

応用情報技術者試験

第5章 システム構成技術

1. システムの構成

クライアント / サーバ

▶ **クライアントサーバシステム**とは、処理を、

- ・ **クライアントプロセス**：処理を依頼するプロセス
- ・ **サーバプロセス**：依頼された処理を実行し、結果を返すプロセス

という2種類に分け、クライアントとサーバが協調しながら処理を進める形態をいう。

▶ ただし、クライアントやサーバはプロセスを指す概念であり、マシンを指すものではない。したがって、1台のコンピュータがクライアントとサーバを兼ねることもできるし、複数のサーバの機能を1台のコンピュータ上に実現することも可能となる。

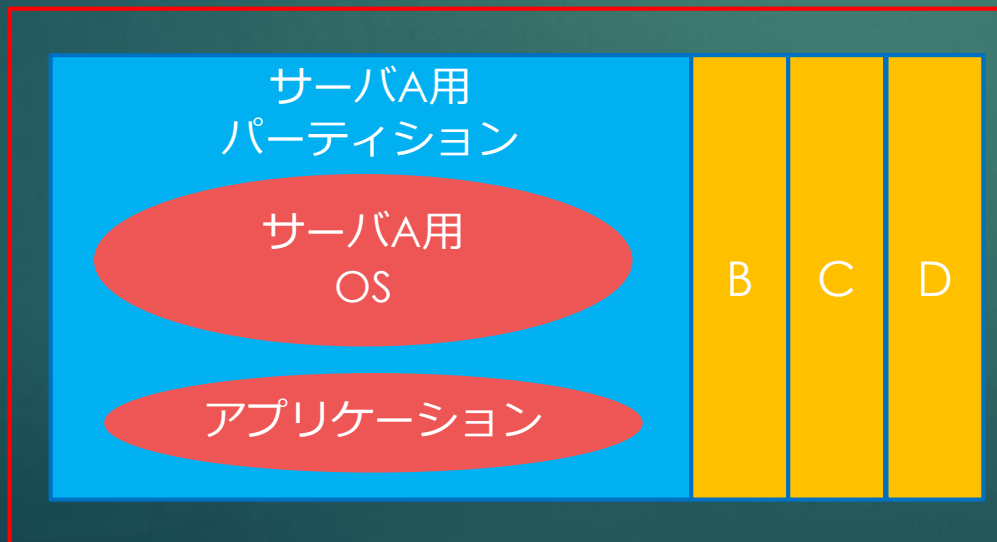
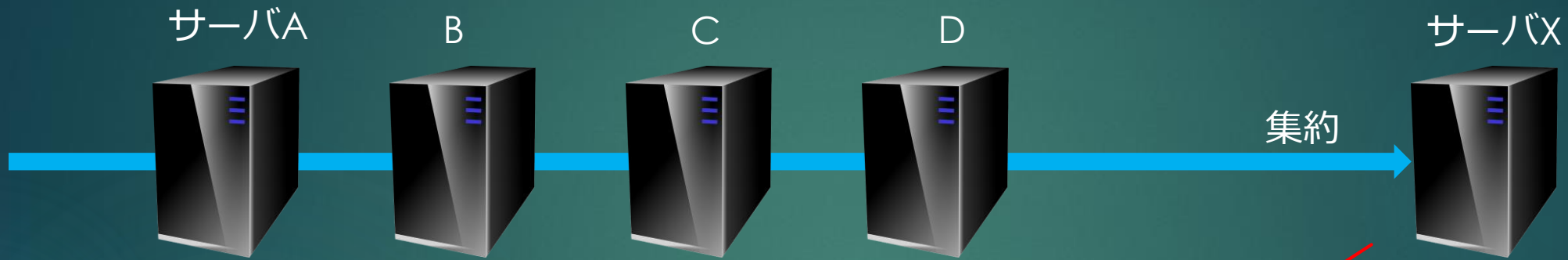
仮想化

▶ 仮想化

1台のコンピュータ上でOSを含めた複数の実行環境を提供する技術

- ▶ 仮想化によって、**1台のサーバマシン上で「OSの異なる複数のサーバプロセス」を実行する**ことができるようになる。見かけ上は、複数のサーバマシンが稼働しているように見える。
- ▶ コスト削減の他にも、既存システムに影響を与えることなく新システムを導入できるなどの利点がある。

仮想化



複数のサーバ機能を
1台のマシンに集約する

仮想化

▶ 複数サーバでの運用との比較による特徴

- ・ 物理的な資源の運用管理：簡単になる
- ・ 物理的な資源の利用率：高くなる
- ・ オーバヘッドによる負荷：たかくなる

▶ ライブマイグレーション

仮想サーバを停止することなく、OSやアプリケーションをほかの物理サーバへ移動すること

シンククライアント / リッチクライアント

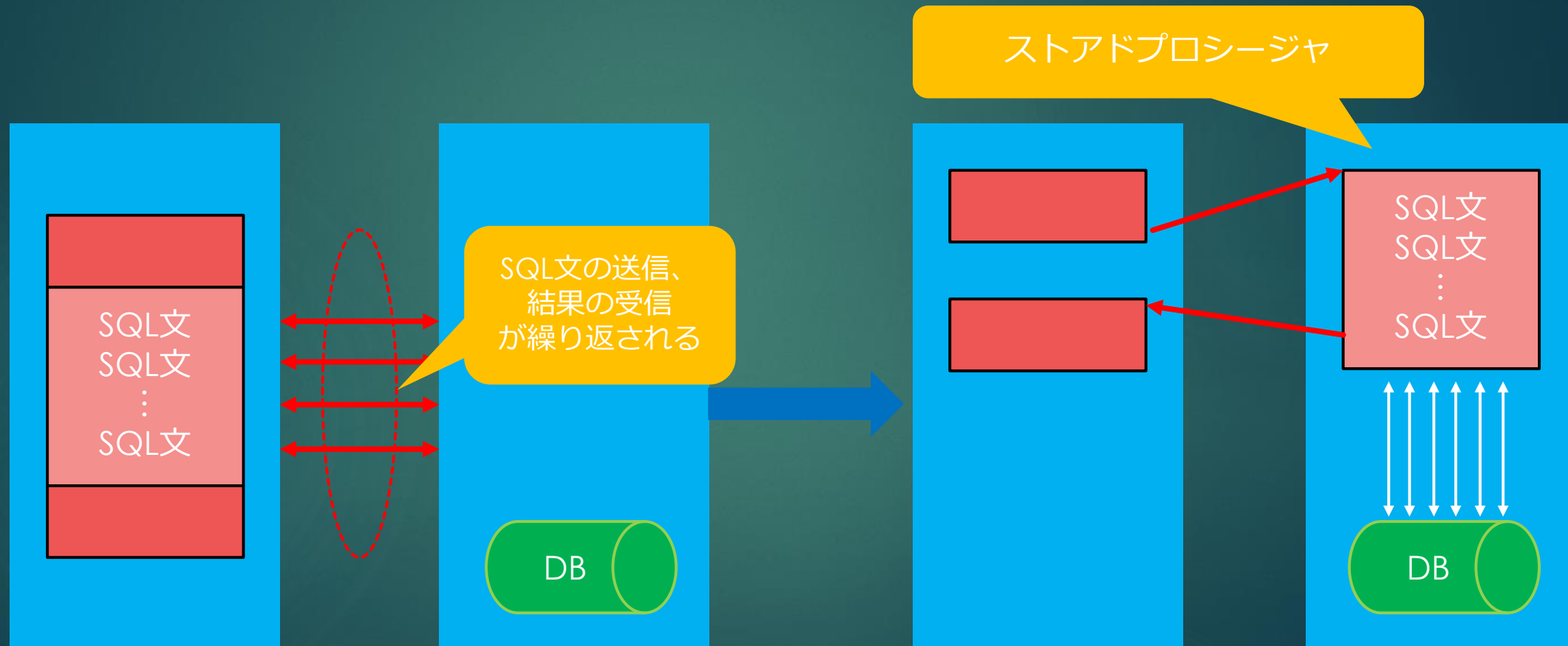
▶ クライアントの分類

シンククライアント	Webブラウザなどの 最小限の機能のみを搭載したクライアント形態 。主な処理はデータ管理はサーバ側で実行する
リッチクライアント	アプリケーションの実行環境を持たせたクライアント形態 。必要に応じてアプリケーションをダウンロードする。

ストアドプロシージャ

- ▶ アプリケーションをクライアントで実行するとき、クライアントとサーバ間で「SQL文の送信」と「実行結果の送信」が何度も繰り返されることになる。その結果、クライアントとサーバ間の通信量が増大し処理効率も低下する。
- ▶ これを避けるために、データベースにアクセスする定型的な処理をサーバ側に移動し、必要に応じてクライアントが呼び出すようにする。
- ▶ このときサーバ側に移した定型処理を**ストアドプロシージャ**とよぶ。また、ストアドプロシージャの呼出しにあたっては、クライアントから遠隔サーバ内の手続きを呼び出す**RPC(Remote Procedure Call)**という仕組みを用いる。
- ▶ ストアドプロシージャを使えば、**複数のSQL文からなる手続きを1回の呼出して実行**でき、クライアントとサーバ間の通信の負荷を軽減することができる。

ストアドプロシージャ



3層クライアント / サーバ

- ▶ **3層クライアント / サーバ**は従来はクライアント側にインストールされていたアプリケーションをサーバ側に移した構成のことをいう。

プレゼンテーション層	ユーザインタフェース(GUI)の提供
ファンクション層	アプリケーション機能（データ処理）の提供
データ層	DBへのアクセス

3層クライアント / サーバ

▶ Webショッピングシステム

プレゼンテーション層

クライアント

Web
ブラウザ

ウィンドウの表示、
データの入力など

ファンクション層

Webサーバ

Webアプリ
ケーション

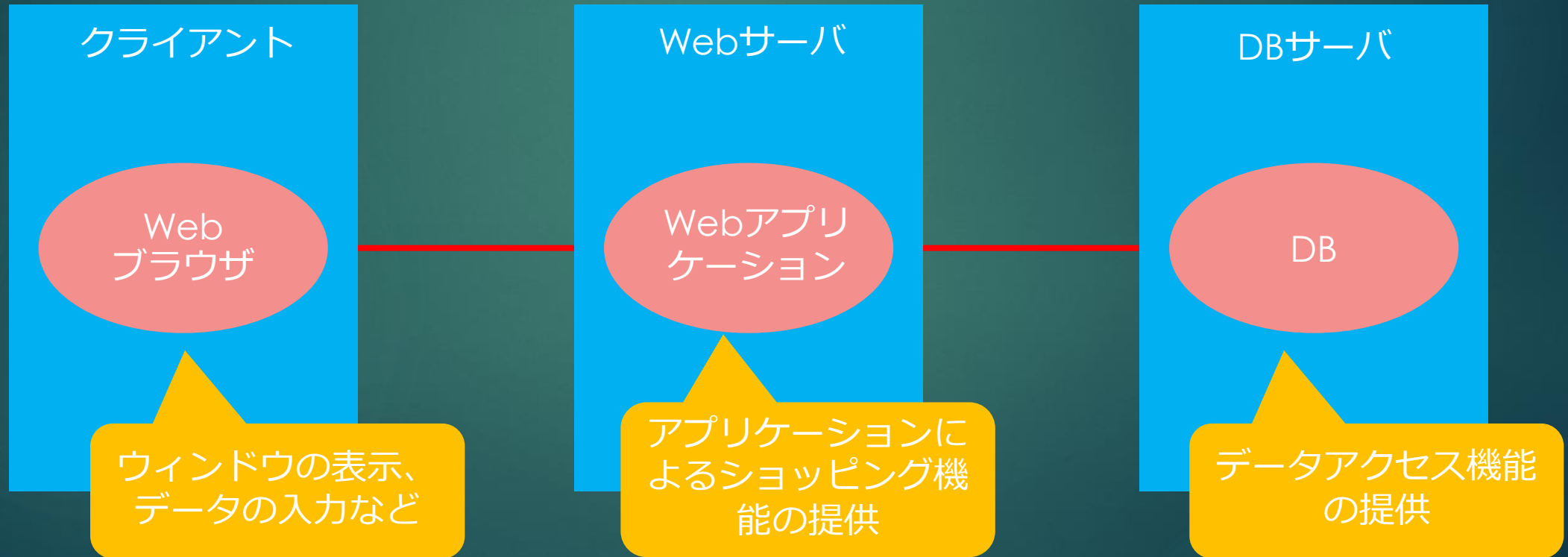
アプリケーションに
よるショッピング機
能の提供

データ層

DBサーバ

DB

データアクセス機能
の提供



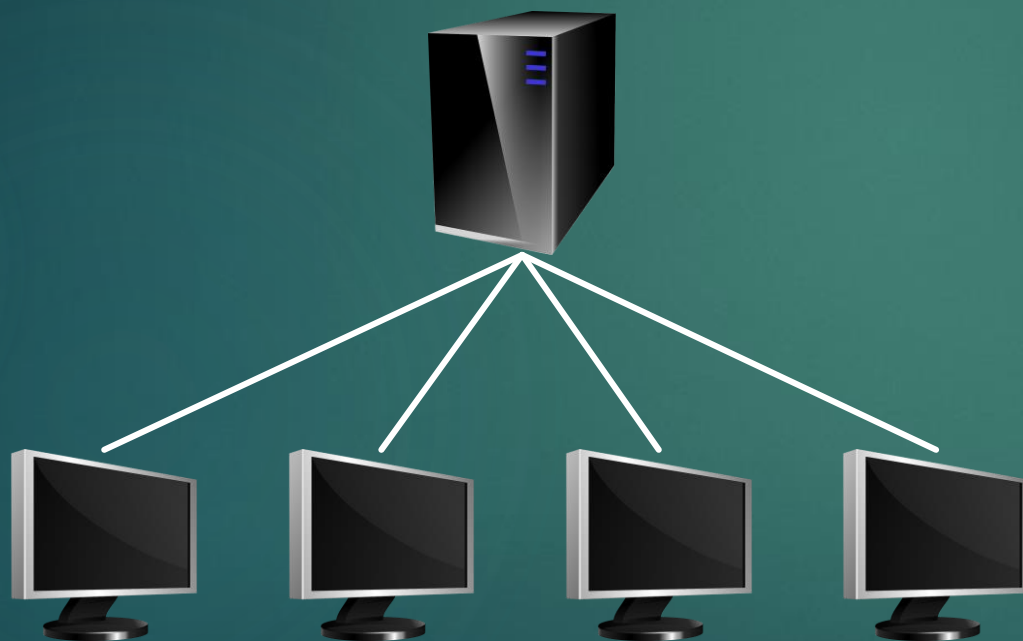
3層クライアント / サーバ

- ▶ 3層クライアント / サーバに対して、アプリケーションをクライアント側に実装する従来型の構成を、**2層クライアント / サーバ**とよぶことがある。2層クライアント / サーバと比較したとき、3層クライアント / サーバは次の利点をもつ。
 - ・ 各層の独立性が高く依存性も少ないので、**開発生産性の向上**が期待できる。
 - ・ クライアント側にはGUIを提供する機能のみが実装されるため、**ロジックの変更や追加に対応しやすい**

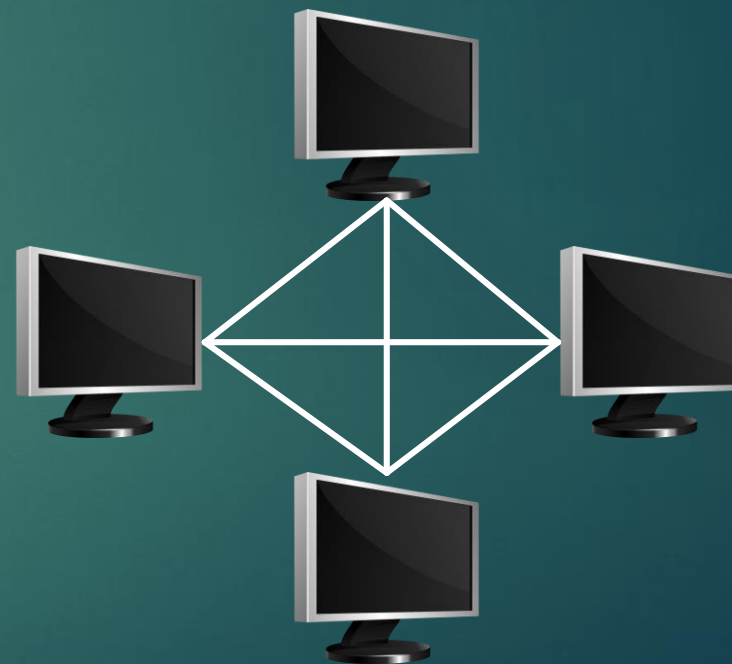
P2P (Peer to Peer)

- ▶ クライアント / サーバに対するシステム形態に**P2P**がある。

クライアント / サーバ



P2P



P2P (Peer to Peer)

- ▶ クライアント / サーバは明確な役割分担をもつが、**P2Pはお互いが対等な立場**で通信を行う。そのため、システムの規模が大きくなっても一度にアクセスが集中することがない。ファイル共有サービスやIP電話などが、P2Pの形態をもっている。
- ▶ P2Pにおける通信端末を**ピア**または**ノード**とよぶ。これらは**クライアントとサーバの機能を併せ持つ**ことから、**サーバント**と呼ばれることもある。

性能向上のためのシステム構成

- ▶ システムの構成要素を**多重化**し、並列に動作させることでシステム全体の性能を向上させることができる。多重化は、マルチプロセッサからクラスタリングやグリッドコンピューティングまで、広い範囲で用いられる。

- ▶ 多重化

同一要素を並列に接続することで、信頼性や性能を向上させること

マルチプロセッサシステム

- ▶ マルチプロセッサシステムは、主記憶を共有するか否かにより二つに分類することができる。
- ▶ マルチプロセッサシステムの分類

密結合型	複数のプロセッサが 主記憶を共有する マルチプロセッサシステム。単一のOSのもとで並列処理を行う
疎結合型	プロセッサごとに独立した主記憶とOSを持つ マルチプロセッサシステム。各プロセッサは通信回線で結ばれる

クラスタリング

▶ クラスタリング

同一の機能をネットワークで結合した複数のコンピュータで提供することで、信頼性や性能の向上を図る技術。スーパーコンピュータの構築から、Webサーバの負荷分散までさまざまな場面で利用されている。

▶ スケールアウト

サーバ増やして処理能力を上げること



グリッドコンピューティング

▶ グリッドコンピューティング

広い範囲に存在するコンピュータシステムを、インターネットなどの広域網を用いて接続し、大規模な処理を実行する方法

▶ PCから大型コンピュータまで、複数のコンピュータを結ぶことで、単体では対応できない規模の計算を実行することができる。

▶ HPC(High Performance Computing) (参考)

クラスタやグリッドコンピューティングなどの、並列化による高性能計算技術

信頼性向上のためのシステム構成

- ▶ システムの構成要素を二重化しておけば、一方に障害が発生したときに他方を切り替えることができるため、システムの信頼性は大幅に向上する。
- ▶ 信頼性向上に関する用語

フォールトトレランス	システムの構成要素に障害が生じても正常に稼働し続けること。 主要な構成要素を多重化 することで実現できる。
フォールトアボイダンス	障害の発生自体を抑えることで、信頼性を向上させること。品質管理などを通して、 個々のシステム要素の信頼性を高める ことで実現できる。
フェールオーバー (fail over)	多重化された要素の一方に障害が発生したとき、 自動的に他方に切り替える 機能
フェールバック (fail back)	フェールオーバーにより切り替えられた要素を、 障害復旧後に元の状態に戻す こと
フェールソフト (fail soft)	障害により機能が低下しても、完全に停止させずに処理を継続すること
フォールバック (fall back)	フェールソフトにより、機能を縮小した状態で運転すること(縮退運転)
フェールセーフ (fail safe)	障害が発生したとき、あらかじめ指定された安全な状態に誘導すること

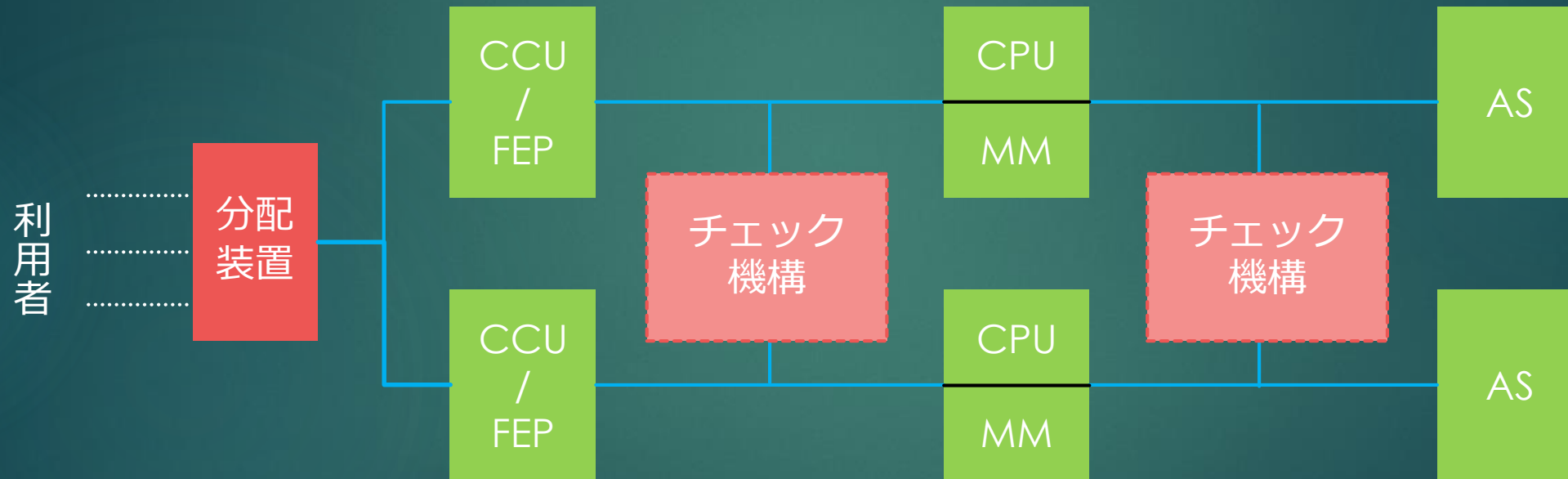
信頼性向上に関する用語

- ▶ フォールトトレランスは信頼性向上の根幹をなす考え方で、多重化による高信頼システムを**フォールトトレラントシステム**とよぶ。フォールトトレランスシステムにおいて障害が発生した場合は**フェールオーバー**が行われ、切り離された機器は復旧後に**フェールバック**される。
- ▶ **フェールソフト**と**フェールバック**は停止させると危険なシステムに採用される。
- ▶ **フェールセーフ**は、障害時にシステムの状態を安全側へ誘導し、その状態を保つ

デュアルシステム

▶ デュアルシステム

同じ処理を2つの処理系統で同時に行い、結果を照合するシステム

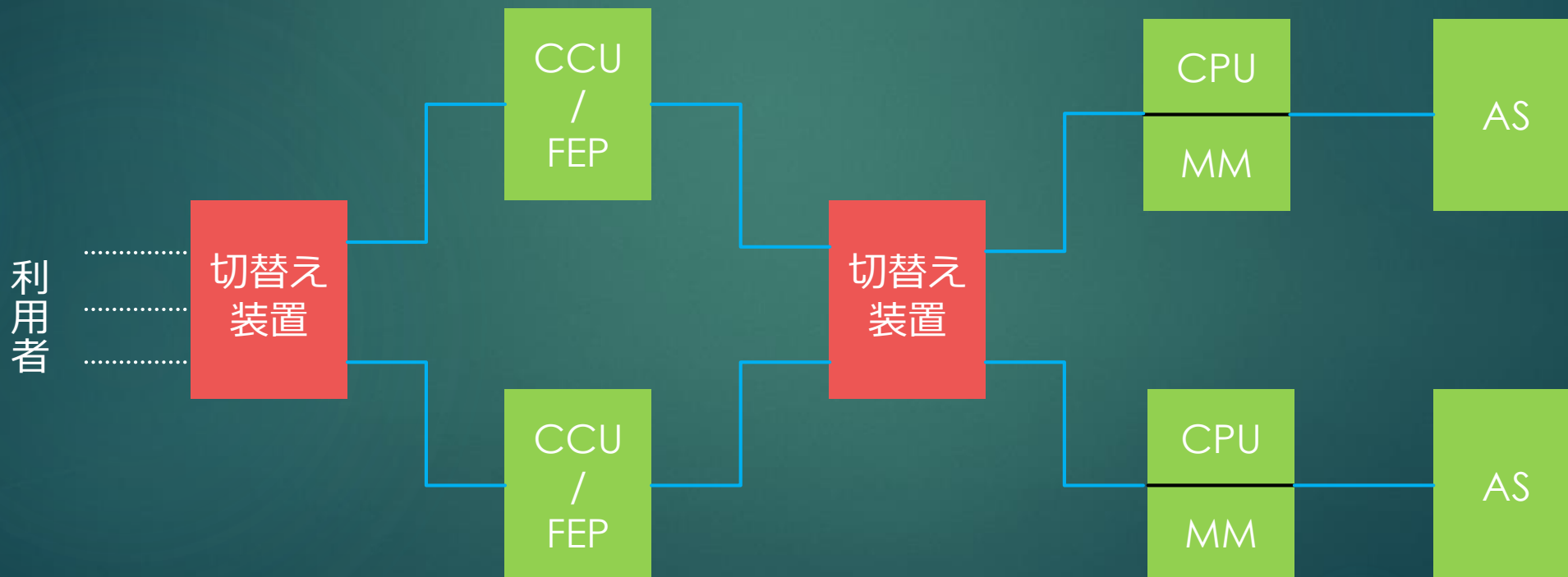


- ▶ 照合結果が一致しなかった場合、一般に診断用プログラムが実行される。これによってどちらか一方が障害状態と判定された場合は、その系統を切り離し、正常な系統だけで処理を継続することができる。

デュプレックスシステム

▶ デュプレックスシステム

プロセッサや記憶装置などを二重化しておき、**一方の系統に障害が発生した場合はもう一方での処理に切り替えるシステム**



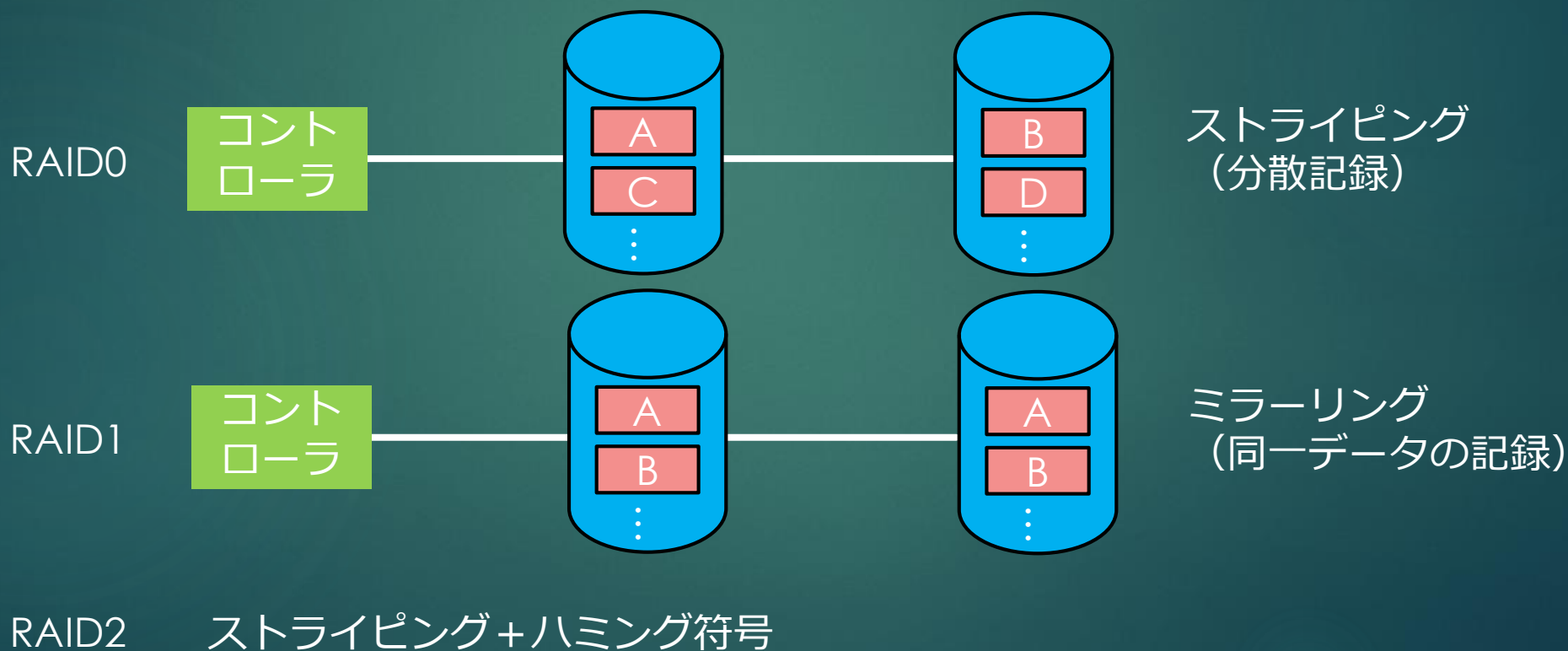
デュプレックスシステム

▶ 予備系の待機方法

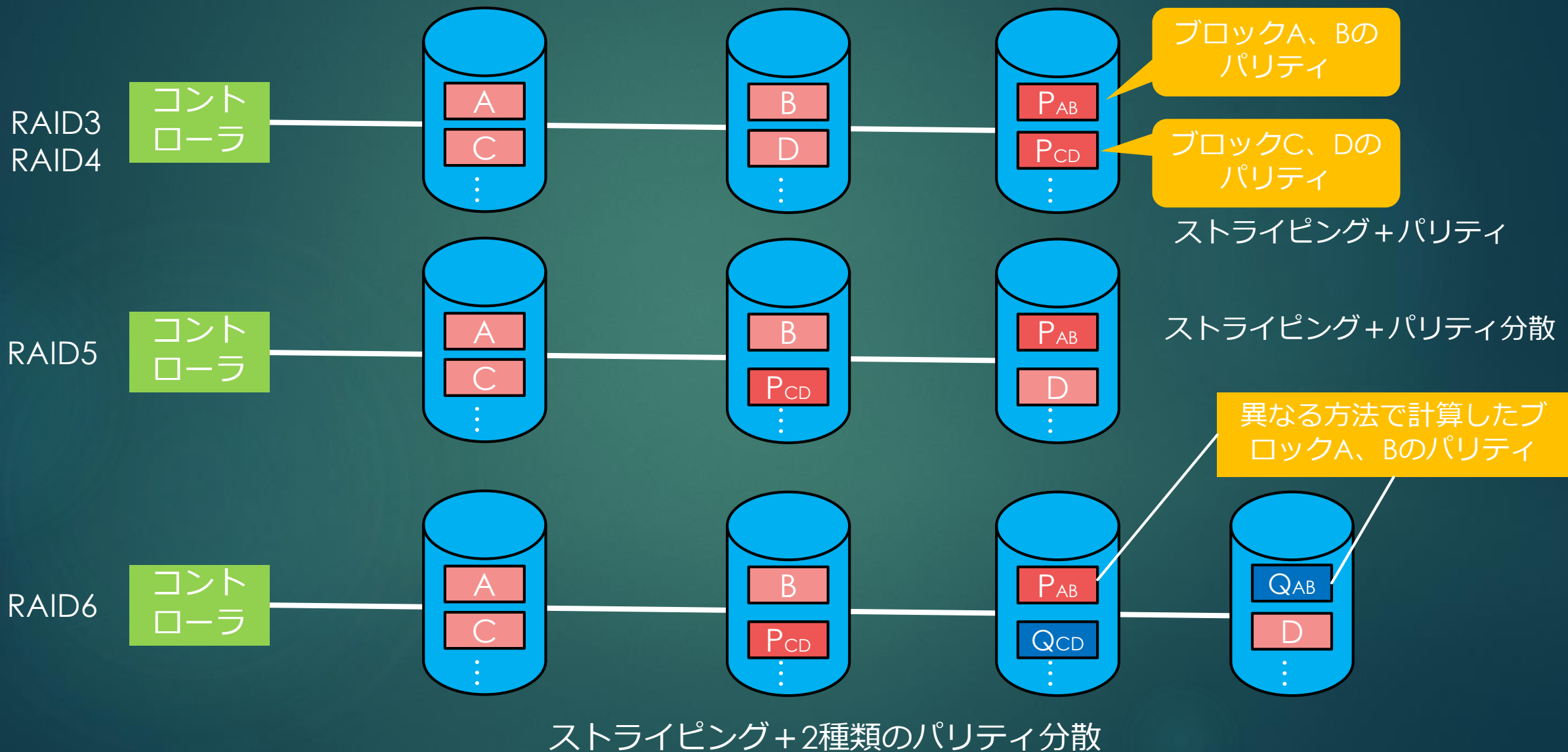
ホットスタンバイ	あらかじめ主処理用のOSおよび業務システムをロードし、 すぐに切り替えられる状態で待機 する
コールドスタンバイ	準備が整っていない、あるいは現用系とは異なる処理を実行している状態で待機する。現用系に障害が発生してから、OSの立上げや各種設定、業務システムの起動を行う
ウォームスタンバイ	ホットスタンバイとコールドスタンバイの中間。現用系と同じ状態で立ち上がっているが、業務システムは起動されていない状態で待機する

RAID(ディスクアレイ)

- ▶ 磁気ディスクはシステムの主要な要素である。そのための性能や信頼性を向上させるためには、磁気ディスクの並列化や多重化が欠かせない。磁気ディスクにおけるそれらの技術を**RAID**と総称する。RAIDにはレベルが存在する。



RAID(ディスクアレイ)



RAID(ディスクアレイ)

- ▶ **RAID0**は連続するデータブロックを、複数のディスクに分散して記録する(**ストライピング**)方式です。並列アクセスによる高速化が期待できるが、信頼性は向上しません。
- ▶ **RAID1**はデータブロックのコピーを副ディスクに記録する(**ミラーリング**)方式である。RAID0とは逆に信頼性は向上するが、高速化は望めない。
- ▶ **RAID2**はストライピング方式に加えて、誤り訂正のためのハミング符号を記録する方式である。
- ▶ **RAID3および4**は、ストライピングに加えて、誤り訂正のための**パリティ**を記録するので、**ディスク1台の故障であれば、その内容をほかのディスク内容から復旧できる**。RAID3とRAID4の違いはストライピング単位にある。ビット単位のRAID3に比べ、RAID4はブロック単位でストライピングするので、その分、並列してアクセスできる範囲が大きくなり、さらなる高速化が期待できる。

RAID(ディスクアレイ)

- ▶ RAID3,4では**パリティディスク**にアクセスが集中する。これを避けるため、**RAID5**はパリティブロックを分散して記録している。
- ▶ **RAID6**は、異なる計算で求めた2種類のパリティを分散記録する方式である。**同時に2台ディスクが故障した場合でも、その内容を復旧できる。**

パリティディスク
パリティのみを記録したディスク

NAS / SAN

- ▶ ネットワークの進歩に伴い、補助記憶装置もまたネットワーク接続され共有されるようになった。

NAS(Network Attached Storage)	LANに接続 され、複数のコンピュータからアクセスされる補助記憶装置
SAN(Storage Area Network)	補助記憶装置を接続する専用の高速ネットワーク。ファイバチャネルを用いて構築する。

- ▶ 通常のLANに接続される**NAS**に比べ、**SAN**をもちいたファイル共有はLANに与える負荷を少なくできる。

2. キャパシティプランニングと性能評価

キャパシティプランニング

- ▶ **キャパシティプランニング**とは、システムの開発や既存システムの改修において、

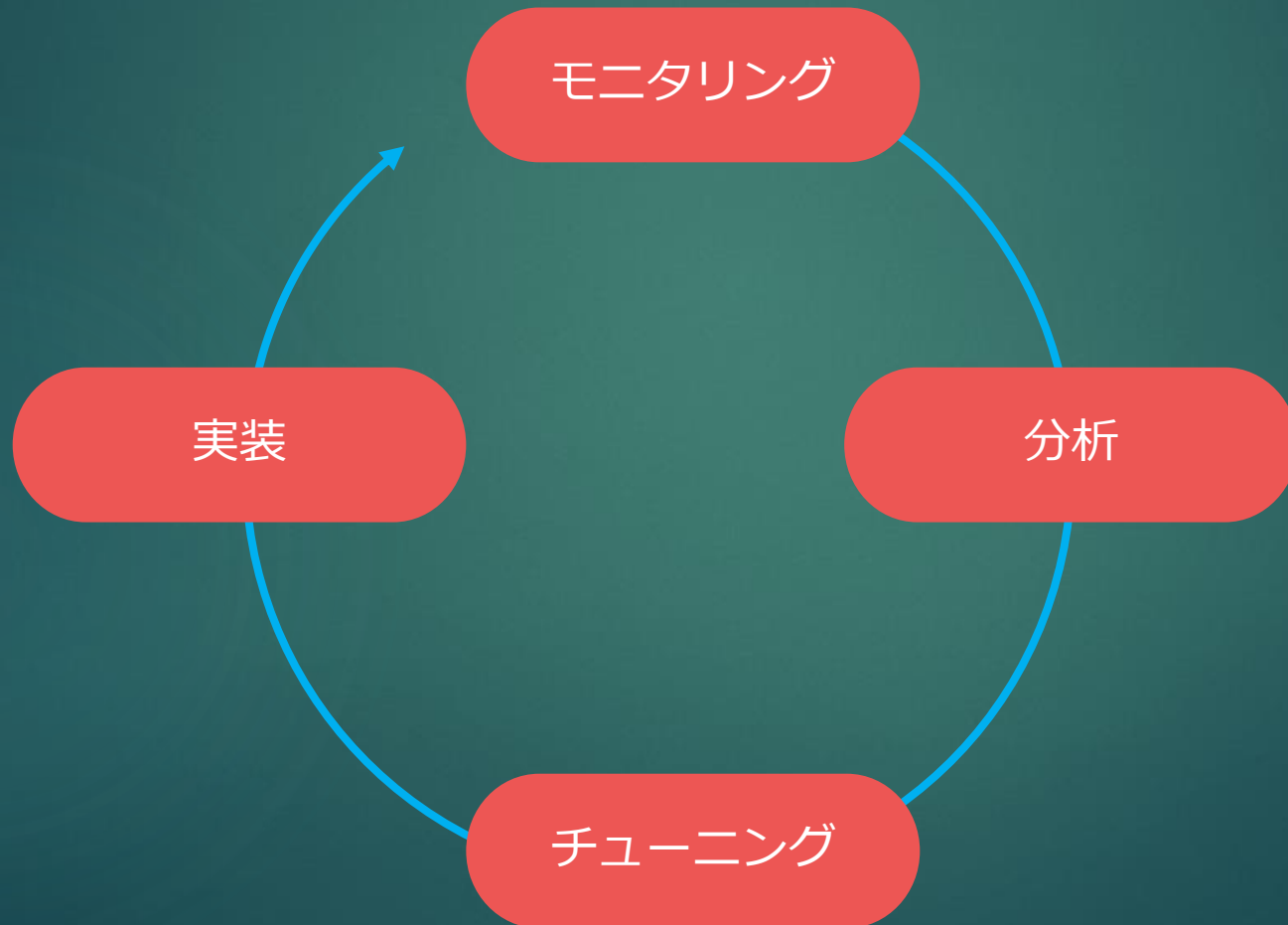
「**需要（ユーザ要求）を満足させ、負荷のピーク時にも耐え得るシステム構成**」

を設計をすることである。

- ▶ 具体的には、各アプリケーションごとに想定される必要な資源を算出し、システムの各構成要素について、その機能要素、量、配置場所などを決定することを指す。

キャパシティプランニングのサイクル

▶ キャパシティランニングのサイクル



キャパシティプランニング

▶ 各サイクルで行う活動

モニタリング	プログラムの実行状態や資源の利用状況などを測定し、 改善のためのデータを収集 する。測定周期や報告時期の計画も含まれる
分析	モニタリングで収集したデータをもとに、傾向や最大負荷など分析する
チューニング	現状システムの性能を最適化する ための変更箇所の検討や変更策を決定する
実装	現状システムを起点とし、将来的な予測に基づいて必要となるリソース調達やシステム増強などを計画する

キャパシティプランニング

▶ 主な性能評価指標

レスポンスタイム	要求入力完了の直後から、結果の出力開始まで要する時間
ターンアラウンドタイム	ジョブを提出してから、結果が戻るまでの時間
スループット	単位時間内に実行できるジョブの量
TPS	1秒間に処理できるトランザクション数

キャパシティプランニング

▶ ベンチマーク

システムに標準的なプログラムを実行させ、その結果をもとに性能を測定する方法。

▶ 主なベンチマーク

SPECint	整数演算能力を測るベンチマーク
SPECfp	浮動小数点数演算能力を測るベンチマーク
TPC-C	受発注業務システム(OLTP)をモデルとし、トランザクション処理能力を測るベンチマーク

トランザクション処理時間の算出

- ▶ スループットやTPS の算出に当たっては、トランザクションの処理時間を算出することが求められる。単位を秒として式は以下のようになる

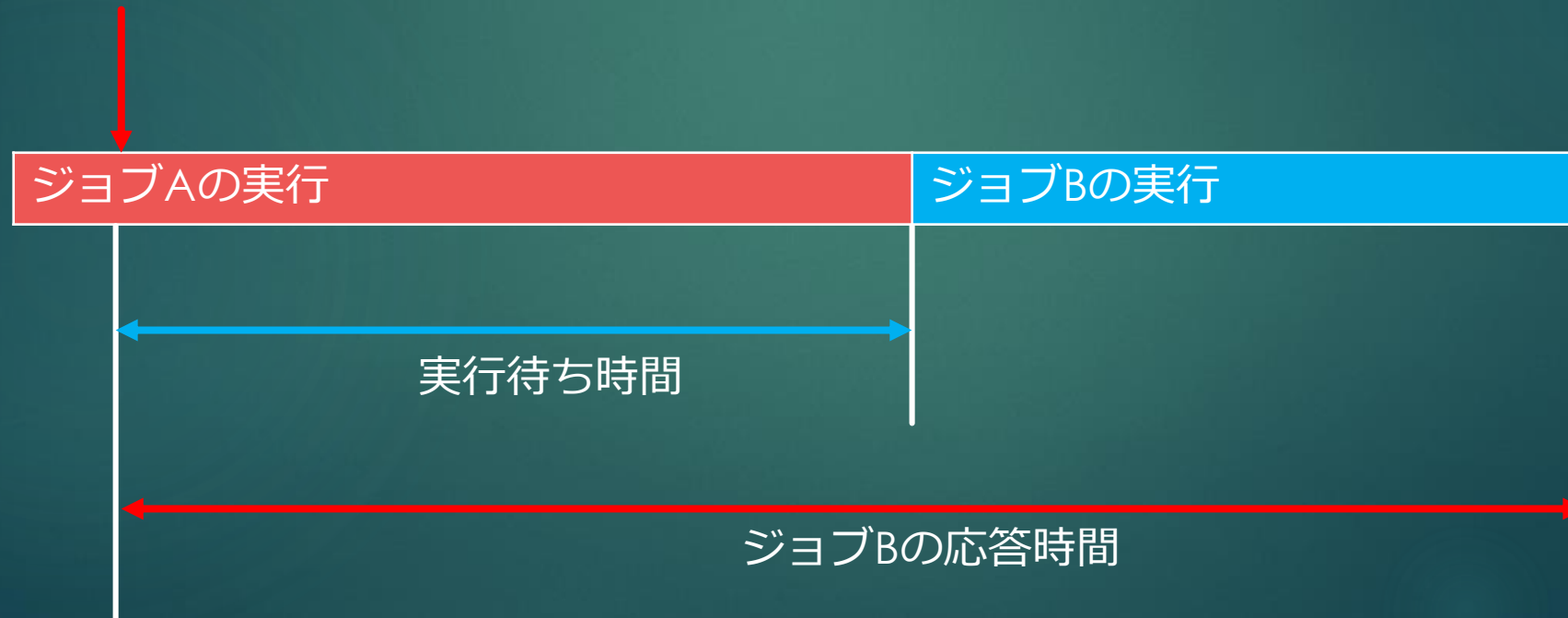
$$\text{トランザクションの処理時間} = \frac{\text{トランザクションの命令数}}{\text{MIPS値} \times 10^6}$$

- ▶ また、トランザクションの処理時間の逆数は1秒間に実行できるトランザクション数となる。

待ち時間を考慮した応答時間の算出

- ▶ ターンアラウンドタイムやレスポンスタイムの算出にあたって、時間をジョブやトランザクションの到着を含めて計算しなければならない。そのため、応答時間は実行の待ち時間を含める。

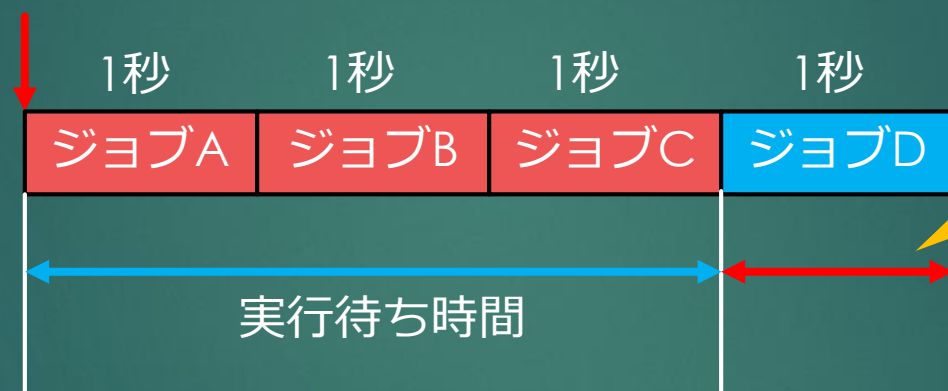
ジョブBの到着



スループットの算出

- ▶ スループットは単位時間に実行できるジョブやトランザクションの数を表す。個の算出に当たっては、待ち時間は計算に含めない。

ジョブDの到着

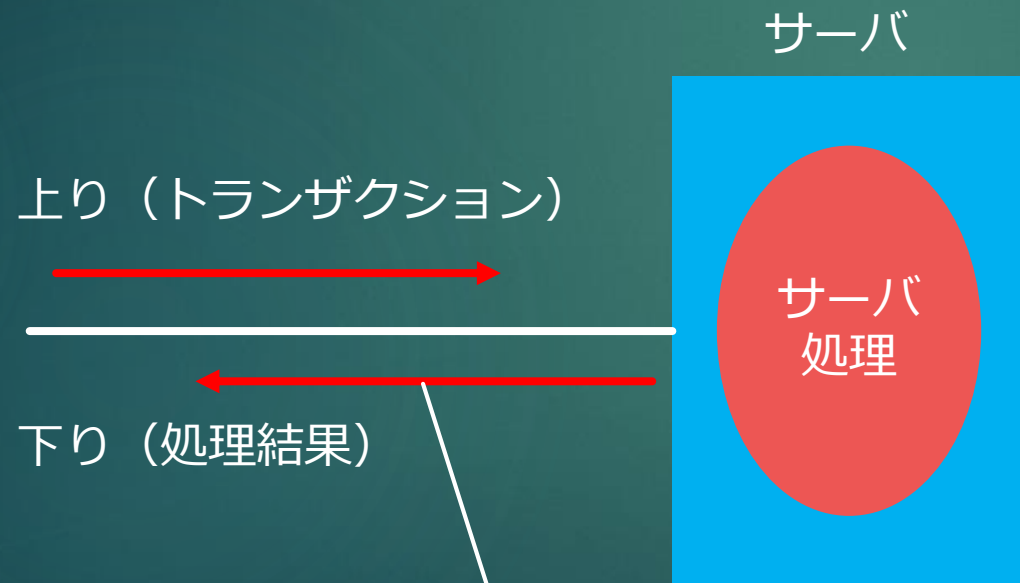


システムから見た
ジョブDの占有時間

- ▶ ジョブDから見ると到着から実行までに4秒必要となるが、待ち時間の間にもシステムではジョブA~Cが実行されているため、ジョブがシステムを占有する時間は1秒となる。よって、スループットは
$$3600 \div 1 = 3600 \text{ (件/時)}$$

伝送時間

- ▶ ネットワークを介してトランザクションや実行結果をやりとりするとき、伝送時間がボトルネックになることがある。以下の例で検討する
- ▶ (例) 伝送時間と処理時間



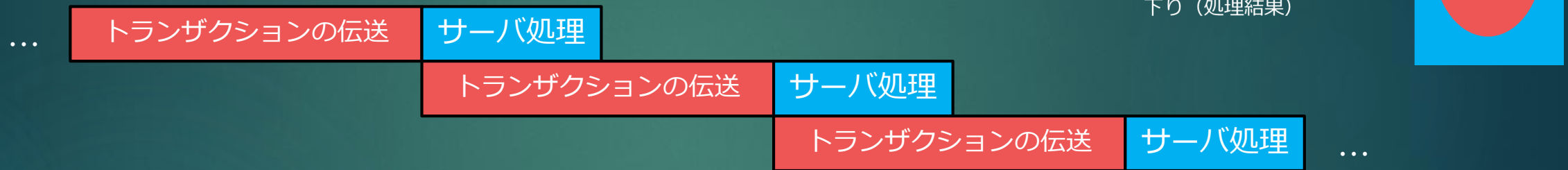
CASE1 伝送時間 : 1秒
サーバ処理時間 : **0.5秒**

CASE2 伝送時間 : 1秒
サーバ処理時間 : **2秒**

簡単化のため下り時間は無視とする

伝送時間(CASE 1)

▶ トランザクションの実行



- ▶ 伝送と実行は並列に行えるため、サーバがあるトランザクションを実行している間、ネットワークからは次のトランザクションが伝送される。

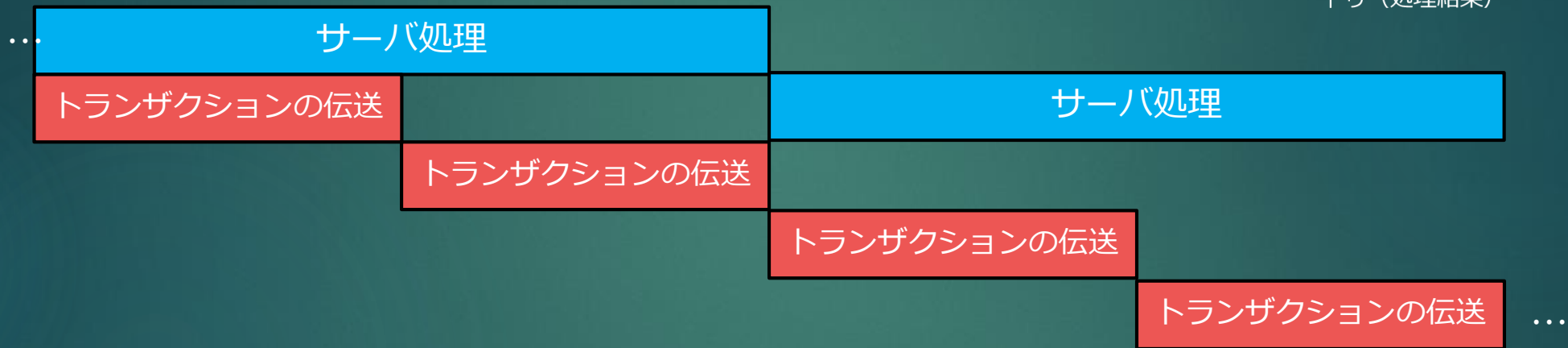
ネットワークの伝送時間 > サーバの処理時間

であるから、伝送時間がボトルネックとなり、サーバ処理を連続して実行できない。したがって、スループットは

$$1 \div \text{伝送時間} = 1(\text{件/秒})$$

伝送時間(CASE 2)

▶ トランザクションの実行



- ▶ トランザクションが次々と伝送されてくるが、サーバ処理が追いつかない状況であるからスループットは

$$1 \div \text{サーバ処理時間} = 0.5(\text{件/秒})$$

利用率と待ち時間の関係

- ▶ サーバにトランザクションが集中すると、それだけサーバの待ち時間が長くなる。このとき、待ち時間および応答時間の利用率 ρ を用いて次のようになる。

$$\text{待ち時間} = \frac{\rho}{1 - \rho} \times \text{サーバ処理時間}$$

$$\text{応答時間} = \frac{1}{1 - \rho} \times \text{サーバ処理時間}$$

たとえば、1秒間に3件のトランザクションが到着し、1トランザクションあたり0.2秒のサーバ処理が発生するとき、サーバ利用率は

$$0.2 \times 3 / 1 = 0.6$$

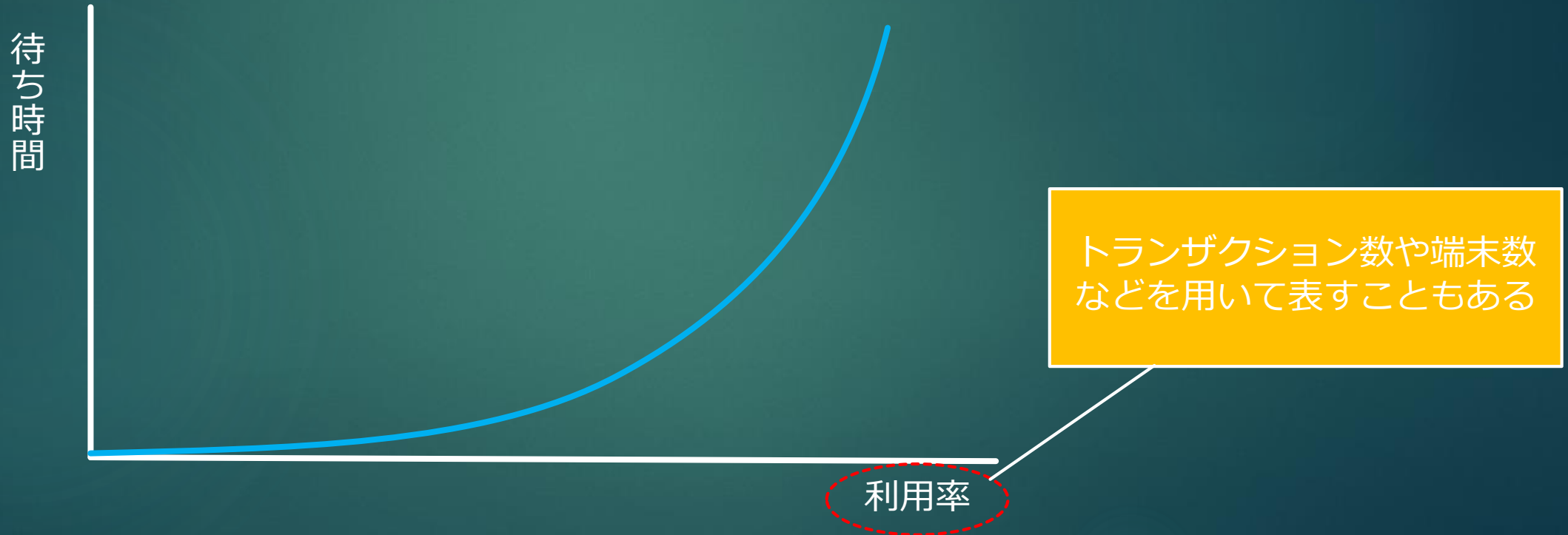
となり、

$$\text{待ち時間} = 0.6 / 0.4 \times 0.2 = 0.3(\text{秒})$$

$$\text{応答時間} = 1 / 0.4 \times 0.2 = 0.5(\text{秒})$$

利用率と待ち時間の関係

- ▶ 待ち時間や応答時間は利用率の増加に伴い極端に変化する。利用率が50%であれば、待ち時間はサーバ処理時間と等しいが、利用率が99%になればサーバ処理時間の99倍にまで悪化する。



ベンチマークを相乗平均で評価する

- ▶ ベンチマークの結果を相乗平均で評価することもある。たとえば、2本のベンチマークプログラムをコンピュータX, Yで実行したときの実行時間が以下の場合であったとする。

	プログラム1	プログラム2
コンピュータX	50	200
コンピュータY	100	100

- ▶ プログラム1、2の重みが同じ場合X、Yは同等の性能といえるが、X、Yの算術平均をとるとXは125、Yは100となりYの方がよいとなる。
- ▶ ここで相乗平均をとると

$$Xの評価 = \sqrt{50 \times 200} = 100$$

$$Yの評価 = \sqrt{100 \times 100} = 100$$

となり、性能比が評価値に反映されることになる。

なお、ベンチマークではなくモニタリングなどの「実際の処理時間の計測値」であれば、算術平均でも妥当な評価が得られる。

3. システムの信頼性

信頼性評価指標

- ▶ システムの信頼性を表す概念に**RASIS**がある。これは次の性質の頭文字をとったものである。

Reliability（狭義の信頼性）	システムの故障のしにくさ を表す →評価指標としてMTBFを用いる
Availability（可用性）	システムが使用できる可能性 を表す →評価指標として稼働率を用いる
Serviceability（保守性）	システムの保守しやすさ を表す →評価指標としてMTTRを用いる
Integrity（一貫性、完全性）	不整合の起こりにくさを表す
Security（安全性）	障害や不正アクセスに対する耐性を表す

信頼性評価指標

▶ MTBF(Mean Time Between Failures : 平均故障間隔)

故障が回復してから、次の故障が発生するまでの平均時間を指す。つまり、**故障なく稼働（連続稼働）する平均時間**を表し、この時間が長いほど信頼性に優れたシステムといえることができる。

▶ MTTR(Mean Time To Repair : 平均修理時間)

故障が発生から復旧までに要する平均時間を指す。この時間が短いほど、保守性に優れたシステムといえることができる。

▶ 稼働率

稼働率は、故障も含めた時間に対する「稼働時間」の割合である。この値が高いほど可用性に優れたシステムといえ、次式で表される。

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

つまり、稼働率を高めるためには、MTBFを相対的に長く、MTTRを相対的に短くすればよい。両者が同じ割合で変化した場合は稼働率は変わらない。

稼働率の計算

- ▶ 稼働率を計算するためには、平均故障間隔および平均修理時間を求めなければならない。

<例> 平均すると100時間に2回故障が発生し、その都度修理に2時間要する
このシステムの実行スケジュールは次のようになる。



MTBFは48時間となり、稼働率は

$$\frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}} = \frac{48}{50} = 0.96$$

直列システムの稼働率

- ▶ 直列に接続された機器は、CPUとメモリのようにそれぞれがシステムにとって必須の機器であることを表す。
- ▶ 装置Xの稼働率を x 、装置Yの稼働率を y とする。機器X, Yが直列に接続されているとき、システムとして稼働するためにはX, Yがともに稼働しなければならない。その確率は**両装置の稼働率の積 xy** で求めることができる。

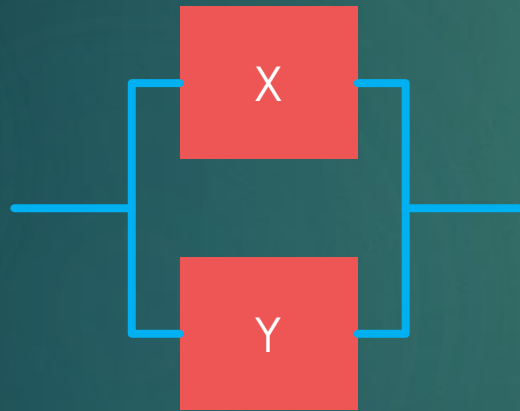


装置X	装置Y	確率
○	○	xy
○	×	$x(1 - y)$
×	○	$(1 - x)y$
×	×	$(1 - x)(1 - y)$

→ 稼働率

並列システムの稼働率

- ▶ 並列に接続された機器は、多重化された状態を表す。
- ▶ 装置Xの稼働率を x 、装置Yの稼働率を y とする。機器X, Yが並列に接続されているとき、システムとして稼働するためにはどちらか一方が稼働すればよいので、その確率は**両装置が稼働する確率に、いずれか一方のみが稼働する確率を加えて**計算することができる。



装置X	装置Y	確率
○	○	xy
○	×	$x(1 - y)$
×	○	$(1 - x)y$
×	×	$(1 - x)(1 - y)$

} この総和

$$\text{稼働率} = 1 - (1 - x)(1 - y)$$

複雑な構成をもつシステムの稼働率

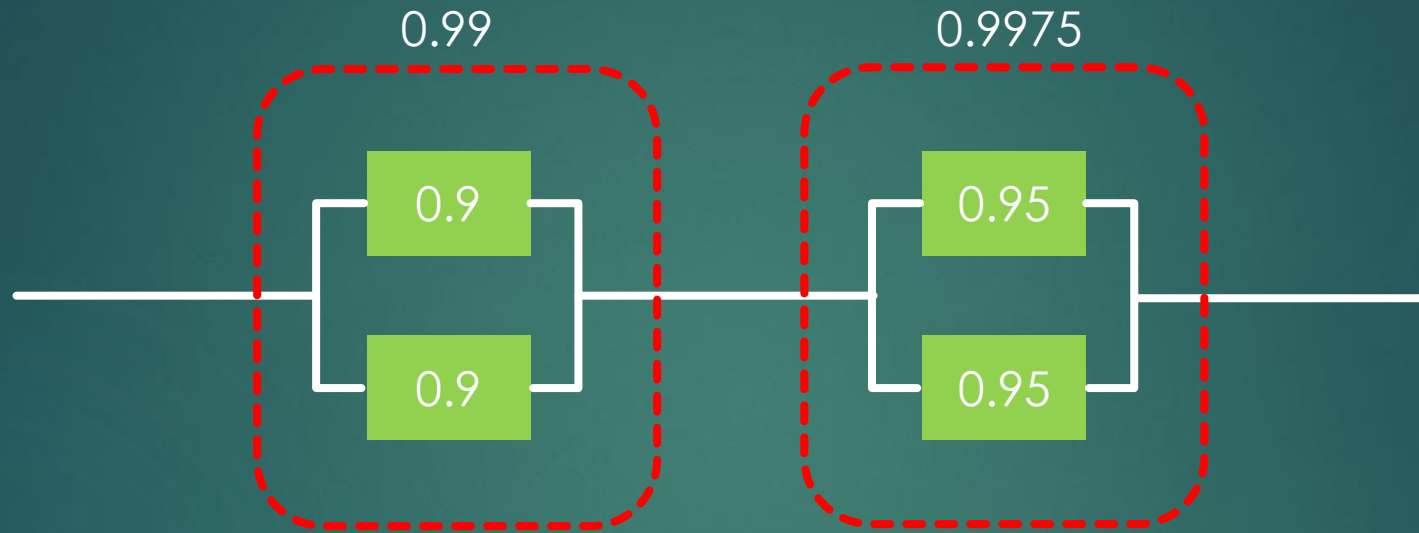
▶ n 重化システムの稼働率

機器を n 重に並列させた場合でも、稼働率の算出法事態は二重化の場合と変わらず、稼働率 x の機器を n 重化したときの稼働率は

$$1 - (1 - x)^n$$

と求めることができる。

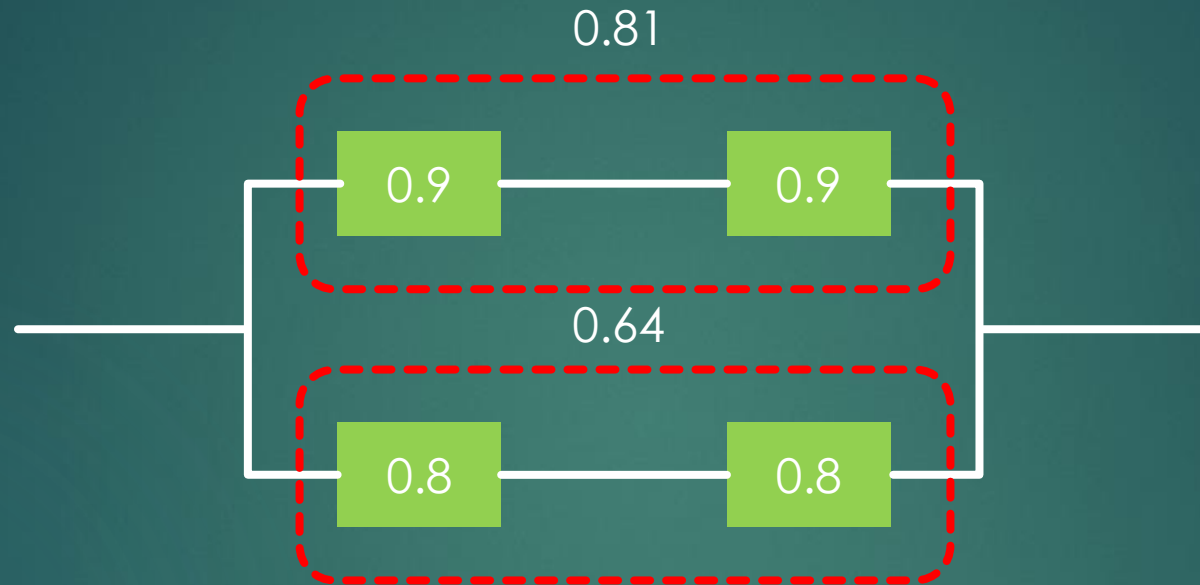
直 / 並列を組み合わせた構成の稼働率



- ▶ 上のような並列接続システムに直列に接続した場合の稼働率はまず並列接続部分の稼働率を求めた後に直列接続システムの稼働率とみて計算すればよい。
- ▶ したがって、稼働率は

$$0.99 \times 0.9975 \approx 0.988$$

直 / 並列を組み合わせた構成の場合



- ▶ 上のような直列接続システムに並列に接続した場合の稼働率はまず直列接続部分の稼働率を求めた後に並列接続システムの稼働率とみて計算すればよい。
- ▶ したがって、稼働率は

$$1 - (1 - 0.81)(1 - 0.64) \approx 0.932$$

狭義の信頼性

- ▶ 狭義の信頼性とは「故障を起こさず連続稼働する」性質のことを指す。
- ▶ 狭義の信頼性を表す指標

故障率 λ	単位時間内に故障が発生する割合 $\lambda = \frac{1}{\text{MTBF}}$
信頼性 $R(t)$	t 時間故障をおこさず連続稼働する確率 $R(t) = e^{-\lambda t}$

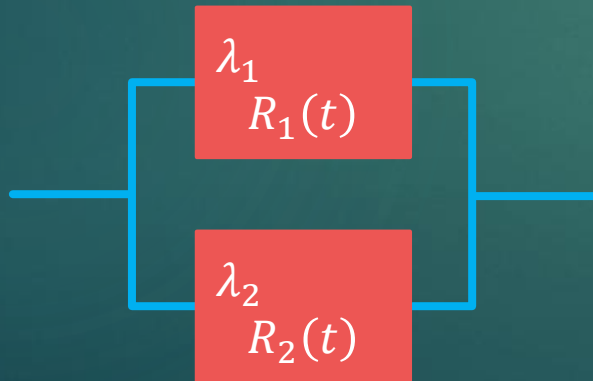
狭義の信頼性

- ▶ 直 / 並列システムの場合、故障率と信頼性は次のように計算することができる。



$$\text{故障率} = \lambda_1 + \lambda_2$$

$$\text{信頼性} = R_1(t) \times R_2(t)$$



$$\text{故障率} = \lambda_1 \times \lambda_2$$

$$\text{信頼性} = 1 - (1 - R_1(t)) \times (1 - R_2(t))$$