

# 3. システム構成要素

[http://cobayasi.com/koza/ap/3\\_systemkose.pdf](http://cobayasi.com/koza/ap/3_systemkose.pdf)

3.1 システム構成技術

3.2 システムの性能・信頼性

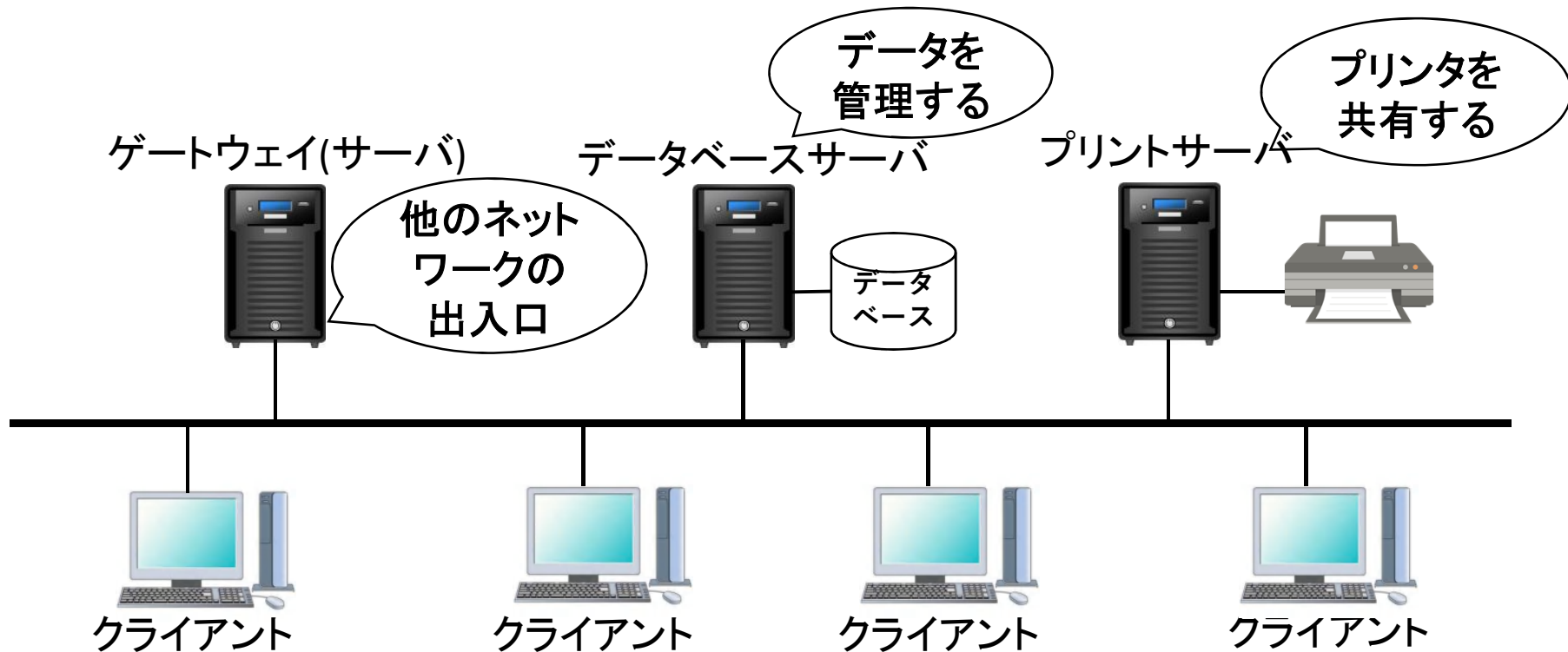
# 3.1 システム構成技術

3.1.1 クライアントサーバシステム

3.1.2 システムの構成方式

3.1.3 処理形態

# 3.1.1 クライアントサーバシステム (Client Server System : CSS)



- 処理を依頼する **クライアント(Client)**
- 処理を提供する **サーバ(Server)**

## ●CSSの特徴(長所と短所)

### [長所]

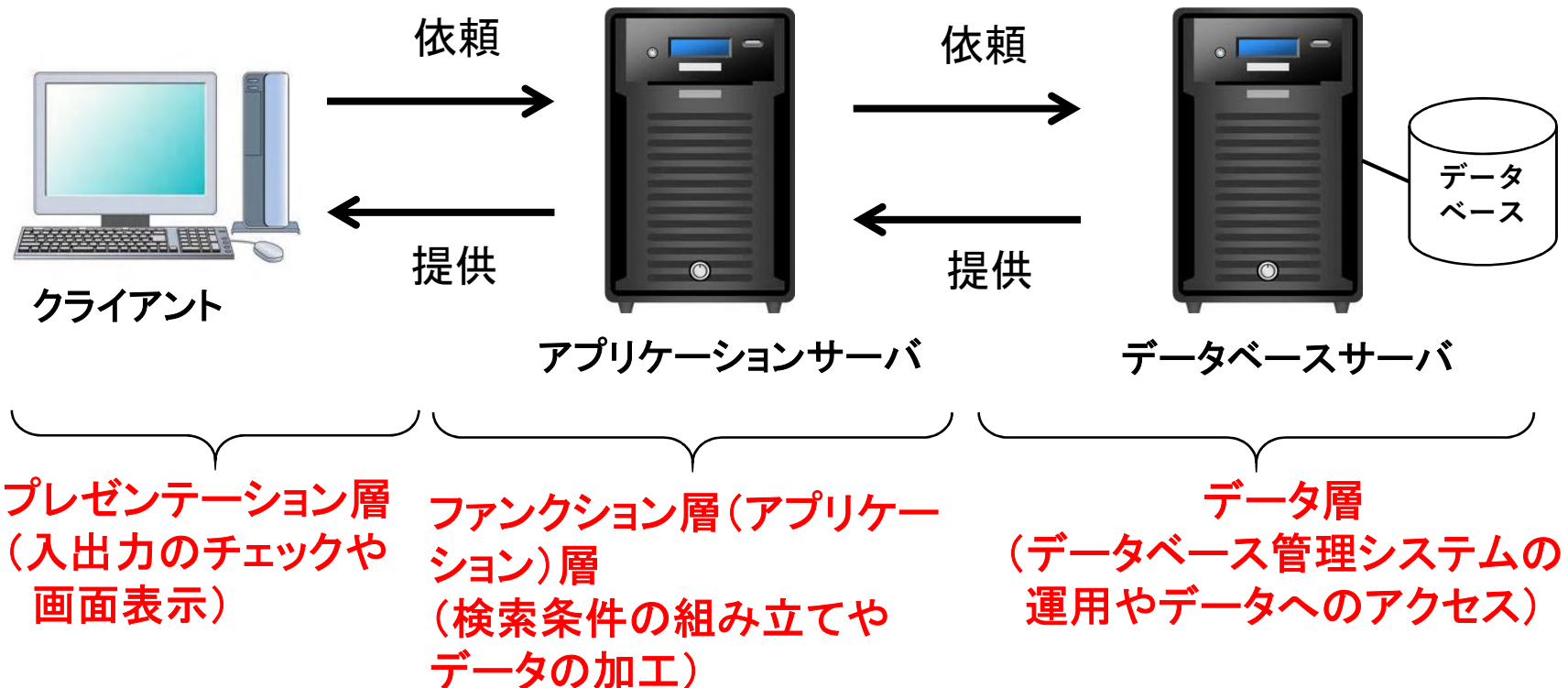
- 機能に対応したハードウェアやソフトウェアの選択が可能
- システムの機能や性能に対して柔軟な拡張性がある
- アプリケーション開発の生産性や保守性が向上
- GUIを利用した操作性の高いアプリケーション開発が可能

### [短所]

- 分散化によってシステムのパフォーマンスが低下
- ネットワークの負荷の予想が困難
- ハードウェアの管理作業や保守作業の量が増加
- 異なるメーカーの製品であると障害発生時の原因の特定が困難
- データ共有にともなうセキュリティの維持が困難

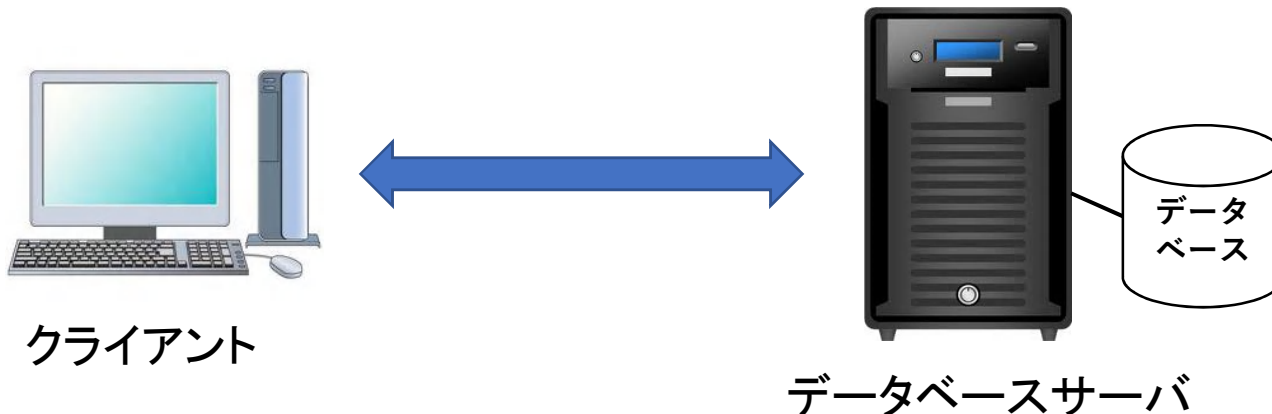
# ● 3層アーキテクチャ (3層クライアントサーバシステム)

クライアントサーバシステムの機能を、3つの役割(プレゼンテーション層、ファンクション層またはアプリケーション層、データ層)に分けて動作させる



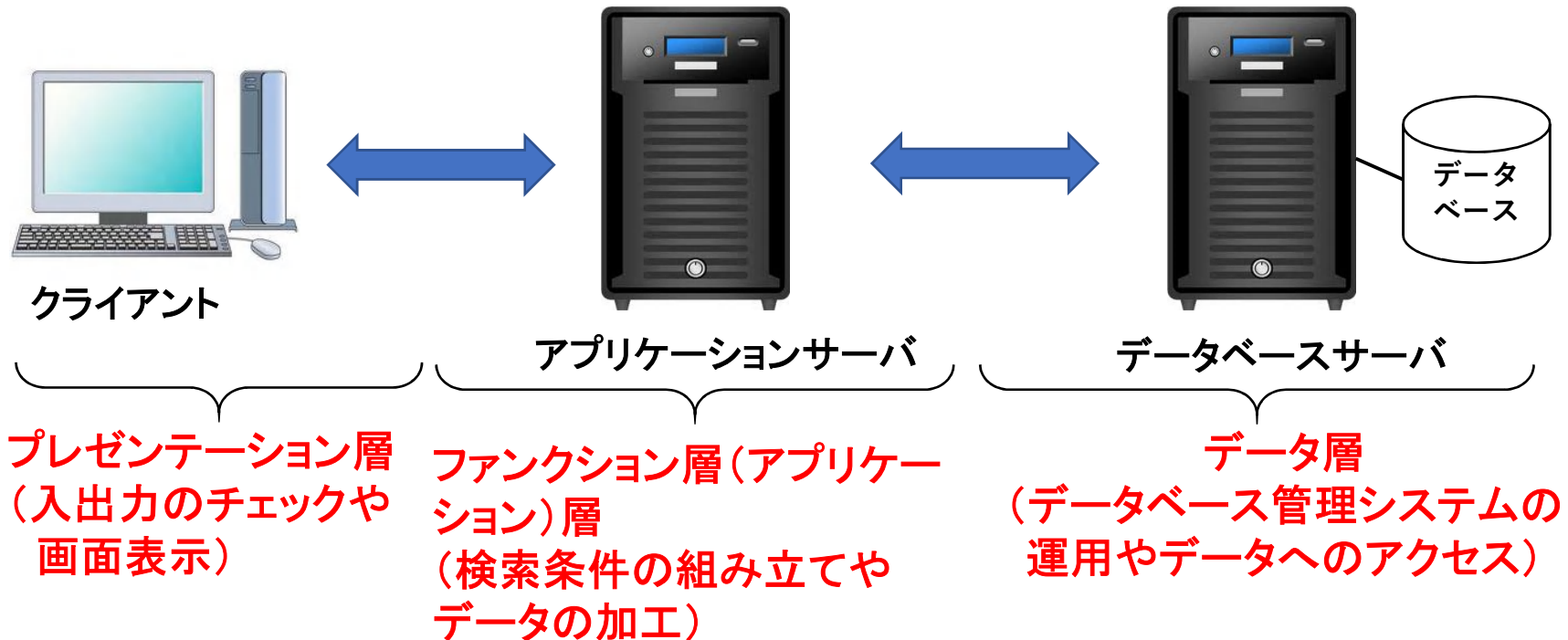
## 通常のクライアントサーバシステムでは、

- アプリケーションがクライアントに存在するため、処理手順に変更が生じたら、クライアントに修正が必要になる
- 処理に必要なデータが、いちいちネットワーク上を流れるので、ネットワークの帯域を圧迫する
- 処理時間やネットワークの速度が遅くなると、データベースをロックする時間が長くなる



### 3層クライアントサーバシステムでは、

- クライアントは、GUI操作を行うだけなので、シンクライアントが使用できる
- ネットワーク上には、処理結果だけが流れるので、データ量を軽減でき、低速度のネットワークでもよい
- 処理手順が変更になっても、サーバだけに修正すればよい
- サーバだけで処理が完了するので、データベースをロックする時間が少なくなる



## 3.1.2 システム構成方式

コンピュータシステムの信頼性や稼働率を向上するための考え方

### フォールト・トレランス

全ての機器（CPUを含む）を多重化する。冗長構成をとることで、信頼性を高める。

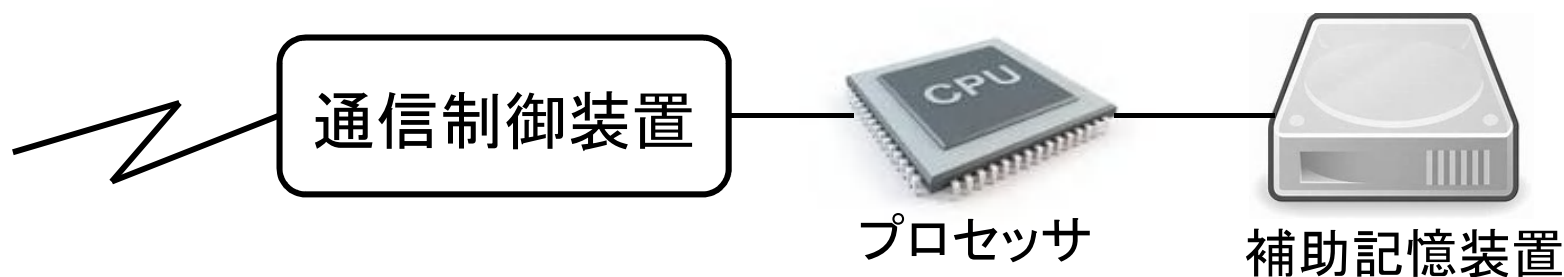
### フォールト・アポイダンス (フォールトイントレランス)

システム構成要素に、信頼性の高いものを使い、障害の発生を抑える



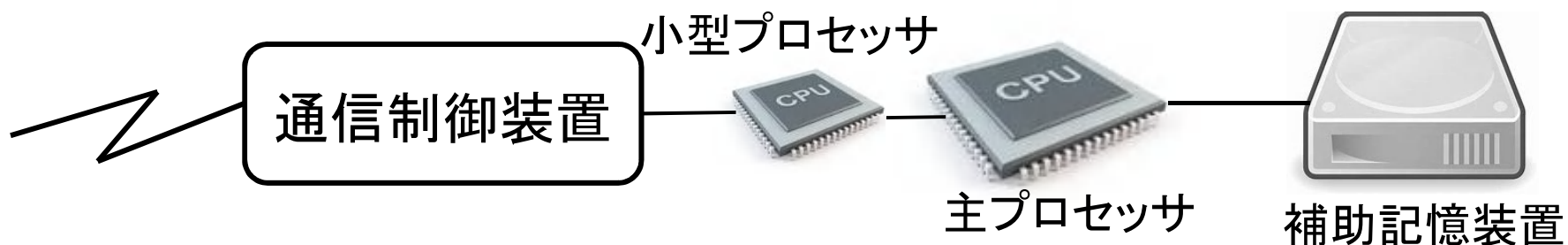
## ● シンプレックスシステム

プロセッサ1台で構成し、経済的ではあるが、安全性、信頼性、処理能力では他の構成には劣る。また、停止しないと、保守や修理ができない、欠点がある。



## ● タンデムシステム

プロセッサ2台を直列に接続した構成で、機能分散や負荷分散を目的としたシステム。信頼性や稼働率の向上を目的でない。



## ●デュプレックスシステム

正規システムでオンライン処理し、待機システムでバッチ処理する。正規システムで障害が発生した場合には、待機システムでオンライン処理を行う。



### ホットスタンバイ方式

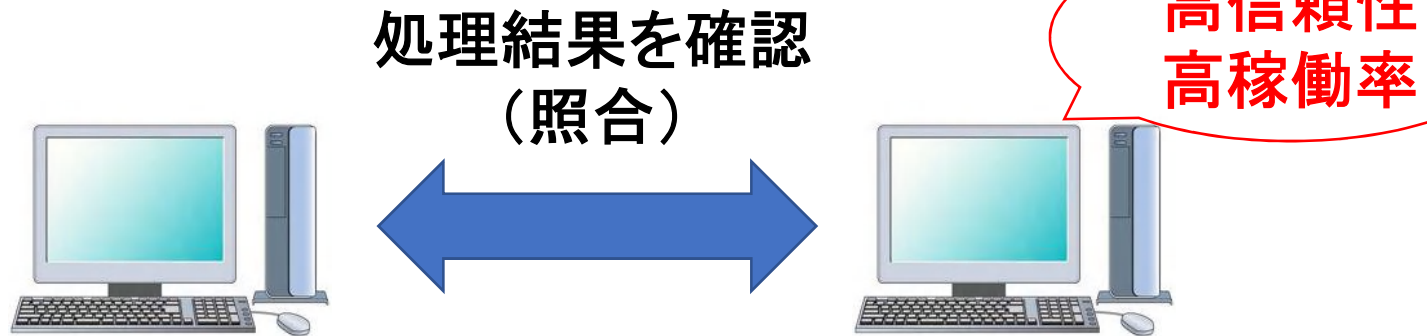
待機システムは常に起動して置き、障害発生時に素早く切り替える

### コールドスタンバイ方式

待機システムは電源断や他のシステムを稼働させ、障害発生時に素早く切り替える

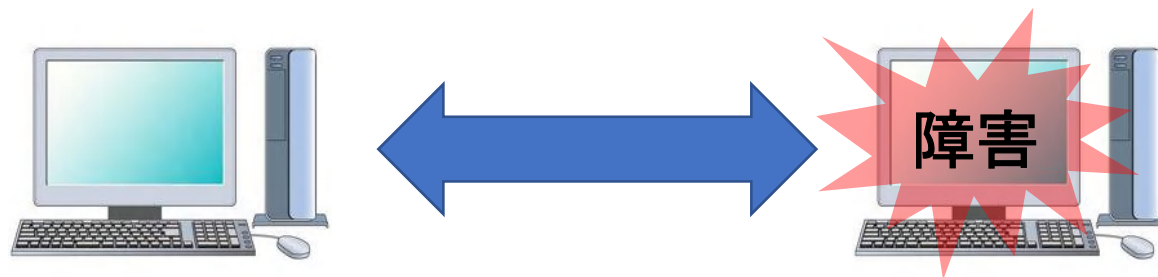
## ●デュアルシステム

同じシステムで、同時に同じ処理(同期運転)をする



### 障害が発生した場合

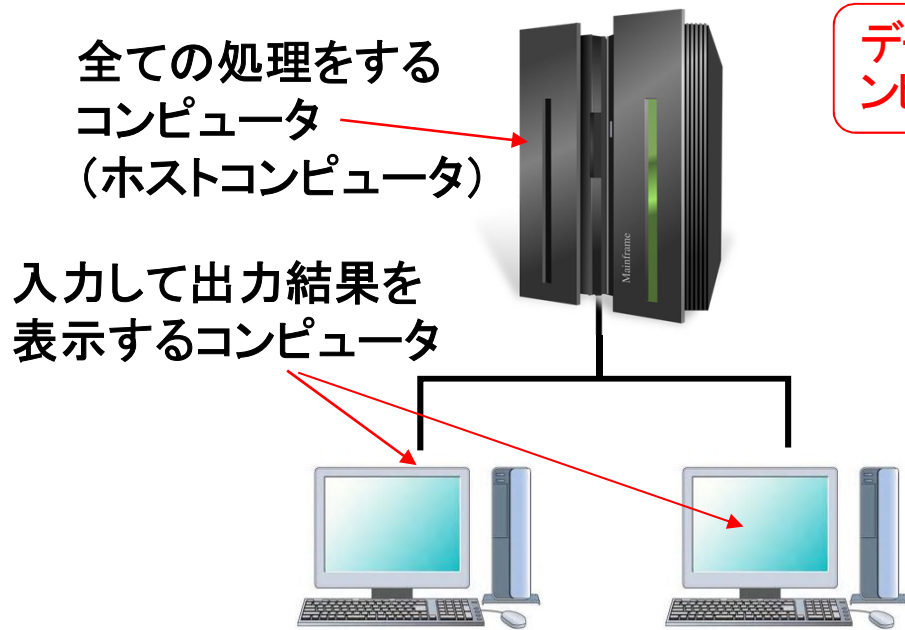
障害発生のをシステムを切り離して(停止して)、正常なシステムで処理を続行する



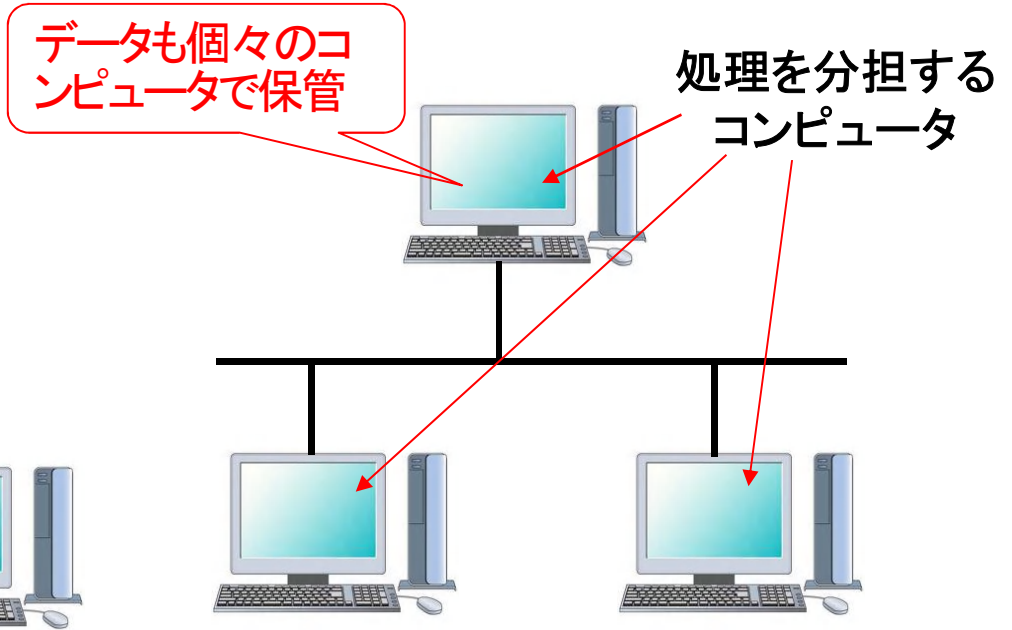
# 3.1.3 処理形態

集中処理

分散処理



ホストコンピュータが集中的に処理を受け持ち、他のコンピュータはそれにぶら下がる



複数のコンピュータが処理を分担して、負荷を分散する

## ■ 集中処理の長所と短所

<b>長所</b>	<ul style="list-style-type: none"><li>• スケールメリット(規模が小さく)</li><li>• 要員、ハードウェア、ソフトウェアに無駄がないので運用面で経済的</li><li>• システム構成の標準化が容易</li><li>• セキュリティの確保が容易</li></ul>
<b>短所</b>	<ul style="list-style-type: none"><li>• バックログの滞積(環境の変化に柔軟な対応が可能)</li><li>• オーバーヘッドが大きくホストに負荷が集中</li><li>• ホストに障害が発生するとシステム全体が停止</li><li>• システムが複雑になり使いにくいことがある</li></ul>

## ■ 集中処理の処理方式

企業の売上情報  
の集計など

**バッチ処理**  
(要求がある程度まと  
まってから処理する)

センタバッチ処理  
リモートバッチ処理  
集信処理(データ収集システム)  
配信処理  
集配信処理(集信処理+配信処理)

銀行の  
ATMなど

**リアルタイム処理**  
(要求されたと同時に  
処理する)

オンラインランザクション処理  
メッセージ交換  
問合せ応答(会話型処理)  
TSS  
リアルタイム制御(プロセス制御)

## ●分散処理の分類

分散の内容 \ 構成	水平分散	垂直分散
機能分散	水平機能分散	垂直機能分散
負荷分散	水平負荷分散	

### ■ 水平分散

機能や負荷に対して上下の関係がなく、**全コンピュータが対等に処理**を行う形態

- ✓ 水平負荷分散: 処理要求が発生すると空いているコンピュータが処理を担当する
- ✓ 水平機能分散: 業務別やアプリケーションごとに処理するコンピュータを分ける

## ■ 垂直分散

処理機能や機器に階層を持たせたシステム形態

- ✓ 垂直機能分散: 個々の利用者が、共同で利用するコンピュータなどの間で機能を分散する処理形態

クライアントサーバ  
(サーバの処理機能を複数のクライアントに対して提供している)



## ■ 分散処理システムの長所と短所

<b>長所</b>	<ul style="list-style-type: none"><li>• 管理組織と機器の対応がとれるので、管理責任者は明確である</li><li>• 災害や障害の発生を局所化できる</li><li>• マルチベンダ製品の導入でコストパフォーマンスが向上する</li><li>• アプリケーションや機器の追加が容易である(保守性が高い)</li></ul>
<b>短所</b>	<ul style="list-style-type: none"><li>• システムの実体がとらえにくい</li><li>• システム全体の性能はネットワークの性能に左右される</li><li>• データの不整合が発生しやすい</li><li>• 外部からの侵入に弱い(ウイルスなどに感染しやすい)</li></ul>

## ■ 過去問題1

3層クライアントサーバシステムの説明のうち、適切なものはどれか？

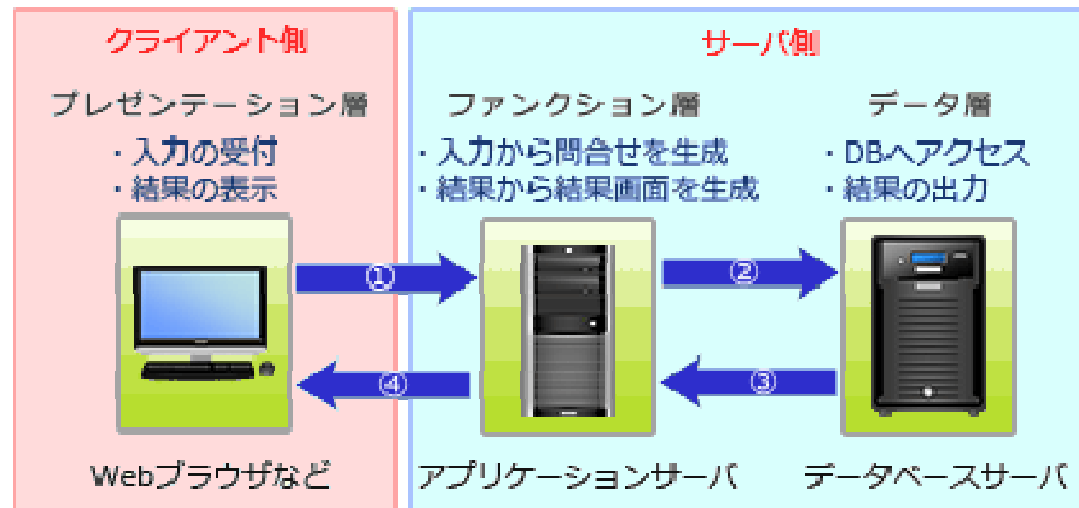
ア:システムを機能的に、Webサーバ、ファイアウォール、クライアントの3階層に分けたシステムである

イ:システムを機能的に、アプリケーション、通信、データベースの3階層に分けたシステムである

ウ:システムを物理的に、メインフレーム、サーバ、クライアントの3階層に分けたシステムである

エ:システムを論理的に、プレゼンテーション、ファンクション、データベースの3階層に分けたシステムである

3層クライアントサーバシステムは、ユーザの入出力を担当する「プレゼンテーション層」、業務処理に依存するデータ加工を行う「ファンクション層※」、データベース処理を行う「データ層」の3層に分離したモデル



※ファンクション層は、アプリケーション層、トランザクション層、ビジネスロジック層などの名前でも呼ばれる

## 3.2 システムの性能・信頼性

3.2.1 性能指標

3.2.2 信頼性

3.2.3 稼働率と故障率

# 3.2.1 性能指標

コンピュータシステムの性能の評価指標

- CPUの処理能力
- バッチ処理の処理能力
- オンラインシステムの処理能力

## ● CPUの処理能力

- MIP(Million Instruction Per Second)  
1秒間に実行可能な命令数( $10^6$ 単位で表示)
- FLOPS(Floating Point Operation Per Second)  
1秒間に実行可能な浮動小数点命令数
- 命令ミックス  
よく使用する命令の使用頻度と実行時間から求める1命令当たりの平均実行時間
- SPEC(Standard Performance Evaluation Corporation)  
UNIXで使用する総合的な処理能力評価のベンチマーク

- バッチ処理の処理能力

- スループット(”処理能力“と呼ぶ)  
単位時間当たりの処理が可能なジョブ数やデータ量
- TAT(Turn Around Time)  
ジョブを入力してから結果が出力するまでの時間

- オンラインシステムの処理能力

- 応答時間(“レスポンスタイム”と呼ぶ)  
入力が完了してから、結果の出力が開始するまでの時間
- TPS(Transaction Per Second)  
1秒間に処理可能なトランザクション件数
- TPC(Transaction Processing performance Council)  
オンライントランザクション処理の性能評価のベンチマーク

- 命令ミックスとMIPSの計算方法

命令の種類	実行時間	出現頻度
演算	80ns	20%
移動	20ns	50%
比較	40ns	20%
その他	60ns	10%

- 命令ミックス(平均命令実行時間)

$$= \sum \text{命令実行時間} \times \text{出現頻度}$$

$$= 80 \times 0.2 + 20 \times 0.5 + 40 \times 0.2 + 60 \times 0.1$$

$$= 40ns$$

演算命令

移動命令

比較命令

その他命令

- $MIPS = 1 \div (40 \times 10^{-9}) \div 10^6 = 25MIPS$

## ● ベンチマークテスト

コンピュータシステム全体を性能評価するためのプログラムを実行し、数値化して表示する。

- SPEC(Standard Performance Evaluation Corporation)

コンピュータの公平で意味のあるベンチマークを作成することを目指して設立された非営利団体(システム性能評価協会)。

SPECのベンチマークは、コンピュータシステムの性能評価に今日広く使われていて、その測定結果はSPECのウェブサイト上で公表されている。

指標名	意味
SPECmark	SPECintとSPECfpを統合して評価する
SPECint	整数演算を評価する
SPECfp	浮動小数点演算を評価する



- TPC(Transaction Processing performance Council)  
コンピュータ関連の有力企業が集まり、データベースのトランザクションに関して実際のシステムに近い性能指標を作成する目的で設立された非営利団体(トランザクション処理性能評議会)。  
TCPベンチマークは、システム全体の性能評価ができ、性能の値とコストパフォーマンスで表す。

指標名	意味
TPC-C	複雑な受注業務をモデル化し、実際の企業内システムに似たOLTPを評価する
TPC-H	大型基幹業務向け意思決定支援環境をシミュレートして評価する
TPC-R	ビジネスレポートと意思決定支援環境をシミュレートして性能を評価する
TPC-W	WebベースのOLTPの性能を評価する

## ● TPS(Transaction Per Second)の算出

トランザクション処理の処理能力を表すTPSは、以下のように算出する

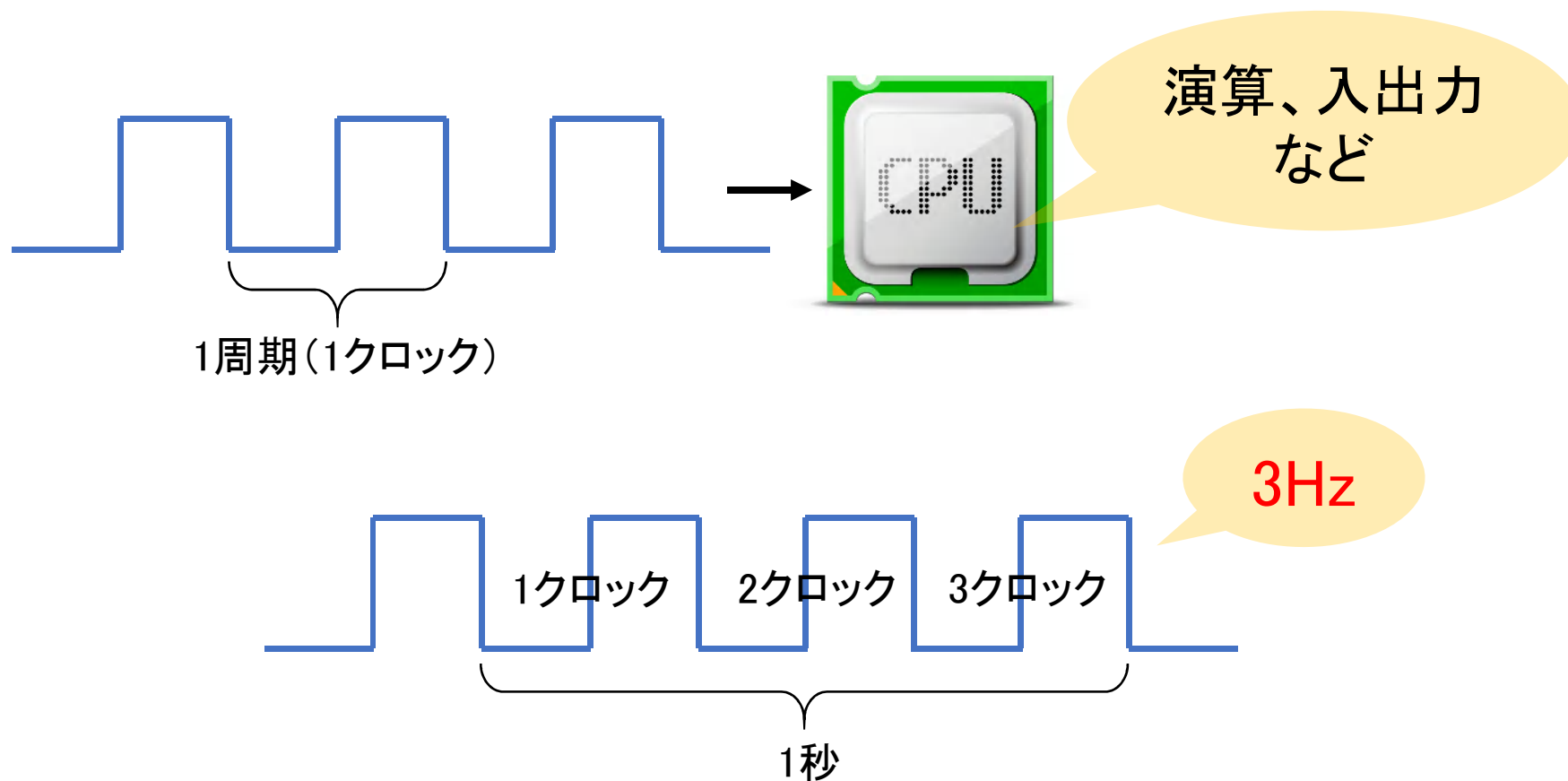
**TPS(件/秒)=MIPS値÷トランザクションの処理ステップ数**

2MIPSのコンピュータで、トランザクションの実行ステップが平均10万命令、CPUの使用率が80%のとき、

$$\text{TPS} = 2 \times 10^6 \times 0.8 \div (10 \times 10^4) = 16 \text{ (件/秒)}$$

## ●クロック(周波数)

CPUの内部回路を動かしている基準信号(クロック)の速度  
1秒間に刻むクロック数で表す → 単位[Hz](ヘルツ)



## ●モニタリング

動作中のコンピュータシステムの実行結果を、ソフトウェアやハードウェアを使って収集し、その収集結果からシステムの性能や稼働状況を評価する



- ハードウェアモニタリングの監視項目
  - ✓ 命令実行回数と所要時間
  - ✓ 命令種別の実行頻度
  - ✓ バッファやキャッシュメモリのヒット率
  - ✓ 主記憶のアクセス分布
  - ✓ マルチプロセッサの競合状態
- ソフトウェアモニタリングの監視項目
  - ✓ スーパーバイザモードの時間の割合
  - ✓ 装置の負荷状況
  - ✓ メモリの使用状況
  - ✓ 応答時間
  - ✓ 仮想記憶のページング状況
  - ✓ ジョブステップごとの資源の利用状況
  - ✓ ジョブの多重度と待ち状況
  - ✓ 事象トレース(タスク切替、割込み)
- ハードウェアモニタリングとハードウェアモニタリングの共通監視項目
  - ✓ プロセッサやチャネル、入出力装置の利用率
  - ✓ 磁気ディスクのアクセス分布

## 3.2.2 信頼性

### コンピュータシステムの信頼性

- 停止しない
- 決められた時間内で完全に動作する
- データに誤りが発生しない

これらが満た  
されること

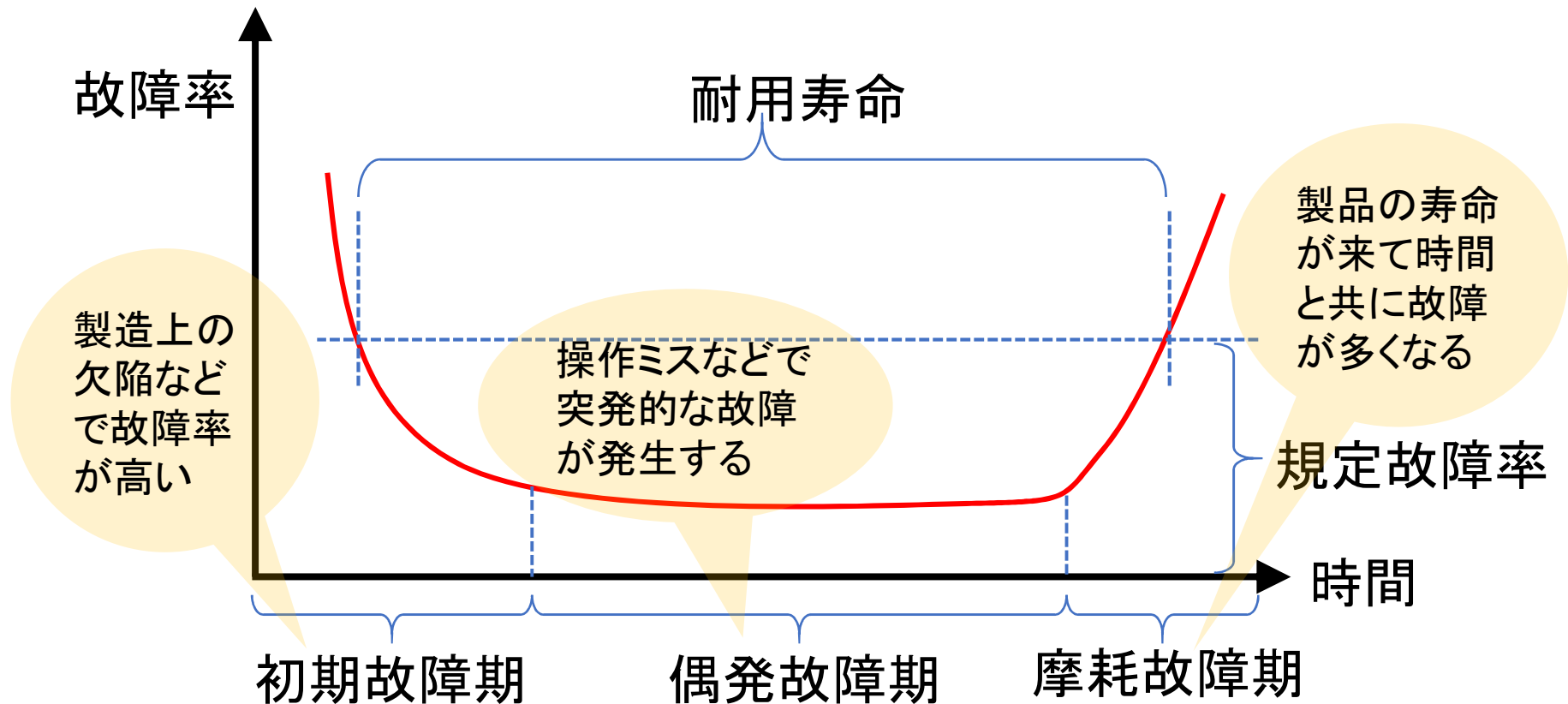
# ● 信頼性の評価指標 ( RASIS )

コンピュータシステムを安全に使用できるように、システムが備えるべき性質

- ①信頼性( **R**eliability ) システムが正常に動作していること  
指標⇒**平均故障率** MTBF( Mean Time Between Failures )
- ②可用性( **A**vailability ) 必要な時にいつでも利用できること  
指標⇒**稼働率**
- ③保守性( **S**erviceability ) 故障などの障害時に、どれだけ早く発見や修復ができること  
指標⇒**平均復旧率** MTTR( Mean Time To Repair )
- ④完全性( **I**ntegrity ) 誤動作がなく、データが正しく保たれていること
- ⑤機密性( **S**ecurity ) 不正利用に対してシステムが守られていること

# ● バスタブ曲線

機械や装置の故障率と使用期間の関係をグラフ化したもの





# ● 信頼性設計

機械や装置が故障しても動作を続けられるように設計する(フェールトレラント)

## フェールトレラントを実現する方法

- フェールセーフ **デュアルシステム**  
故障が発生した時に、安全性を確保するように動作させる
- フェールソフト **デュプレックスシステム**  
故障が発生した時に、システム全体が停止しないようにする
- フールプルーフ  
誤操作しても、故障しないようにする

## 3.2.3 稼働率と故障率

コンピュータシステムの可用性の1尺度である稼働率は、システムを導入してから、全動作時間内で、正常に動作している時間の割合

$$\text{稼働率} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

MTBF: 平均故障間隔

MTTR: 平均修理時間

# ● MTBFとMTTR

RASISのR(信頼性)の指標

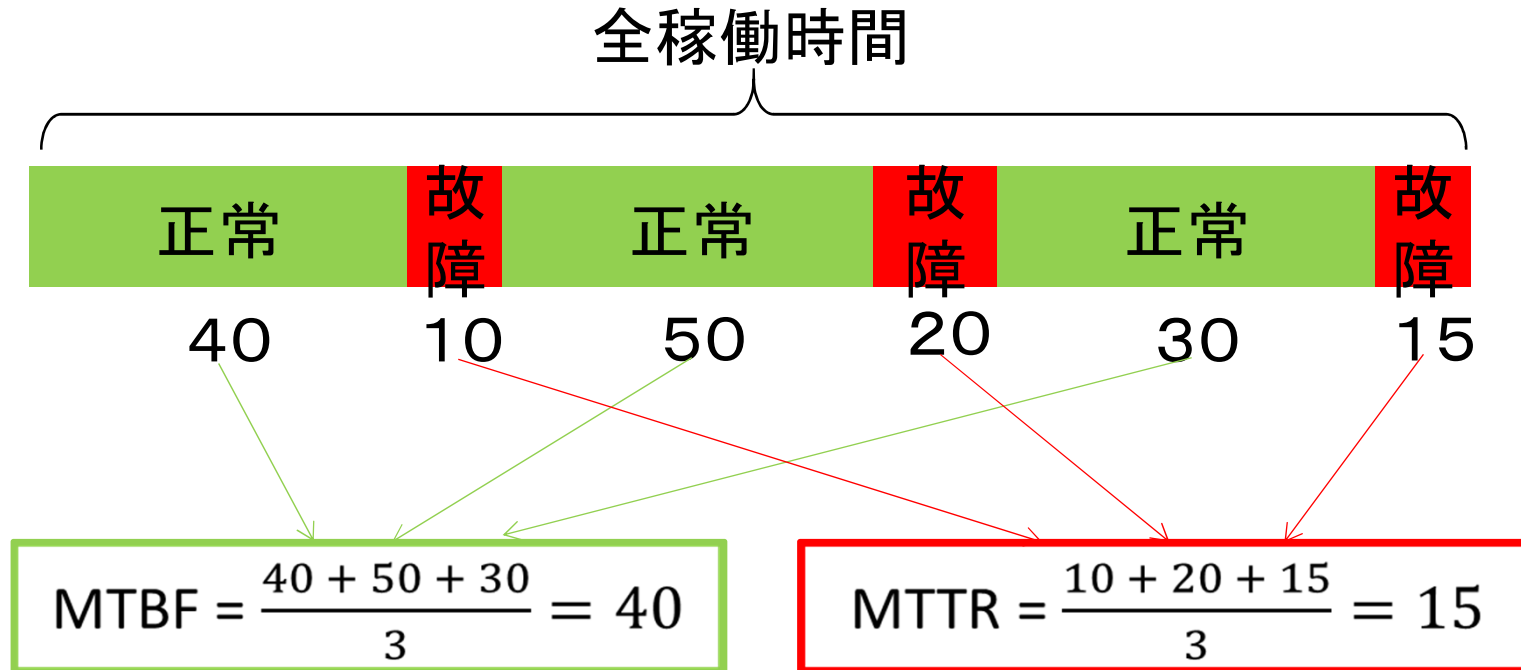
- 平均故障間隔( **MTBF** : **M**ean **T**ime **B**etween **F**ailures )

システムが正常に稼働している平均時間

RASISのS(保守性)の指標

- 平均修復(復旧)時間( **MTTR** : **M**ean **T**ime **T**o **R**epair )

システムが故障している平均時間



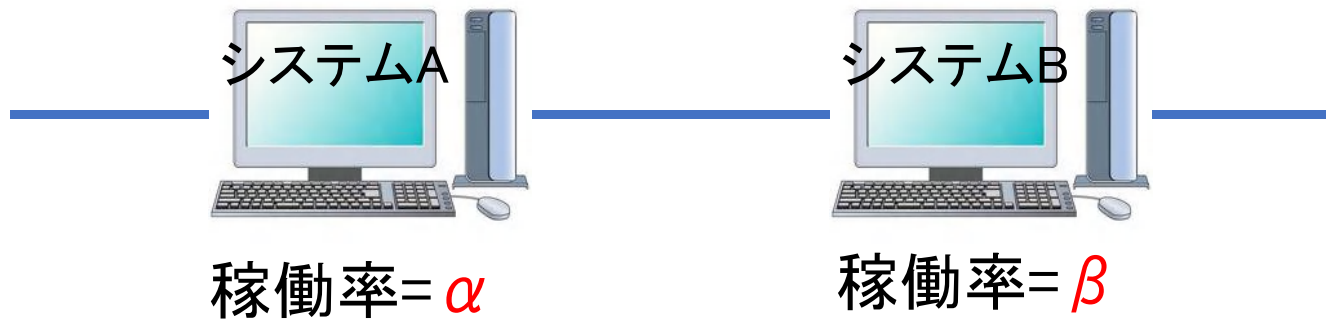
# ● 稼働率

コンピュータシステムを導入してから、全動作時間内で、正常に動作している時間の割合

- 計算式は、稼働率 =  $\frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$
- RASISのA(可用性)の指標
- 稼働率が、1に近いほど可用性(A)が高い  
(全く故障しないときは、稼働率=1)

- 直列接続の稼働率

どちらか1つが故障すると、システム全体は稼働しない



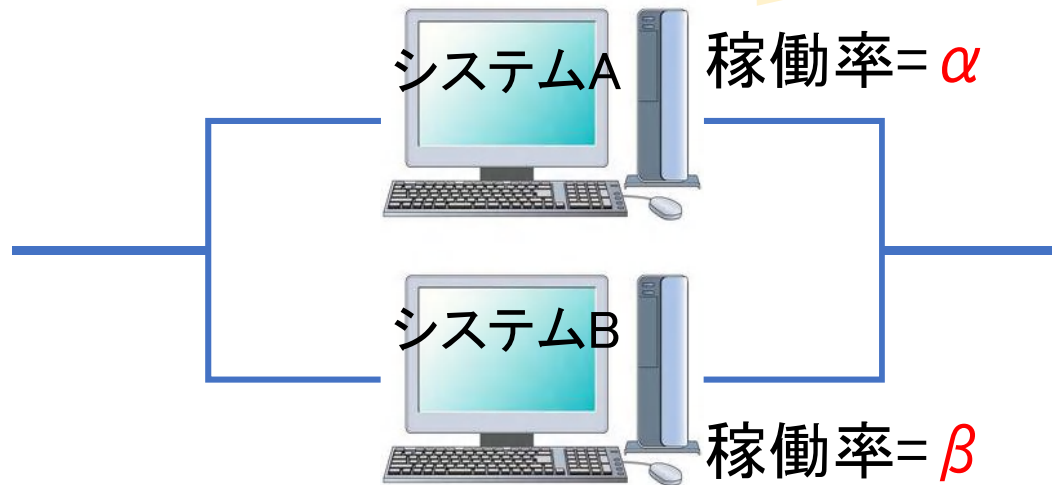
全稼働率 = システムAの稼働率  $\alpha$  X システムBの稼働率  $\beta$

システムAの稼働率=0.8 システムBの稼働率=0.8 のとき

システム全体の稼働率=0.8 × 0.8=0.64=64%

- 並列接続の稼働率

どちらか1つが故障しても  
システム全体は稼働する



$$\text{全稼働率} = 1 - \underbrace{(1 - \text{システムAの稼働率 } \alpha)}_{\text{システムAの不稼働率}} \times \underbrace{(1 - \text{システムBの稼働率 } \beta)}_{\text{システムBの不稼働率}}$$

システムAの稼働率=0.8 システムBの稼働率=0.8 のとき

$$\text{システム全体の稼働率} = 1 - ((1 - 0.8) \times (1 - 0.8)) = 1 - (0.2 \times 0.2) = 0.96 = 96\%$$

# ● 故障率

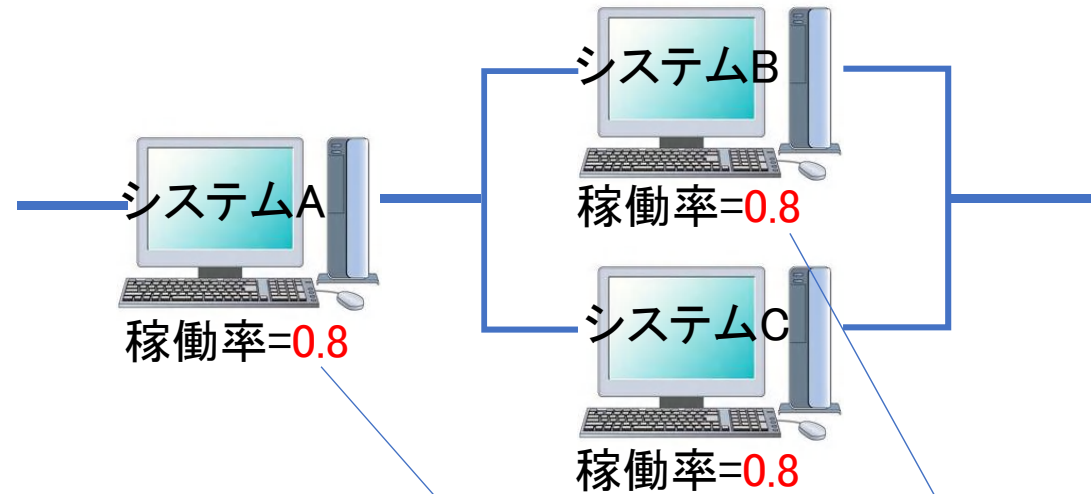
コンピュータシステムの単位時間当たりの故障回数

$$\text{故障率} = \frac{1}{\text{MTBF}}$$

平均故障間隔

- ✓ 故障率の単位は、FIT(Failure In Time)
- ✓ 1FIT =  $10^{-9}$ 回故障する
- ✓ システムAの故障率 $\lambda_1$ ,システムBの故障率 $\lambda_2$ のとき、システム全体の故障率 $\lambda$ は、 $\lambda = \lambda_1 + \lambda_2$

# ● 稼働率の計算



$$\begin{aligned}\text{システム全体の稼働率} &= 0.8 \times (1 - (1 - 0.8) \times (1 - 0.8)) \\ &= 0.8 \times (1 - 0.2 \times 0.2) \\ &= 0.8 \times 0.96 \\ &= 0.768 = 76.8\%\end{aligned}$$



## ■ 過去問題2

動作クロック周波数が700MHzのCPUで、命令の実行に必要なクロック数とその命令の出現率が表に示す値である場合、このCPUの性能は約何MIPSか？

命令の種類別	命令実行に必要なクロック数	出現率 (%)
レジスタ間演算	4	30
メモリ・レジスタ間演算	8	60
無条件分岐	10	10

ア:10

イ:50

ウ:70

**エ**:100

最初に、1命令を実行するのに必要な平均クロック数を求める

各命令を実行するのに必要なクロック数に、出現率を乗じたものを足し合わせて、

$$(4 \times 0.3) + (8 \times 0.6) + (10 \times 0.1) = 1.2 + 4.8 + 1.0 = 7.0$$

上の計算結果から、1命令に必要な平均クロック数は**7クロック**であるとわかる

次に、CPUの動作クロック数が700MHzなので、1秒間の命令実行可能回数を求める

$$700 \times 10^6 \div 7 = 100 \times 10^6 \text{回}$$

MIPSは1秒間に実行できる命令数を、百万単位で表す指標なので、このCPUの性能は**100MIPS**となる

## ■ 過去問題3

MTBFが $x$ 時間、MTTRが $y$ 時間のシステムがある。使用条件が変わったので、MTBF、MTTRがともに従来の1.5倍になった。新しい使用条件での稼働率はどうなるか？

ア： $x, y$ の値によって変化するが、従来の稼働率よりは大きい値になる

イ：従来の稼働率と同じ値である

ウ：従来の稼働率の1.5倍になる

エ：従来の稼働率の $2/3$ 倍になる

MTBFとMTTRが共に1.5倍になったとすると、

$$\begin{aligned} & 1.5\text{MTBF} / (1.5\text{MTBF} + 1.5\text{MTTR}) \\ &= 1.5\text{MTBF} / (1.5(\text{MTBF} + \text{MTTR})) \\ &= \text{MTBF} / (\text{MTBF} + \text{MTTR}) \end{aligned}$$

と式を変形できるので、従来のシステムの稼働率と同じになることがわかる