Vol.2013-DBS-158 No.7
2013/11/26

数値的なシミュレーションを用いた CAP 定理の検証と対応する分散システムの条件の検討

尾形篤史[†1]　　木村昌臣[†2]

CAP 定理は，2000 年に Eric A. Brewer が提唱した定理である．CAP 定理では，分散システムが持つべき 3 つの性質（一貫性・可用性・分断耐性）のうち，少なくとも 2 つしか同時に満たせないと述べている．2002 年に Seth Gilbert と Nancy Lynch が CAP 定理の証明を可用性が常に成り立つという前提で行った．しかし，証明では CAP 定理の各性質が成り立つ状況を網羅的に示しておらず，様々な分散システムに対して CAP 定理が成り立つことを示していない．また，CAP 定理の各性質は論文によって定義が異なり，曖昧であるという問題がある．そこで，本研究では，CAP 定理の各性質を厳密に定義するため，各性質の条件式をグラフ理論の隣接行列を基に定式化する．そして，シミュレーションによって分散システムを網羅的に検証し，CAP 定理の各性質が成り立つ状況の検討を行い，CAP 定理の一貫性と可用性，一貫性と分断耐性，可用性と分断耐性が成り立つ分散システムの条件を検討する．

Verify CAP Theorem Based on Numerical Simulation and Discuss the Requirement for a Distributed System

ATUSHI OGATA[†1]　MASAOMI KIMURA[†2]

CAP theorem proposed by Eric A. Brewer in 2000. CAP theorem tells that it is impossible for a distributed system to concurrently have the following three properties: consistency, availability, and partition-tolerance. In 2002, Seth Gilbert and Nancy Lynch proved this theorem. However, in their proof, they assumed a system always guarantee availability. Namely, they did not discuss that every distributed system held CAP theorem. Moreover, in many other studies, three properties were not given a clear definition and were discussed ambiguously. In this study, we gave these three properties mathematical definitions based on an adjacency matrix. Based on this, we conducted a simulation to verify whether CAP theorem holds in every distributed system. Finally, we discuss the requirement of a distributed system that satisfy CA, AP or CP.

1. はじめに

CAP 定理は 2000 年に Eric A. Brewer が提唱した分散システムに関する定理[1]である．この定理は．分散システムが持つべき性質である，一貫性，可用性，分断耐性の 3 つの性質うち，少なくとも 2 つの性質しか同時に満たすことができないという定理である．Brewer が定義した CAP 定理において分散システムが持つべき 3 つの性質を以下に示す．

- 一貫性

一貫性は，データを保持するすべてのノードが単一の最新データを保持していることを保証する．一貫性はネットワークが分断されたときも，データを保持するノードが単一の最新データを保持していなくてはならない．

- 可用性

可用性は，システムに分断が起きた場合やノードの故障が起きた場合でも，システムが応答することを保証する．システムに分断が起きた場合，分断された各システムはすべての機能を保持しなくてはならない．例えば，ユーザから読み込みや書き込みの要求があった場合には，システムは読み込み処理と書き込み処理の両方を処理しなくてはならない．そのため，読み込みか書き込みのどちらかができない場合，故障したとみなし，可用性は満たさない．

- 分断耐性

分断耐性は，システムがネットワークの分断により，1 つのグループが複数のグループに分割されたとき，それぞれのグループが独立に動作することを保証する．ただし，ここでの応答とは，可用性の応答とは異なり，応答の制限を行ってもよい．応答の制限とは，読み込みや書き込みができたシステムが読み込みのみや書き込みのみに制限することである．

しかし，CAP 定理の各性質は定義が曖昧であるため，各性質に様々な見解がある[2][3].

2002 年に Seth Gilbert と Nancy Lynch が CAP 定理の証明として可用性が常にある前提で証明した[4]．しかし，可用性が常にある前提で議論したため，可用性と一貫性を満たす場合と可用性と分断耐性を満たす場合の議論しか行っていないという問題があった．

そこで，本研究は，定義が曖昧であるという問題を解決するため，言葉で定義されていた CAP 定理の一貫性，可用性，分断耐性の定義を数式で定式化する．その結果，曖昧であった CAP 定理の各性質を明確に定義することができる．本研究では，一貫性，可用性，分断耐性の定義をグラフ理論に基づき定式化する．そして，CAP 定理の一貫性と

†1 芝浦工業大学大学院理工学研究科電気電子情報工学専攻
Graduate School of Engineering and Science, Shibaura Institute of Technology
†2 芝浦工業大学工学部情報工学科
Department of Information Science and Engineering, Shibaura Institute of Technology

ⓒ2013 Information Processing Society of Japan

可用性，分断耐性をすべて満たすような分散システムが存在しないかシミュレーションを用いて定量的な検証を行う．また，シミュレーションにより様々な分散システムを網羅的に検証することができるため，一貫性，可用性，分断耐性のいずれかが成り立つような分散システムを分類し，CAP 定理に対応する分散システムの検討を行う．

2. Seth Gilbert と Nancy Lynch による CAP 定理の証明

本章では，Seth Gilbert と Nancy Lynch による可用性がある前提での CAP 定理の証明[4]の内容と問題点を述べる．

Seth Gilbert と Nancy Lynch の証明では，単純なデータモデルを用いて，CAP 定理の 3 つの性質が同時に成り立たないことを示している．例えば，2 つのノードから構成されるシステムを想定する．そして，ユーザか一方のノードに対して更新処理を行い，データを書き換えたとき，CAP 定理の各性質の条件を満たすことができるか議論している．なお，このシステムは 2 つのノードから構成されるため，可用性は常に満たされている状態である．

まず，可用性が成り立つ前提で一貫性を満たすことを考える（図 1）．一貫性を満たすためには，ノード 1 のデータに更新があった場合，そのデータをノード 2 に同期しなくてはならない．データを同期するため，ノード 1 とノード 2 は通信可能であり，ノード同士が繋がっていなくてはならない．このようにすることで，ノード 1 とノード 2 のデータを単一に保つことができる．

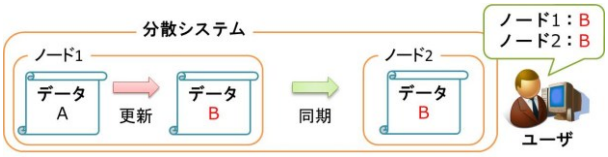

図 1　可用性を満たす前提で一貫性を満たす例

Figure 1　An example that a system guarantees Availability and Consistency

次に，上記の可用性が成り立つ前提で一貫性を満たしているとき，分断耐性を満たすことを考える（図 2）．一貫性を満たすことを考えた場合と同様に，ノード 1 のデータに更新があったとする．システムは一貫性を保つために同期しようとするが，伝送路に分断があるため，同期をとることができない．つまり，分断耐性を満たそうとすると，同期をすることができないため，ノード 1 とノード 2 で違うデータを保持してしまう．従って，一貫性を諦めなければならない．

その結果，3 つの性質を同時に満たすことができないため，CAP 定理は正しいとしている．

図 2　可用性を満たす前提で分断耐性を満たす例

Figure 2　An example that a system guarantees Availability and Partition-Tolerance

Seth Gilbert と Nancy Lynch の証明では，可用性を満たしている前提で，一貫性と分断耐性が同時に成り立たないことを示している．つまり，彼らの証明では，可用性と一貫性を満たすシステムと可用性と分断耐性を満たすシステムの議論しかしておらず，一貫性と分断耐性を満たすシステムの議論をしていない．従って，彼らの証明は不十分であると考えられる．

また，彼らの証明では，データの読み込みや書き込みに関する制限を考慮していない．彼らの前提では，システムは読み込みも書き込みも両方行えると考えていた．しかし，CAP 定理の議論を行うためには，読み込みと書き込みの機能に関する考慮が必要である．分散システムで一貫性を保つためには，書き込みの制限を行うことが考えられる．可用性は，分断が起きる前と同じ機能を保持しなくてはならないため，データの読み込みや書き込みの制限を行ったときに失われてしまうと考えられる．このように，CAP 定理の議論では，データの読み込みや書き込みを考慮することが必要であると考えられる．

そのため，本研究では一貫性と分断耐性を満たすシステムでも CAP 定理が成り立つかどうかを検証する．また，分散システムでデータの読み込みや書き込みの機能を考慮したときでも CAP 定理が成り立つか検証する．そして，様々な分散システムで CAP 定理が成り立つかを網羅的に検証する．

3. CAP 定理の定式化

本章では，CAP 定理の各性質の有無を表す条件を定式化する．分散システムを数式で表現するため，グラフ理論の隣接行列を用いてネットワークを表現する．

3.1 分散システムの定式化

3.1.1 分散システムの構成

分散システムには，マスターノードとスレーブノード（データノード）から構成されているシステムが存在する．マスターノードは，ユーザからの要求を受け付け，スレーブノードを管理するノードである．スレーブノードは，実際にデータを保持するノードである．

ⓒ2013 Information Processing Society of Japan

3.1.2 システム上のノードの繋がりを表現する式

分散システムのノード同士が応答するための条件には，以下の 3 つの条件が必要である．

1　各ノードが伝送路で繋がっている
2　ノード間の伝送路に障害が存在しない
3　伝送路で繋がる 2 つのノードが応答可能である

なお，伝送路とは，データの授受を行うためのノード間のリンクのことである．

分散システム上のノードの繋がりを表現するため，3 つの条件を行列と変数を用いて表現する．

まず，各ノードが伝送路で物理的に繋がっていることを行列Lで表現する．行列Lは，ノードiとノードjが伝送路で繋がっているとき，$L_{\mathrm{ij}}=1$となり，伝送路で繋がっていないとき，$L_{\mathrm{ij}}=0$となる．

次に，ノード間の伝送路に障害がないことを行列hで表現する．行列hでは，ノードiとノードjに障害がないとき，$h_{\mathrm{ij}}=1$となり，障害があるときは$h_{\mathrm{ij}}=0$となる．

最後に，分散システムが伝送路で繋がっていた場合でも，伝送路で繋がっているノード同士が応答可能でなければならない. そのため，ノードが応答可能であるかを 2 値変数φで表現する．ノードiが応答可能であるとき，$\varphi_i=1$となり，応答できないとき，$\varphi_i=0$となる．また，通信を行うためには，伝送路で繋がる 2 つのノードが共に応答可能でなければならない. そのため, 2 つのノードの変数φの積をとる．

分散システムのノード同士が応答するためには，この 3 つの条件がすべて成り立つときにノード間の通信が可能となる．そのため，各条件の積をとる．従って，システム上のノードの繋がりを表現する式を以下のように定義する．

$$A_{ij}=L_{ij}h_{ij}\varphi_i\varphi_j \tag{1}$$

3.1.3 スレーブノードの読み込みと書き込みの考慮

マスターノードについては応答可能かどうかを判断するため，変数φを用いて表現する．

一方で，スレーブノードには，データを読み込む機能と書き込む機能がある．そのため，応答可能であるかを表す 2 値変数φの代わりに, 読み込み可能であるかを表す 2 値変数φ^rと書き込み可能であるかを表す 2 値変数φ^wを用いる．

3.1.4 システムの経路の定式化

分散システムは，複数のノードを利用して応答する場合がある．そのため，システムが利用するノードの経路を定式化する．

システムが応答するためには，応答するときに使用するすべてのノードが伝送路で繋がっており，かつ，応答可能でなければならない．例えば，3 つのノード（ノード 0，ノード 1，ノード 2）から構成されるシステムで 3 つのノードを利用して応答する場合，ノード 0 とノード 1 が繋がっており，かつ，ノード 1 とノード 2 が繋がっている必要がある．つまり，システムが応答するために必要な伝送路上にあるノード間では，行列Aの成分の値が 1 でなければならない．

そのため，行列Aの成分をシステムが使用するノードの順に積をとるS^xを定義する，S^xの値が 1 であるとき，そのシステムは応答可能であると判断できる．従って，システムが応答できるかどうかを表現する式を以下のように定義する．

$$S^x=\prod_{k=0}A_{S_k^xS_{k+1}^x}=1 \tag{2}$$

3.2 CAP 定理における各性質の有無を表す条件式

3.1 節で定義した分散システムの条件式を用いて CAP 定理の 3 つの性質の有無を表す条件を定式化する．本研究では，単純化するため各スレーブノードは同じデータを保持しているとする．また，ネットワークによるデータ転送の遅延は無いものとする．

3.2.1 一貫性

一貫性は，すべてのスレーブノードが単一の最新データを保持しなくてはならない．一貫性を満たす場合として以下の 2 つがある．

1. すべてのスレーブノードが繋がっている場合
2. すべてのスレーブノードが書き込み不可能な場合

1.の場合は各スレーブノードが繋がっていれば，すべてのスレーブノードに最新データを転送できるため，どのスレーブノードも単一の最新データを保持することができる．従って，一貫性を満たすことができる．

2.場合は，すべてのスレーブノードに書き込みが行われない場合である．この場合は，各スレーブノードのデータが更新されないため, 常に同じデータを保持する. 従って，一貫性を満たすことができる．

また，これら 2 つの場合は，両方を満たす必要はなく，どちらか 1 つを満たせば，一貫性を満たすことができる．

まず，1 つめの条件を定式化する．すべてのスレーブノードが繋がっていることを判断するため，距離行列を求める．これは，各スレーブノード間の距離が有限であれば，各スレーブノードが繋がっていると判断できるためである．よって，1 つめの場合が成り立つか判断するためには，行列Aをn乗し，各成分が 1 以上となるか確認する．

また，データを保持しているのはスレーブノードのみであるため, 一貫性では, スレーブノードのみを対象とする．そのため，スレーブノードを表すベクトルDを定義し，行列Aの両側から乗ずることでスレーブノードのみを対象とする．このとき，すべてのスレーブノードで同じデータを保持するためには，マスターノードを経由してデータの同期をとることも考えられるため，行列Aのn乗をしたのち，ベクトルDを乗ずる．

以上より，一貫性があると考えられる条件式は，以下のようになる．

$$^{\exists}n<\infty\,,\ ^{\forall}i,j\,,(D^t(A)^nD)_{i,j}\geq 1 \tag{3}$$

次に，2 つめの条件を定式化する．すべてのスレーブノードで書き込みが不可能であることを表現するため，スレ

ーブノードが書き込み可能であるかを表す変数φ^wの和をとる．そして，その値が 0 であるとき，すべてのスレーブノードが書き込み不可能であると判断できる．すべてのスレーブノードが書き込み不可能であることを表現する式は以下のようになる．

$$\sum_{k=0} \varphi_k^w = 0 \quad (4)$$

3.2.2 可用性

可用性は，分散システムに故障が起きたときに，他のシステムを用いて応答し続けることである．可用性では，全ての機能（読み込みと書き込み）を保持しなければならない．もし，読み込みか書き込みのどちらかの機能を保持しないときは，故障とみなし，可用性は満たさないとする．従って，必ず両方の機能を保持しなくてはならないため，φ^rとφ^wの積をとる．また，伝送路で繋がっているノード同士で同じ機能を保持していなくては，その機能を実現できない．そのため，ノードが持つ機能ごとに積をとる．

可用性での読み込みと書き込みに関する条件を表現するために，(1)式を変更する．読み込みと書き込みの両方を満たす場合の行列$A^{(a)}$は以下のようになる．

$$A_{ij}^{(a)} = L_{ij} h_{ij} \varphi_i^r \varphi_j^r \varphi_i^w \varphi_j^w \quad (5)$$

可用性を満たすためには，1 つ以上のシステムが応答すればよい．図 3 のように分散システム内に 2 つのシステム（システム A とシステム B）がある場合，どちらかのシステムが応答可能であればよい．この条件を(2)式を用いて表すと，$S^{\mathrm{A}} = 1$または$S^{\mathrm{B}} = 1$と表せため，その和が 1 以上であれば，複数のシステムが応答可能であると考えられる．

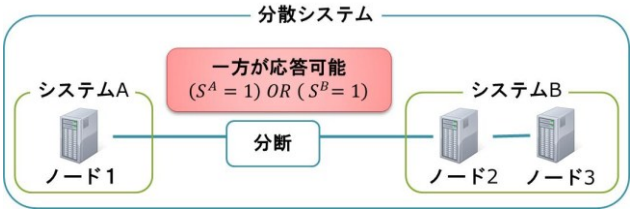

図 3　可用性を満たすための条件
Figure 3　A condition to satisfy availability

以上の考えを一般のシステムに適用できるように条件式を定義する．(2)式をシステムごとに計算し，和をとればよいため，可用性があると考えられる条件式は，(2)式と(5)式を用いて以下のように定義できる．

$$\sum_{l=0} S^i = \sum_{l=0} \prod_{k=0} A_{S_k^l S_{k+1}^l}^{(a)} \geq 1 \quad (6)$$

3.2.3 分断耐性

分断耐性は，システムに分断が起きたとき各システムが独立に動作することを保証する．分断耐性では，可用性と異なり一部の機能（読み込みか書き込み）に制限してもよい．そのため，分断耐性ではどちらかの機能を保持していればよいため，φ^rとφ^wの論理和をとる．また，伝送路で繋がっているノード同士が同じ機能を保持していなくてはならないため，ノードが持つ機能ごとに積をとる．

分断耐性での条件を表現するために，(1)式を変更する．読み込みか書き込みのどちらかの機能を満たす場合の行列$A^{(p)}$は以下のようになる．

$$A_{ij}^{(p)} = L_{ij} h_{ij} \left(\varphi_i^r \varphi_j^r + \varphi_i^w \varphi_j^w - \varphi_i^r \varphi_j^r \varphi_i^w \varphi_j^w \right) \quad (7)$$

分断耐性を満たすためには，各システムが独立に応答すればよい．図 4 のようにシステム A とシステム B の間に分断が起き，2 つの独立したシステムに分断された場合，各システムが独立に動作しなければならない．この条件を(2)式を用いて表すと，$S^{\mathrm{A}} = 1$かつ$S^{\mathrm{B}} = 1$と表せるので，その積が 1 であれば，各システムが独立に応答可能であると考えられる．

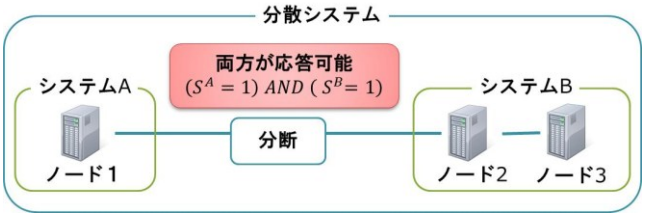

図 4　分断耐性を満たすための条件
Figure 4　A condition to satisfy Partition-Tolerance

以上の考えを一般のシステムに適用できるように条件式を定義する，(2)式をシステムごとに計算し，積をとればよいため，分断耐性があると考えられる条件式は，(2)式と(7)式を用いて以下のように定義できる．

$$\prod_{l=0} S^l = \prod_{l=0} \prod_{k=0} A_{S_k^l S_{k+1}^l}^{(p)} = 1 \quad (8)$$

4. シミュレーション

本研究では，CAP 定理が様々な分散システムで成り立つか網羅的に検証するため，シミュレーションを行う．

4.1 シミュレーションの前提条件と方法

シミュレーションでは，マスターノードとスレーブノード（データノード）がある分散システムを想定する．このとき，マスターノードの数を 2 台，スレーブノードの数を 3 台とする．また，各スレーブノードは同じデータを保持しているとする．マスターノードとスレーブノードから構成される分散システムは，システム内に 2 種類のノードが存在するときに応答することができるため，2 種類のノードが存在する場合のみ応答可能とする．

シミュレーションでは，まず，シミュレータに各種類のノード数を設定する．次に，そのノード数から考えられるすべての分散システムの構成（隣接行列）を作成する．最後に，各分散システムが CAP 定理の 3 つの性質を満たすこ

とができるかを本研究で定義した式をもとに検証する．

この手順に従って，シミュレーションでは，システムに機能の制限がない場合とすべてのスレーブノードのデータが書き込み不可能な場合の 2 種類を行った．

4.2 シミュレーション結果

シミュレーションの結果，1,024 通りの隣接行列が作成された．1,024 通りのシミュレーション結果のうち，互いに独立な結果は 120 通りであった．その中で，各性質の条件を満たさなかった結果を示す．

- 一貫性または可用性を満たさない場合

一貫性または可用性を満たさなかった結果は，図 5 のように 2 つのマスターノードとスレーブノードから構成されるシステムであった．この例の結果を表 1 に示す．

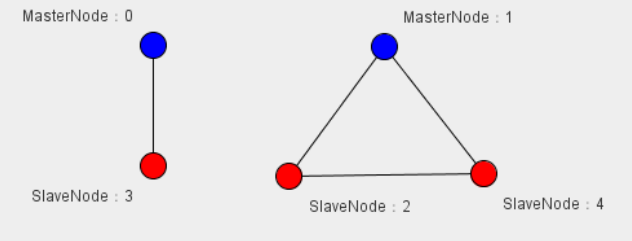

図 5　マスターノードとスレーブノードを含む 2 つのシステムに分断されたときの構成図

Figure 5　A system configuration where each part of the system contains both master node and slave node.

表 1　図 5 に対するシミュレーション結果

Table 1　A result of the simulation for the system in Figure 5

性質	読み書き可能	書き込み不可能
一貫性	満たさない	満たす
可用性	満たす	満たさない
分断耐性	満たす	満たす

すべてのノードが読み書き可能な場合は，一貫性のみ満たさないという結果となった．一貫性を満たさない理由として，システムに更新が起きたとき，スレーブノードによって違うデータを保持してしまうためであると考えられる．可用性は，2 つのシステムが存在し，読み書きが行えるため，満たすことができたと考えられる．分断耐性は，両方のシステムが独立に応答することができるため，満たすことができたと考えられる．

一方，書き込みの制限を行ったとき，可用性のみ満たさないという結果となった．可用性を満たさない理由として，可用性の条件である読み込みと書き込みの両方ができるという条件を満たしていないためであると考えられる．一貫性は，スレーブノードがすべて繋がっていないにも関わらず満たすという結果となった．これは，分断されてできた両方のシステムで書き込みが行われないため，すべてのスレーブノードで常に同じデータを保持するためであると考えられる．分断耐性は，両方のシステムで読み込みができるため，分断耐性の読み込みと書き込みの機能のうち，どちらかを保持しているという条件を満たしたため満たすことができたと考えられる．

- 分断耐性を満たさない場合

分断耐性を満たさなかった結果は，図 6 のように「マスターノードのみのシステム」と「マスターノードとスレーブノードを含むシステム」に分断された場合であった．この例のシミュレーションの結果を表 2 に示す．

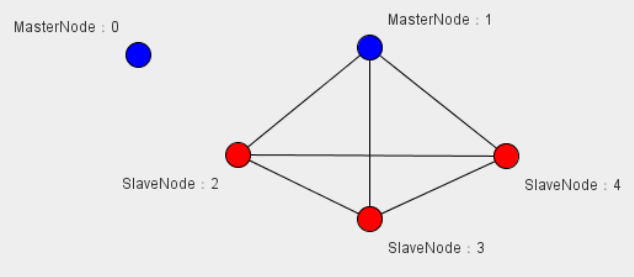

図 6　マスターノードのみのシステムに分断されたときの構成図

Figure 6　A system configuration where the system contains master node only.

表 2　図 6 に対するシミュレーション結果

Table 2　A result of the simulation for the system in Figure 6

性質	読み書き可能	書き込み不可能
一貫性	満たす	満たす
可用性	満たす	満たさない
分断耐性	満たさない	満たさない

すべてのノードが読み書き可能な場合は，分断耐性のみ満たさないという結果となった．図 6 のようにマスターノードのみのシステムが存在する場合，マスターノードのみではスレーブノードがないため，応答することができない．従って，分断耐性の条件である，両方のシステムが独立に動作することを満たさなかったため，分断耐性を満たさなかったと考えられる．一貫性は，すべてのスレーブノードが繋がっているため，満たすという結果となった．可用性は，マスターノードとスレーブノードを含むシステムを用いて読み書きの要求に応答することができるため，満たすという結果となった．

書き込みの制限がある場合は，可用性と分断耐性を満たさないという結果となった．分断耐性を満たさない理由は，上記の理由と同じであると考えられる．また，可用性は，機能の制限があることにより，すべての機能を保持するという条件に反したため，満たさないという結果となった．

ⓒ2013 Information Processing Society of Japan

4.3 シミュレーション結果のまとめ

シミュレーションの結果から，CAP 定理の 3 つの性質をすべて満たす分散システムは，システム内に分断がない分散システムであった．システム内に分断がない場合は，すべてのノードが繋がっており，読み込みや書き込みの制限もされないため，すべての性質を満たすことができたと考えられる．システム内に分断がある場合，3 つの性質をすべて満たす分散システムは存在せず，最大で 2 つの性質しか満たすことができなかった．

以上より, 分断が起きない前提の分散システムでは, CAP 定理の 3 つの性質の条件をすべて満たすことができる．しかし, 分断耐性は, 分断が起きる前提で定義がされている．そのため, 分断が起きる前提の分散システムで CAP 定理を考えたときは，3 つの性質のうち最大で 2 つの性質の条件しか満たすことができなかった．そのため，分断が起きる前提の分散システムは，CAP 定理が成り立つ．

次に，シミュレーションの結果から，各性質の条件を満たす分散システムの条件を検討する．

一貫性を満たす分散システムは，すべてのノードが書き込み可能でかつすべてのスレーブノードが繋がっている分散システムであった．また，すべてのスレーブノードが繋がっていない場合でも，書き込みの制限を行うことにより一貫性を満たすことができた．

可用性を満たす分散システムは，読み込みと書き込みが両方機能する分散システムであった．そのため，機能の制限をかけた場合は，可用性を満たさない分散システムとなった．

分断耐性を満たす分散システムは，複数のシステムに分断されたとき，すべてのシステムにマスターノードとスレーブノードが 1 台以上含まれるように構成されている分散システムであった．

5. まとめと今後の展望

本研究は，CAP 定理の各性質を定式化し，CAP 定理が様々な分散システムで成り立つかシミュレーションを行い網羅的に検証した．また，CAP 定理の一貫性や可用性を議論するときにデータの読み込みや書き込みの機能を考慮する必要があると考えられるため，本研究では，読み込みと書き込みを考慮し各性質の条件式を定義した．

シミュレーションでは，マスターノードとスレーブノードを含む分散システムを想定し，シミュレーションで考えられるすべての分散システムに対して CAP 定理が成り立つか検証した．その結果，分断が起きる前提の分散システムでは，CAP 定理の 3 つの性質のうち最大で 2 つの性質の条件しか満たすことができなかった．そのため，分断が起きる前提の分散システムは，CAP 定理が成り立つという結果となった．

今後の課題として，今回行ったシミュレーションの結果が分散システムのノード数によらず，一般的に成り立つことを示す方法の検討を行う．また，従来の分散システムと異なる新たな分散システムの構造と実現可能性をシミュレーションの結果をもとに検討する．

参考文献

1) Eric A. Brewer: Towards robust distributed systems, Proceedings of the nineteenth annual ACM symposium on Principles of distributed computing , pp.7-10 (2000).
2) C. Hale: You Can’t Sacrifice Partition Tolerance, http://codahale.com/you-cant-sacrifice-partition-tolerance.
3) Eric A. Brewer: CAP twelve years later: How the "rules" have changed, IEEE Computer Volume 45 Issue 2, pp.23-29 (2012).
4) Seth Gilbert and Nancy Lynch: Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services, ACM SIGACT News Volume 33 Issue 2, pp.51-59 (2002).

ⓒ2013 Information Processing Society of Japan