

サーバのリソースが一定である時の、システムの可用性が最大となるサービスの実装形態や各ソフトウェアの冗長度合いを探る

1 システムモデル

• サービス (M 個):

- サービスはアプリケーションシステムを実現するために必要となる、部分的な機能を提供する。 i 番目のサービスを m_i と表す
- サービス m_i の可用性を a_{m_i} とする

• サーバ (H):

- サーバは複数のソフトウェアをホスト可能。 j 台目のサーバを h_j と表す
- 全てのサーバの可用性は等しいものとし a_h とする
- 各サーバに 1 つのソフトウェアがホストされるものとする
- サーバ j にソフトウェア k がホストされている状態を $b_{(j,k)}$ という変数を用いて $b_{(j,k)} = 1$ と表し、そうでなければ 0 とする

• ソフトウェア (S 個):

- サービスをサーバへ実装する際の実装単位。 k 個目のソフトウェアを s_k と表す
- 各サービスは 1 つのソフトウェアにのみ内包されており、ソフトウェア k にサービス i が内包されている状態を $c_{(k,i)}$ という変数を用いて $c_{(k,i)} = 1$ と表し、そうでなければ 0 とする。 s_k に含まれるサービスの数を s_{k_m} とする。 s_{k_m} は下のよう表される

$$s_{k_m} = \sum_i c_{k,i}$$

- ソフトウェア s_k の可用性を a_{s_k} とし、ソフトウェア s_k が内包するサービスの可用性の積に等しいものとする。 a_{s_k} は下のよう表される

$$a_{s_k} = \prod_{i|c_{k,i}=1} a_{m_i}$$

- ソフトウェアは複数のサーバに冗長化して配置可能
- ソフトウェア s_k がホストされているサーバ台数 r_k は下のよう表される

$$r_k = \sum_j b_{j,k}$$

また、ソフトウェア s_k が必要とするリソース R_k は、サービスが追加されるごとに必要となるリソースの増加量 r_{add} を用いて以下のように表される。

$$R_k = r_k \cdot \{1 + (s_{k_m} - 1) \cdot r_{add}\}$$

- ソフトウェアがシステム内で少なくとも 1 つ動いていれば、そのソフトウェアは可用であるとする。そのため、システムにおけるソフトウェア s_k の可用性は下のよう表される

$$1 - (1 - a_{s_k} \cdot a_h)^{r_k}$$








• アプリケーションシステム:

全てのサービスが実行されることでアプリケーションシステムは可用となる。そのため、アプリケーションシステムの可用性 a_{sys} は下のよう表される

$$a_{sys} = \prod_k (1 - (1 - a_{s_k})^{r_k})$$

注釈の一覧 :honda_result.pdf

ページ :1

	番号 : 1	作成者 : hasegawa	タイトル : ハイライト表示	日付 : 2024/07/19 16:25:01
<div>- 添字の m は付けてはいけない (変数の定義に m が入っていないから)</div> <div>- s という変数はソフトウェアそのものを表すのに使っているおので、ソフトウェア数に同じ変数sは使ってはいけない。</div>				
	番号 : 2	作成者 : hasegawa	タイトル : ノート注釈	日付 : 2024/07/19 16:25:57
<div>k, i にカッコがない</div> <div>これ以降にも同じミスがあるので確認すること。</div>				
	番号 : 3	作成者 : hasegawa	タイトル : ノート注釈	日付 : 2024/07/19 16:26:33
<div>リソース数</div> <div>とか</div> <div>リソースの量</div> <div>とかにすべきです。</div>				
	番号 : 4	作成者 : hasegawa	タイトル : 挿入テキスト	日付 : 2024/07/19 16:27:21
<div>と仮定する。</div>				
	番号 : 5	作成者 : hasegawa	タイトル : 挿入テキスト	日付 : 2024/07/19 16:27:11
<div>を決定する係数</div> <div> r は増加量そのものは表してはいない。</div>				
	番号 : 6	作成者 : hasegawa	タイトル : 取り消し線	日付 : 2024/07/19 16:27:49
	番号 : 7	作成者 : hasegawa	タイトル : 挿入テキスト	日付 : 2024/07/19 16:28:22
<div>可用である</div> <div> 「実行」という概念は、このシステム全体のどこにも存在しない。</div>				

2 ソフトウェアの冗長化度やサービスの実装形態の最適化

- 決定変数: $c_{k,i}, b_{j,k}$
- 目的関数: u_{sys}
- 制約条件: システムのリソースを R で表す時、制約条件は以下の通り
 - システムにおいて使用される合計リソースは R 以下

$$\sum_k R_k \leq R$$

- 各サービスが1つのソフトウェアにのみ内包されている

$$\sum_k c_{k,i} = 1, \forall i$$

- 各サーバに1つのソフトウェアがホストされている

$$\sum_k b_{j,k} = 1, \forall j$$

この最適化問題は、サービスが多い場合に総当たりでやると NP 困難となるため、ヒューリスティックなアルゴリズムを考える必要がある。

アルゴリズムを考えるにあたり、システムにおけるソフトウェアの数、サービスがどのソフトウェアに実装されるかを表すサービス実装形態、それぞれのソフトウェアの冗長化数について、それぞれの要素がシステム可用性に与える影響を解析的評価を通して、明らかにする。

3 解析的評価

本研究では、10 個のサービスを含むシステムにおいて、ソフトウェア数、サービス実装形態、冗長化度合いの3つの要素がシステム可用性に及ぼす影響を、リソースおよび r_{add} の変化に応じて解析した。各要素の影響を評価するため、以下の3つの分布を分析した：

- 様々な実装形態に対する、冗長化パターンとソフトウェア数の網羅的探索による最適な可用性の分布
- 様々な冗長化パターンに対する、実装形態とソフトウェア数の網羅的探索による最適な可用性の分布
- 様々なソフトウェア数に対する、冗長化パターンと実装形態の網羅的探索による最適な可用性の分布

図2に示す9つのグラフは、横軸にシステム可用性（範囲は0.8~1.0）、縦軸にシステム可用性の累積密度関数を表している。プロットの横幅が大きいほど、当該要素がシステム可用性に与える影響が大きいことを示す。全グラフにおいて、すべてのサービス可用性およびサーバ可用性は0.99に設定した。ソフトウェアの冗長化度合いは最大4とし、95%以上のリソース使用率を示すパターンを評価対象とした。

図1にソフトウェア数、サービス実装形態、冗長化度合いの3つの要素とグラフの対応を示す。緑色の線がサービスの実装形態による影響を表し、黄色の線が冗長化度合いによる影響を表し、青色の線がソフトウェア数による影響を表す。

ページ :2

番号 : 1 作成者 : hasegawa タイトル : ノート注釈 日付 : 2024/07/19 16:29:05

ソフトウェア総数 S も決定変数のはずです。

番号 : 2 作成者 : hasegawa タイトル : ノート注釈 日付 : 2024/07/19 16:29:15

maximize
と書く。

番号 : 3 作成者 : hasegawa タイトル : ハイライト表示 日付 : 2024/07/19 16:35:16

この3つを、式で表現してください。これ重要です。

(その通りに評価ができているかの目安にもなります)

例えば1つ目は、

$c_{(k,j)}$ のセット C が1つ与えられた時に、2.の最適化問題 (ただし決定変数から $c_{(k,j)}$ を除いたもの を問いた時の目的関数の値を、 $a_{\text{sys}}(C)$ とする。

で、全ての C に対する $a_{\text{sys}}(C)$ の分布を見る。

ということですよ。

、、、と、ここまで書いて、ちょっと疑問が出てきています。

$c_{(k,j)}$ の1つのセットを決めるには、サービス数 S を決める必要がありますよね。そうすると、この1つ目の話はいったいどうなっているのかわかりません。 S も決めているのに、解く最適化問題の決定変数にも S がある、ということになっている、、、。

、、、???

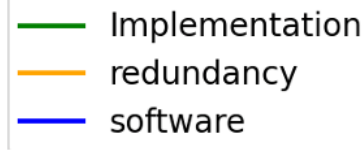


Figure 1: それぞれの要素とラベルの対応

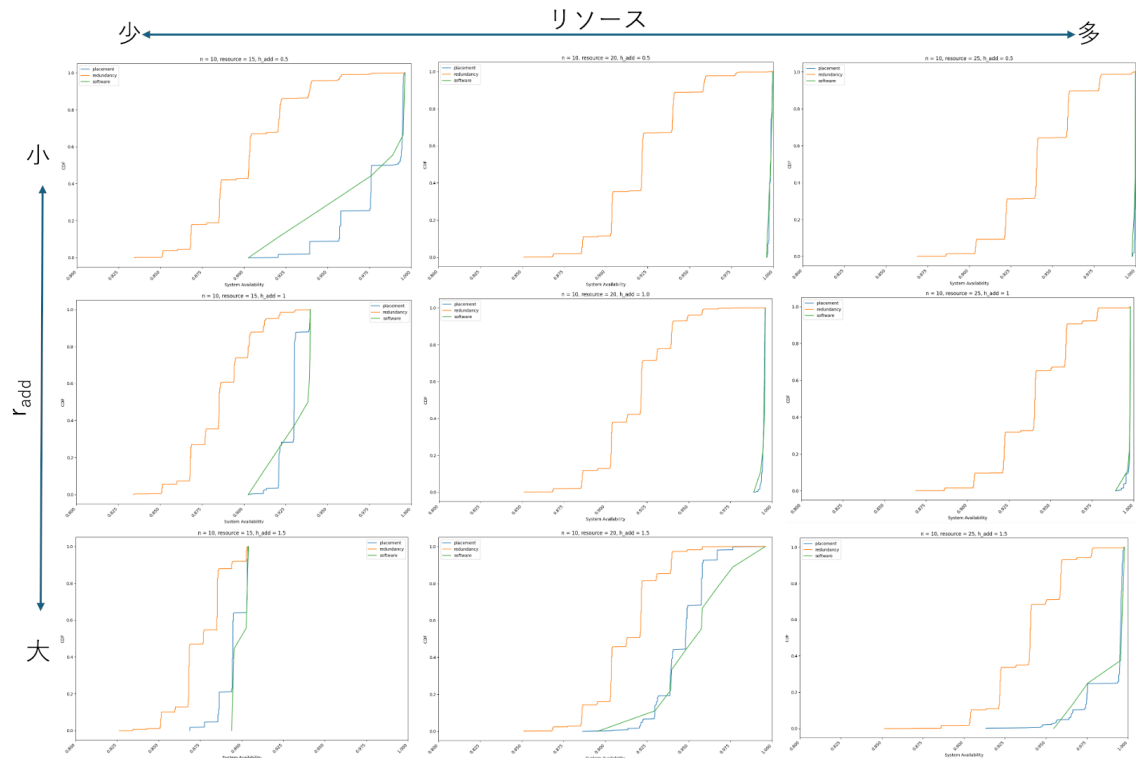


Figure 2: 3つの要素が与える影響の比較

分析結果とその考察

- r_{add} が小さく、リソースが多い場合、冗長化度合いがシステム可用性に最も顕著な影響を与える。これは、システム内のソフトウェアが十分に冗長化されると、サービスの実装形態やソフトウェア数に依存せず、高可用性が達成されるためである。
- リソースが小さいか、 r_{add} が大きい場合、冗長化度合いに加え、サービスの実装形態やソフトウェア数も影響を及ぼす。これは、リソース制約下で十分な冗長化が実現できない場合、サービスの実装形態やソフトウェア数の最適化も重要となるためである。
- リソースが小さい時に、 r_{add} が増大するほど各要素の影響は減少する傾向にある。これは、複数のサービスを内包するソフトウェアが要求するリソースが増大し、冗長化数、サービスの実装形態、ソフトウェア数の変動範囲が制限されるためである。
- ソフトウェア数とサービス実装形態は同様の影響を与えている。これは、各ソフトウェア数 (1 10)

このページには注釈はありません

において、サービス実装形態によるシステム可用性の変化が小さいため、ソフトウェア数とサービス実装形態が似たようなシステム可用性の分布を取るのだと考えられる（プロットで確認します）

3.1 サーバ可用性の変化による影響

次に、サーバ可用性の変化がこれらの要素に与える影響を検討した。図4に示す9つのグラフは、横軸にシステム可用性（範囲は0.4～1.0）、縦軸にシステム可用性の累積密度関数を表している。全グラフにおいて、すべてのサービス可用性は0.99に固定し、サーバ可用性を0.9、0.95、0.99と変化させた。ソフトウェアの冗長化度合いは最大4とし、95%以上のリソース使用率を示すパターンを評価対象とした。

図3にそれぞれのサーバ可用性におけるソフトウェア数、サービス実装形態、冗長化度合いの3つの要素とグラフの線の対応を示す。緑色の線がサービスの実装形態による影響を表し、黄色の線が冗長化度合いによる影響を表し、青色の線がソフトウェア数による影響を表す。また、線の種類がサーバ可用性の違いを表しており、実線が0.99、破線が0.95、一点鎖線が0.9の場合である。

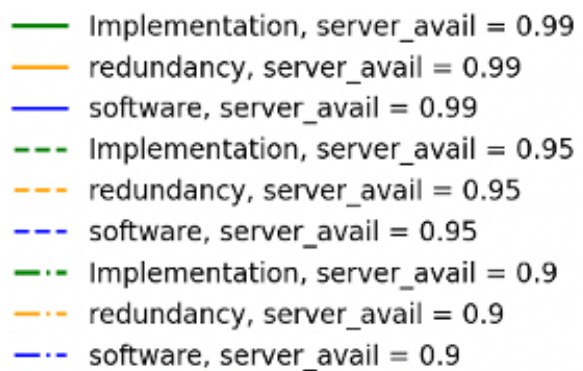


Figure 3: それぞれの要素とラベルの対応

このページには注釈はありません

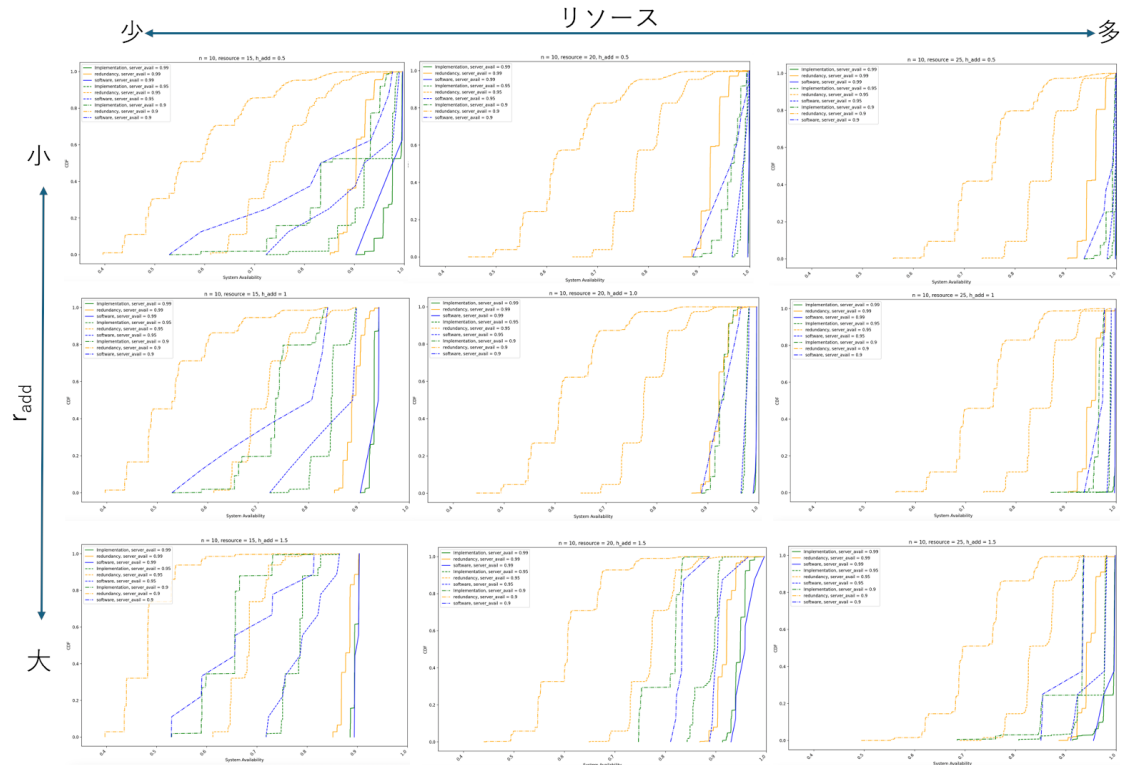


Figure 4: サーバ可用性変化時の比較

分析結果とその考察

- サーバ可用性の低下に伴い、全要素がシステム可用性に及ぼす影響が増大する。これは、ソフトウェア数の増加、冗長化度合いの向上、サービス実装により、サーバ可用性の影響が顕著となり、結果としてシステム可用性が低下するためである。
- リソースが少なくなることで、サーバ可用性の低下に伴うサービス実装形態・冗長化度合い・ソフトウェア数の影響が増大する。これは、リソース制約により、特定のサービス実装形態で十分な冗長化が実現できないなどの理由で、サーバ可用性低下の影響を強く受けるためである。
- r_{add} が大きい時、サーバ可用性の低下に伴いソフトウェア数とサービス実装形態の影響が大きくなる。これは、 r_{add} の増大により1つのソフトウェアに多数のサービスを内包する場合、リソース要求が増大し十分な冗長化が困難となり、システム可用性が低下する一方で、1つのソフトウェアに少数のサービスを内包する場合、サーバ可用性の影響が顕著となりシステム可用性が低下したりと、サービスの実装形態やソフトウェア数によってサーバ可用性の影響が変化するためである。

3.2 サービス可用性の変化による影響

続いて、サービス可用性の変化がこれらの要素に与える影響を検討した。図6に示す9つのグラフは、横軸にシステム可用性（範囲は0.6～1.0）、縦軸にシステム可用性の累積密度関数を表している。全グラフにおいて、サーバ可用性は0.99に固定し、サービス可用性に関しては、80%のサービスは0.99とし、他の20%について0.99、0.95、0.9と変化させた。ソフトウェアの冗長化度合いは最大4とし、95%以上のリソース利用率を示すパターンを評価対象とした。

このページには注釈はありません

図5にそれぞれのサービス可用性の組み合わせにおけるソフトウェア数、サービス実装形態、冗長化度合いの3つの要素とグラフの対応を示す。緑色の線がサービスの実装形態による影響を表し、黄色色の線が冗長化度合いによる影響を表し、青色の線がソフトウェア数による影響を表す。また、線の種類がシステムにおけるサービス可用性の組み合わせの違いを表しており、実線は全て0.99、破線は80%のサービスが0.99、20%のサービスが0.95、一点鎖線は80%のサービスが0.99、20%のサービスが0.9の場合である。

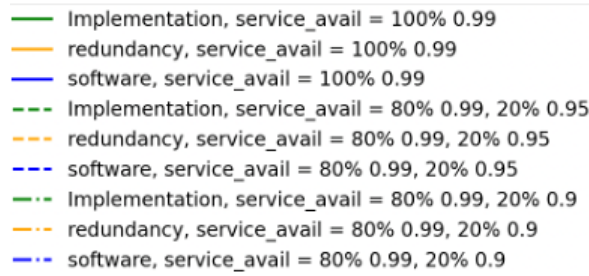


Figure 5: それぞれの要素とラベルの対応

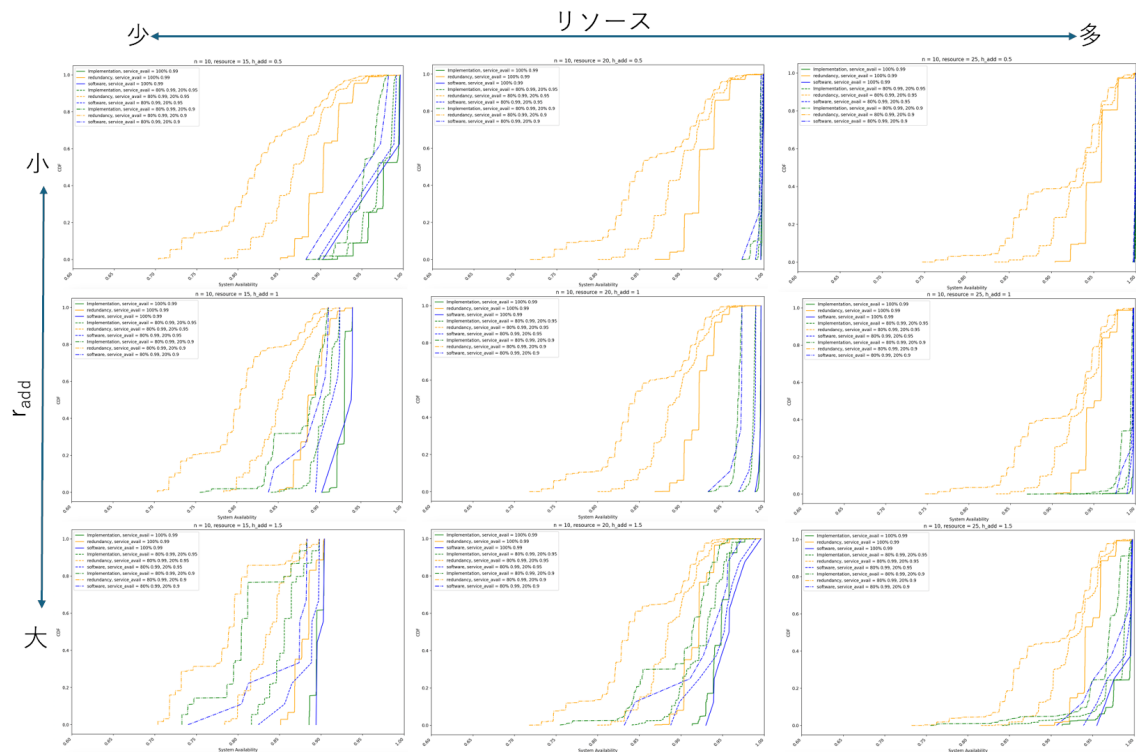


Figure 6: サービス可用性変化時の比較

分析結果とその考察

- 一部のサービス可用性の低下に伴い、全要素の影響が増大する。これは、可用性の低いサービスの実装方法や、そのサービスを含むソフトウェアの冗長化の仕方によって、システムの可用性が大きく変化するためである。

このページには注釈はありません

- リソースが少なくなるほど、ソフトウェア数・サービス実装形態・冗長化度合いの影響が増大する。これは、リソース制約により、可用性が低下する組み合わせの十分な冗長化や、高可用性を実現するサービス実装形態の実現が困難になるためである。
- r_{add} の増大に伴い、ソフトウェア数とサービス実装形態の影響が大きくなる。可用性の低いサービスには高度な冗長化が必要であるが、サービス実装形態やソフトウェア数によっては、可用性の低いサービスが他のサービスと同一ソフトウェアに実装され、冗長化に必要なリソースが増大し、十分な冗長化が実現できない組み合わせが生じるためである。

本解析におけるサービス可用性とサーバ可用性それぞれの変化による影響をみると、一部のサービス可用性の変化よりも、サーバ可用性の変化の方が、それぞれの要素がシステム可用性に与える影響が大きくなることがわかった。

また、3つの解析的評価を通して、冗長化度合いがシステム可用性に大きな影響を与えることが明らかになった。リソースが少なかったり、 r_{add} が小さい場合には冗長化度合いだけでなく、サービス実装形態やソフトウェア数もシステム可用性に影響を与える。

このページには注釈はありません