散システム S は，与えられたデータを保持し，要求に応じてそのデータを返すストレージシステムである．システム S は n 個のノードで構成され，その数は途中で増減するものは無いものとする．但し，障害の発生によって一時的にノードが離脱する場合がある．障害については後述する.

データの保持

与えられるデータは，単一のノードだけではすべてを保持することができないくらい大きいものとする．そのため，データは分割され，複数のノードにまたがって分散配置される必要がある．このとき，単にデータを分割しただけでは，データの可用性が大きく損なわれてしまう．データの冗長性を高めるために，データは n 個（ノード数と同数）のデータブロックに等分され，それぞれ m 個の複製が作成される．各ノードはデータブロックを m 個保持する能力を持ち，システム固有のデータ配置に従って，特定のデータブロックの複製を保持する．このノードを模式的に表しているのが図 **1** である.

障害モデル

システム S に属するノードは，障害の影響で，他のノードの一部，もしくは全部と通信ができなくなることがある．このとき，障害は以下の二つの条件を満たすものとする.

対称性　ノード A から他のノード B へメッセージが送信可能な場合は，B から A へのメッセージ送信も可能である．また，A から B へメッセージが送信不可能な場合は，B から A へのメッセージ送信も不可能である.

推移性　A と B が互いに通信可能で，かつ，もう一つのノード C と B が通信可能な場合は，A と C も通信可能である．但し，通信可能とは，双方向にメッセージが送受信可能であることとする.

このモデルのもとでは，システム障害は，ノード集合の分断の形で現れる．このモデルは，ノードの故障による障害，および，ネットワーク故障による障害の両方をカバーする.

また，本論文では，各ノード間の障害に対する相関関係は未知であるとして議論を進める．つまり，各ノードは何らかの相関関係があってそれが相関障害を引き起こすが，そのとき，あるノードがどの部分集合に含まれるかは独立事象として扱う．もし，事前に相関関係を見積もることが可能である場合は，より適切なデータ配置を決定できる可能性がある．その方法は今後の検討課題の一つである.

相関障害への耐性

システムの耐故障性について議論する場合，ある特定の障害について耐故障性があるか否かという議論がしばしば行われる[*1]．本論文では，そのような耐故障性とは異なる意味で，「相関障害への耐性」を以下のように定義する.

定義 1. 相関障害によってノード集合が分断され，ある部分集合の大きさが n_p　$(0 \leq n_p \leq n)$ であった場合，その部分集合に含まれるノードが保持しているデータブロックだけで元々のデータ全体が復元できる確率 $f_S(n_p)$ を，システム S の持つ「相関障害への耐性」と呼ぶ.

明らかに，$f_S(0) = 0$，$f_S(n) = 1$ である．また，f_S は単調非減少な関数である．n_p が増加するにつれ，$f_S(n_p)$ はある点で $f_S(n_p) = 0$ から $f_S(n_p) > 0$ に転じ，そこから徐々に増加を始める．本論文では，この転換点をシステムの特徴を表す重要な性質であると捉え，以下のように定義する.

定義 2. あるシステム S において，$f_S(n_p - 1) = 0$ かつ $f_S(n_p) > 0$ であるような n_p を $n_{\min}$ と呼ぶ.

⋆1 例えば，RAID5 はディスク 1 台の故障には耐故障性があるが，ディスク 2 台の同時故障に対しては耐故障性はない，など.

3. 検　　討

3.1 ノードが持てる複製数が 2 の場合

ここでは，一つのノードが持つことができるデータブロックの数（すなわち m）が 2 である場合について検討する．数が 2 に固定されていることで，あるデータ配置が持つ本質的な構造を n 次の置換群の元に帰着させることができるために，議論が容易になる．

複製数が固定されているために，ここで議論する $m = 2$ のシステムは常に有用であるとは言えない．それでも $m = 2$ の場合を検討する意義は次の二つである．まず第一に，ノード数 n が比較的小さい（たとえば 10 前後）システムの場合には複製数が 2 でも有効性が見出せるためである．第二に，$m \geq 3$ の場合でも同様の取扱いを行うことができる可能性が予見されるためである．

3.1.1 データ配置の置換としての表現

ここでは，各データブロックの等価性に着目することで，あるシステムにおけるデータ配置が，n 個の記号に対する置換群の元として表現できることを示す．

各データブロックには，一般に，それぞれを識別するための通し番号が付けられる．しかし，相関障害への耐性の観点からは，これらの通し番号自体には識別子として以上の意味は無く，通し番号になっている必要は無い．なぜなら，各データブロックは，与えられたデータを単純に n 等分したものであるためである．もしも，ある程度のオーバーラップを持たせて分割するなどして，データブロック間に何らかの相関関係がある場合は，その順序を無視できない．しかし，今回はそのような相関はないので，「データがすべて揃う」という目的に対して各データブロックが持つ貢献力は等しい．

そこで，各ノードが持つデータブロックの識別子を，扱いが簡単になるように付け変えることを考える．具体的には，各ノードが持つデータブロックの一つと，そのノードの識別番号自体が一致するような一対一の対応を考え，それに従ってデータブロックの識別番号を付け変える．

例を使って説明する．**表 1** は，図 1 に示されたデータ配置を表にまとめ直したものである．この例では，各ノード上でデータブロックを格納する場所をスロット 1 およびスロット 2 として区別している．これは，システムの実装上は必要のない区別であるが，説明の便宜上このようになっている．各ノードにおいてスロット 1 に着目すると，そこに格納されているデータブロックは各ノード毎にすべて異なっている．そのため，ノード番号とスロット 1 に格納されているデータ番号は一対一の関係になっている．そのため，この一対一関係に従ってデータブロックの番号の付け替えを行うことによって，実際のデータ配置を保ったままで，ノード番号とスロット 1 のデータブロック番号を一致させることができる．同様に，スロット 2 に着目した場合も一対一の対応を作ることができる．

表 1　データ配置の例．$n = 5, m = 2$ の場合．

ノード番号	0	1	2	3	4
スロット 1	3	2	4	0	1
スロット 2	0	2	1	4	3

表 2　スロット 1 に着目して置換した結果．

ノード番号	0	1	2	3	4
スロット 1	0	1	2	3	4
スロット 2	3	1	4	2	0

スロット 1 に着目して番号の付け替えを行った場合，**表 2** に示すようなデータ配置を得る．ノード番号とスロット 1 のデータブロック番号が一致するように番号が付け替えられているため，情報が冗長になっている．そのため，スロット 1 に入っているデータブロック番号がそのノードのノード番号を表すことに決めることで，相関障害への耐性の等価性を失うことなく，この表からノード番号の情報を削除することができる．

ここまで説明した操作の結果，相関障害への耐性を損なう事無く，第 1 行目が整列された $n \times 2$ の行列を得ることができる．その結果，この行列を n 次の置換群 S_n の元の一つ $\sigma \in S_n$ であると解釈することができるようになる．n 次の置換群とは，n の要素からなる集合から自分自身へ全単射のすべてを元とする群である．置換群の元は，ある記号を置換するという操作に相当する．たとえば，表 2 に示した配置を置換群の元として解釈すると，これは $\{0,1,2,3,4\}$ の 5 つの記号からなる記号列に対して，0 を 3 に，1 は 1 のまま，2 を 4 に，$\cdots$ という置換を行うことを表している．本論文では，以降，下記のような表記法を用いて置換操作を表す．

$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 & 0 \end{pmatrix}$$

3.1.2

ここでは，置換群の元が持つ「互いに素な巡回置換の構造」が，データ配置が持つ相関障害への耐性を決定する要因であることを示す．

前節までの説明で，あるデータ配置から置換群の元

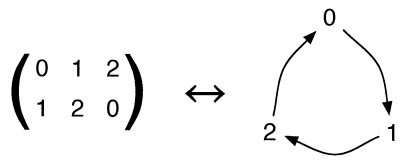

図 2　巡回置換の例

を得る方法を示した．しかし，この方法では，一つのデータ配置から複数の元が得られる可能性がある．前節の説明では，スロット 1 に着目することで置換群の元を得た．もし，スロット 1 ではなくスロット 2 に着目していたら，前節の例とは異なる元を得る．しかし，異なる元に帰着されるとしても，それぞれの操作では相関障害への耐性を保っているはずなので，それぞれの元の中に何らかの共通点が保存されているはずである．

すべての置換は，互いに素な巡回置換に分解することができる．まず，巡回置換の定義，および，それが互いに素であることの定義を示した後に，分解できることを示す．

定義 3. 巡回置換とは， ある m 個の記号集合について，適用するたびに順番に一つずつすべての記号を巡るような置換を言う．たとえば，以下の置換は巡回置換である．

$$\sigma = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \end{pmatrix}$$

この巡回置換では，$\{0,1,2\}$ のうちのどの記号から始めても，すべての記号を巡った後，3 回目の適用で元の記号に戻る (図 **2**)．

定義 4. 巡回置換が互いに素であるということは，2 つ以上の巡回置換について，それぞれの置換が巡回させる記号の集合が互いに共通要素を持たないというとである．たとえば，以下の二つの巡回置換は互いに素である．

$$\sigma_1 = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \end{pmatrix} \quad \sigma_2 = \begin{pmatrix} 3 & 4 \\ 4 & 3 \end{pmatrix}$$

定理 1. 置換群の元は，互いに素な巡回置換に分割できる．

補題 1. 置換 σ が対象とするすべての記号は， σ を有限回適用することで再び元の記号に戻る．(証明省略)

補題 2. 置換 σ が対象とする任意の記号 s について，再び s が出現するまで繰り返し σ を適用することを考える．このとき，その途中で表れた記号の系列は，互いに素な巡回置換を形成している．(証明省略)

証明． 証明は帰納法による．

まず，次数 n が 1 の場合を考える．$n = 1$ であるということは，置換し得る記号はたった 1 種類しかないことを意味する．そのため，その記号で定義できる置換は，唯一の記号をその記号自身に写像するという置換のみである．この置換は，それそのものが巡回置換である．

$1 \leq n \leq k-1$ の全ての置換が互いに素な巡回置換に分割できると仮定して $n = k$ をの場合を考える．いま，次数 k の置換群 S_k から任意の元 σ を取り出し，そこから更に任意の記号 s を一つ取り出したとする．このとき，補題 1 および補題 2 より，この s を含む互いに素な巡回置換を見出すことが可能である．

そこで，σ を，s を含む互いに素な巡回置換と，残りの置換に分割する．もしも何も残らない場合は，σ そのものが単一の大きな巡回置換であるので，分割できたと言える．そうでない場合は，取り出された互いに素な巡回置換の次数は 1 より大きいので，残りの置換の次数は $k-1$ より小さくなり，仮定により分割できる．よって，すべての σ は互いに素な巡回置換に分割できると言える． □

互いに素な巡回置換への分割により，元の置換は，より小さな置換の集合で表すことが可能になった．ここで，互いに素な巡回置換の「構造」を以下のように定義する．

定義 5. ある置換 σ が l 個の互いに素な巡回置換に分解されたとする．そのとき，i 番目の巡回置換の長さを p_i として，$(p_1, p_2, \ldots, p_l)$ を σ の持つ巡回置換の構造と呼ぶ．ただし，$p_i \geq p_{i+1}$ であるとする．また，次数 n の置換に現れ得るすべての巡回置換の構造を要素とする集合を $\mathcal{P}_n$ で表す．

$\mathcal{P}_n$ の要素を列挙するということは，n を和に分解する方法をすべて列挙するということと等しい．この問題は整数分割として広く知られている[1)]．

同じ構造を持つ二つの置換は，記号の付け変えをすることで，互いに一致させることができる．また逆に，異なる構造を持つ二つの置換は，いかに記号の付け変えを行っても，互いに一致させることはできない．そのため我々は，この構造こそが，データ配置を限界まで抽象化した時に残る本質であると考える．

3.1.3 置換の構造から相関障害への耐性を計算する方法

データ配置を置換として解釈し，その置換の持つ巡回置換の構造が求まると，そこからそのデータ配置が持つ相関障害への耐性を計算により求めることができ

る．ここではその計算方法について説明する．まず，計算方法の概略を説明する．次に，$n_p = n_{\min}$ と限定した場合の計算方法を示し，その後で $0 \leq n_p \leq n$ 場合の計算方法を示す．

いま，あるデータ配置の持つ巡回置換の構造が $(p_1, p_2, \ldots, p_l)$ という整数分割で与えられているとする．それはつまり，システムに属するノードが l 個のグループに分割されていることを意味する．この場合，すべてのデータブロックが揃うということは，それぞれの巡回置換毎にすべてのデータブロックが揃うということと同値である．なぜなら，ある巡回置換に含まれるデータブロックは，その巡回置換に属していないノードには全く存在していないためである．そのため，全体ですべてのデータブロックが揃うという事象は，各グループ毎にデータブロックが揃うという事象に分割して考えることができる．以降，p_i の巡回置換に相当するノードのグループのことを，単に i 番目のグループと呼ぶこととする．

$n_p = n_{\min}$ に対する f_S

システム全体において最小限のノード数ですべてのデータブロックが揃うためには，各グループにおいても最小限のノード数でデータブロックで揃っていなければならない．あるグループ p_i について着目すると，そのグループにおいて全てのデータが揃うために必要なノード数は $\lceil \frac{p_i}{2} \rceil$ で表される．ここで，次の関数を定義する．

定義 6. ノード数が x であるグループにおいて，最小限のノード数ちょうどですべてのデータが揃うようなノードの組み合わせの数を $f_{\mathrm{compmin}}(x)$ で表す．

$f_{\mathrm{compmin}}(x)$ のとる値は，x が偶数であるか奇数であるかによって大きく異なる．まず，x が偶数の場合は，x の値によらず常に $f_{\mathrm{compmin}}(x) = 2$ となる．それは，x が偶数の場合，すべてのデータが揃うようなノードの組み合わせは図 **3** の左に示すような 2 通りしか存在しないためである．最小のノード数で効率良くデータブロックを収集するためには，一つおきにノードを取るような選び方をする必要がある．その場合，可能な取り方は，偶数の位置にあるすべてのノードを取るか，奇数の位置にあるすべてのノードを取るか，二つに一つである．

x が奇数の場合を考える．奇数の場合，そのグループが集めるべきデータブロック数は奇数であるのに対し，最小限の数のノードがもたらすデータブロック数は $2 \times \lceil \frac{p_i}{2} \rceil$ で偶数である．そのため，すべてのデータブロックが揃っているということは，どれか一つは必ず重複していることになる．このことに着目すると，

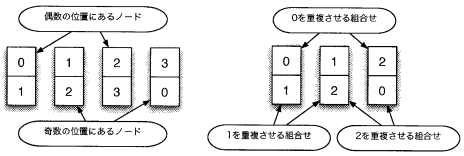

図 **3** すべてのデータブロックを揃えるためのノードの選び方．

x が奇数の場合は，$f_{\mathrm{compmin}}(x) = x$ となることがわかる．なぜなら，そのグループに含まれている x 個のすべてのデータブロックについて，それだけが重複するような組み合わせを考えることができるためである (図 3 の右)．よって，f_{compmin} は以下のように求められる．

$$f_{\mathrm{compmin}}(x) = \begin{cases} 2 & (x \text{ が偶数のとき}) \\ x & (x \text{ が奇数のとき}) \end{cases} \quad (1)$$

各グループにおける組み合わせ数が求まれば，それらを掛け合わせることで，システム全体ですべてのデータブロックが揃う場合の数を計算することができる．よって，システム S が用いるデータ配置が持つ互いに素な巡回置換の構造が $\mathbf{p} = (p_1, p_2, \ldots, p_l)$ で与えられているとき，最小限のノード数 $n_{\min}$ ですべてのデータブロックが揃う確率は下記のように表すことができる．

$$f_S(n_{\min}) = \frac{\prod_{i=1}^{l} f_{\mathrm{compmin}(p_i)}}{{}_n\mathrm{C}_{n_{\min}}} \quad (2)$$

このとき，$n_{\min}$ は以下の式で求められる．

$$n_{\min} = \sum_{i=1}^{l} \left\lceil \frac{p_i}{2} \right\rceil \quad (3)$$

一般 の n_p に対する f_S

次に，$0 \leq x \leq n$ の場合の f_S を計算するために，式 2 を一般の n_p に拡張する．明らかに，$0 \leq n_p < n_{\min}$ の場合は $f_S(n_p) = 0$ であり，$f_S(n) = 1$ である．よって，ここで検討するのは $n_{\min} \leq n_p < 1$ の場合である．まずは $f_{\mathrm{compmin}}(x)$ を一般の x に拡張し，それを用いて $f_S(n_p)$ を求める方法を導く．

まず，f_{compmin} を拡張した関数を次のように定義する．

定義 7. ノード数が x であるグループにおいて，すべてのデータが揃うような $n_{\min} + a$ 台のノードの組み合わせの数を $f_{\mathrm{comp}}(x, a)$ で表す．

次に，$f_{\mathrm{comp}}(x, a)$ を求めるための基礎として，関数 $S(n, t, u)$ を定義する．この関数は，t から n までの間に存在するノードについて，隣接したノードが同時に故障するケースを除外しながら，u 台のノードを

残す組み合わせを一つずつ数え上げる操作を再帰的に表現している．

$$S(x,t,u) = \begin{cases} 1 & (u=0) \\ \sum_{k=0}^{\lfloor \frac{n-t-u}{2} \rfloor} S(x,t+2k+1,u-1) & (\text{それ以外}) \end{cases} \tag{4}$$

S を用いることで，$f_{\mathrm{comp}}(x,a)$ は以下のように求めることができる．

$$f_{\mathrm{comp}}(x,r) = \begin{cases} \sum_{i=1}^{x-2r} S(x,i,2r) & (x \text{ が奇数}) \\ \sum_{i=1}^{x-2r+1} S(x,i,2r-1) & (x \text{ が偶数 かつ } r \neq 0) \\ 2 & (\text{それ以外}) \end{cases} \tag{5}$$

x が偶数のとき $f_{\mathrm{comp}}(x,0) = 2 = f_{\mathrm{compmin}}(x)$ となり，x が奇数のとき $f_{\mathrm{comp}}(x,0) = x = f_{\mathrm{compmin}}(x)$ となることから，$f_{\mathrm{comp}}(x,a)$ は $f_{\mathrm{compmin}}(x)$ の一般化になっていることが確認できる．

更に，$f_S(p_i)$ を計算するために，以下の集合を定義する．

定義 8. $\mathbf{p} = (p_1, p_2, \ldots, p_l) \in \mathcal{P}_n$ のとき，集合 $\mathcal{R}_a^{\mathbf{p}}$ を以下のように定義する．

$$\mathcal{R}_a^{\mathbf{p}} = \left\{ \mathbf{r} \;\middle|\; \sum_{i=1}^{l} r_i = a \text{ かつ } 0 \leq r_i \leq p_i - \left\lceil \frac{p_i}{2} \right\rceil \right\}$$

ただし $\mathbf{r} = (r_1, r_2, \ldots, r_l)$

この集合は，与えられた a に対し，それを各グループに分配するような組み合わせの全パターンを表している．この集合を計算機で列挙する場合は，$\sum_{i=1}^{l} r_i = a$ を満たす場合をすべて列挙し，その中から $0 \leq r_i \leq p_i - \left\lceil \frac{p_i}{2} \right\rceil$ の条件を満たすもののみを抽出することで得られる．$\sum_{i=1}^{l} r_i = a$ の組み合わせを満たす場合の数は ${}_{a+l-1}\mathrm{C}_{l-1}$ になる．

ここまでに説明した関数や集合を用いることで，一般の n_p に対する相関障害への耐性を計算することが可能になる．システム S が用いるデータ配置が持つ互いに素な巡回置換の構造が $\mathbf{p} = (p_1, p_2, \ldots, p_l)$ で与えられているとき，そのシステムが持つ相関障害への耐性 $f_S(n_p)$ は以下の式で与えられる．

$$f_S(n_p) = \frac{\sum_{\mathbf{r} \in \mathcal{R}_{n_p}^{\mathbf{p}}} \prod_{i=1}^{l} f_{\mathrm{comp}}(p_i, r_i)}{{}_n\mathrm{C}_{n_p}} \tag{6}$$

3.1.4 ケーススタディ

ここでは，前節に示した $f_S(n_p)$ を用いて，さまざまな n に対する結果を示し，そこから得られる知見を整理する．まずは，n が小さくて単純なケースから始め，徐々に n が大きいケースへ進む．

ノード数が **5** の場合

$n = 5$ の場合，次数 n の置換群の元に現れうる互いに素な巡回置換の構造は，(5), (4, 1), (3, 2), (3, 1, 1), (2, 2, 1), (2, 1, 1, 1), (1, 1, 1, 1, 1) の 7 通りである．ここでは，次数 1 の巡回置換を含んでいる構造を除外し，(5) と (3,2) の二つを取り上げる．それは，次数 1 の巡回置換を含むということは，あるデータブロックはただ一つのノード上にしか存在していないことを意味し，相関障害への耐性の低い配置であると考えられるためである．

(5) と (3,2) を比較するために，それぞれの $f_S(n_{\min})$ を計算する．なぜなら，唯一異なるのが $f_S(n_{\min})$ であるためである．どちらの場合も $n_{\min} = 3$ であり，かつ，$n_{\min} < n_p$ の時 $f_S(n_p) = 1$ である．以下にそれぞれの $f_S(n_{\min})$ の計算結果を示す．

$$(5):\ f_S(n_{\min}) = \frac{f_{\mathrm{comp}}(5)}{{}_5\mathrm{C}_{n_{\min}}} = 0.5$$

$$(3,2):\ f_S(n_{\min}) = \frac{f_{\mathrm{comp}}(3) \times f_{\mathrm{comp}}(2)}{{}_5\mathrm{C}_{n_{\min}}} = 0.6$$

この結果より，$n = 5$ の場合においては，(3,2) という配置の方が優れていると言える．

ここで (5) や (3,2) というのがどのような配置を表しているかを考える．(5) によって表現されている配置は，全体が一つの大きな巡回置換となるような配置である．これは，いわゆる「シーケンシャルな配置」である．シーケンシャルな配置とは，それぞれのデータブロック毎に担当のノードを決め，そのノードから見て後続のノードにそのデータの複製を配置する方法である．このような配置は，既存の広域分散システムでしばしば見られる．それに対し，(3,2) という配置は，システム全体をより細かい単位で分割し，それぞれで小さなシーケンシャルな配置を作るという配置方法を表現している．

このケーススタディから示唆される知見は，シーケンシャルな配置は，相関障害への耐性が優れていると

は言えないということである．相関障害への耐性を高めることが第一の目的である場合は，シーケンシャルな配置は採用すべきでないだろう．

ノード数が **8** の場合

次に，もう少しノード数が多い，$n = 8$ の場合について検討する．今回も，$n = 5$ の場合と同様に，次数 1 の巡回置換を含むような配置を除外する．そのような配置を除外した結果残る配置は，(8), (6,2), (5,3), (4,4), (4, 2, 2), (3, 3, 2), (2, 2, 2, 2) の 7 通りである．今回は，これらに加えて，ランダムにデータを配置した場合の結果もあわせて比較する．ランダムな配置を検討するために，計算機を用いて $n = 8$ の場合にとり得るすべての配置を列挙し，それぞれの相関障害への耐性を計算し，平均を求めた．ランダムな配置の相関障害への耐性を計算するにあたり，データブロックの重複を許さない場合と許す場合の両方を計算した．ランダムな配置も，分散システムでしばしば見られる配置である．

上記のそれぞれのケースをプロットしたグラフを図 **4** に示す．この結果からまず読み取れることは，(2, 2, 2, 2) がすべての n_p において最も高い相関障害への耐性を達成しているということである．次に，ランダムな配置の平均は，(2, 2, 2, 2) の配置やシーケンシャルな配置に比べて相関障害への耐性が低いということである．重複を許さない場合は，ランダムな配置はシーケンシャルな配置に比べて若干低い．重複を許す場合は，容易に想像できる通り，他に比べて相関障害への耐性が大幅に低い．

このケーススタディから示唆される知見は，ランダムな配置もやはり，相関障害への耐性が優れているとは言えないということである．相関障害への耐性を高めることが第一の目的である場合は，シーケンシャルな配置と同様，ランダムな配置は採用すべきではない．特に，重複を許すランダム配置は避けるべきである．

ノード数が **29** の場合と **30** の場合

これまでの二つのケーススタディで，相関障害への耐性の観点から，シーケンシャルな配置やランダムな配置よりも優れた配置が存在し得ることが示唆されていた，このケーススタディでは，$n = 29$ および $n = 30$ の場合を取り上げて，より大きな n について，偶数の場合と奇数の場合をそれぞれ検討する．

$n = 29$ および $n = 30$ の場合について，計算機を用いてすべての配置を列挙し，それぞれの配置の持つ相関障害への耐性を求めた．それらの結果を図 **5** と図 **6** にそれぞれ示す．この図では，最も相関障害への耐性が高かったものを強調して表示し，それ以外のものについては区別せずにすべて同じ線で表示している．

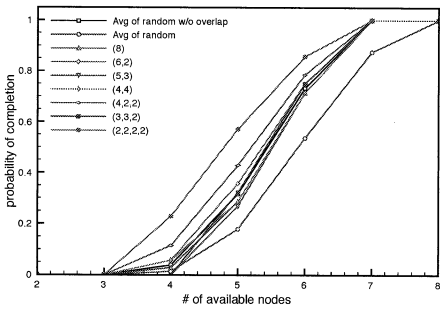

図 **4** $n = 8$ の場合に f_S が取り得る値をすべてプロットした図．但し，次数 1 の巡回置換を含むものを除く．

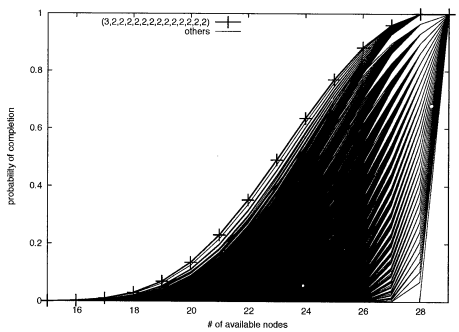

図 **5** $n = 29$ の場合に f_S が取り得る値をすべてプロットした図．最良のケースを強調表示し，それ以外のものは区別していない．

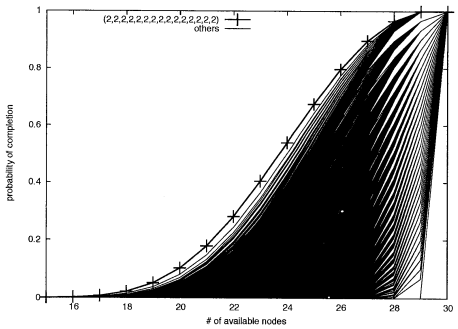

図 **6** $n = 30$ の場合に f_S が取り得る値をすべてプロットした図．最良のケースを強調表示し，それ以外のものは区別していない．

$n = 29$ では $(3, 2, \ldots, 2)$（2 が 13 個続く）というデータ配置が，また，$n = 30$ では $(2, \ldots, 2)$（2 が 15 個続く）というデータ配置が最も高い耐性を示した．この結果から，次数 1 の巡回置換を含まず，かつ，次数 2 の巡回置換をなるべく多く持つような巡回置換を採用することが相関障害への耐性の観点からは

望ましいという結論が示唆される．我々は，この2という数字が $m=2$ という前提に由来していると考えている．

3.2 ノードの持つ複製数が3以上の場合

ここまで，$m=2$ の場合について，データ配置を置換群の元として解釈して取り扱う方法について説明してきた．その発展として，当然，$m \geq 3$ の場合についても同様に評価を行いたいと考えるのは自然である．しかし，$m \geq 3$ の場合については，置換として単純に取り扱うことができないために，$m=2$ の場合とは全く同じ議論は行えない．もし，$m=2$ の場合と同様の議論を試みる場合は，データ配置の解釈を置換から行列へ一般化する必要がある．その結果，ある二つの配置が等価かどうかを判定するという問題はグラフ同型性判定問題[8]に帰着され，取扱いが急に困難になる．

我々は，$m \geq 3$ の場合に対するアプローチとして，二つを検討している．一つは，データの配置にある程度制約を加えた上でグラフ同型性判定問題へ取り組むという方法である．もう一つは，$m=2$ の場合のケーススタディから示唆される結論から出発して，それが $m \geq 3$ の場合でも同様に有効であることを示すという方法である．

4. 議　　論

広域分散システムに保存されたデータの可用性を高めるためのアプローチにerasure codeがある[7]．Erasure codeは，与えられたデータ全体を分割し，ある種の符号化を施すことで，いくつかのデータブロックが欠けていたとしても元のデータ全体を復元することを可能にする．Erasure codeは，必要な数のデータブロックが揃っていれば，確実に，もしくは非常に高い確率で元のデータを復元することができるという優れた性質を持っている．その反面，一般に，データブロックが必要な数だけ揃わない場合は，元のデータを復元できない．それに対し，この論文で議論しているようなデータの複製配置は，少ないノード数でも元のデータが復元できる可能性を，より少ない冗長性で実現できる可能性がある．今後は，これらの性質をうまく組み合わせて，少ない冗長性で効率的に高い耐性を実現できるシステムの設計に取り組みたいと考えている．

5. おわりに

本論文では，広域分散システムにおけるデータブロックの配置が持つ相関障害への耐性について評価する方法について論じた．特に，各ノードが持つデータブロックが高々2の場合について，データ配置を置換群の元として捉えることで，それが持つ互いに素な巡回置換の構造から相関障害への耐性を計算する方法を示した．また，その計算を用いたケーススタディを通じて，シーケンシャルな配置やランダムな配置といったよく見られる配置よりも，次数2の巡回置換で主に構成されるデータ配置の方が相関障害に対して高い耐性を示すという知見を得た．今後は，この知見に対する理論的な検証，および，この配置方法を応用したシステムの実現に取り組みたい．

謝　　辞

本研究の一部は，科学技術振興機構 戦略的創造研究推進事業 CREST の研究プロジェクト「自律連合型基盤システムの構築」の支援によって行われた．

本論文の内容をまとめるにあたって，筑波大学大学院システム情報工学研究科の石川宗寿氏との議論は非常に有益でした．ここに感謝の意を表します．

参 考 文 献

1) Andrews, G. and Ericsson, K.: 整数の分割，数学書房 (2006). 佐藤文広 訳，(原題: Integer Partitions).
2) Dabek, F., Kaashoek, M.F., Karger, D.R., Morris, R. and Stoica, I.: Wide-Area Cooperative Storage with CFS, *SOSP*, pp.202–215 (2001).
3) Haeberlen, A., Mislove, A. and Druschel, P.: Glacier: Highly Durable, Decentralized Storage Despite Massive Correlated Failures, *NSDI* (2005).
4) Kubiatowicz, J., Bindel, D., Chen, Y., Czerwinski, S.E., Eaton, P.R., Geels, D., Gummadi, R., Rhea, S.C., Weatherspoon, H., Weimer, W., Wells, C. and Zhao, B.Y.: OceanStore: An Architecture for Global-Scale Persistent Storage, *ACM ASPLOS*, pp.190–201 (2000).
5) Nath, S., Yu, H., Gibbons, P.B. and Seshan, S.: Subtleties in tolerating correlated failures, *USENIX NSDI* (2006).
6) PlanetLab Consortium: PlanetLab — An open platform for developing, deploying, and accessing planetary-scale services. `http://www.planet-lab.org/`.
7) Plank, J.A.: Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems, *Software — Practice & Experience*, Vol. 27, No.9 (1997).
8) 戸田誠之助：グラフ同型性判定問題，日本大学文理学部 (2001).