



東邦大学

いのち
生命の科学で未来をつなぐ

パイプライン

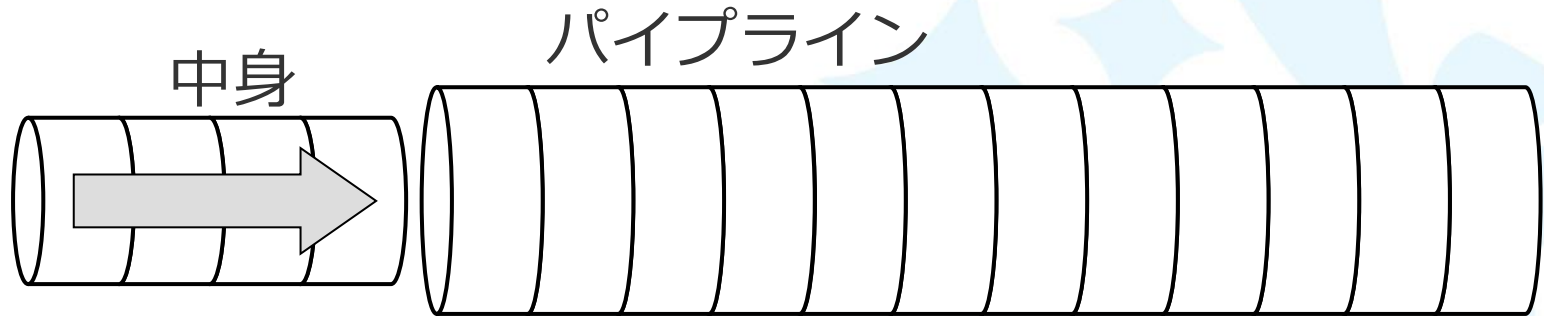
同じ原理を使って 2つのストーリーが



まず原理 ～ パイプラインとは

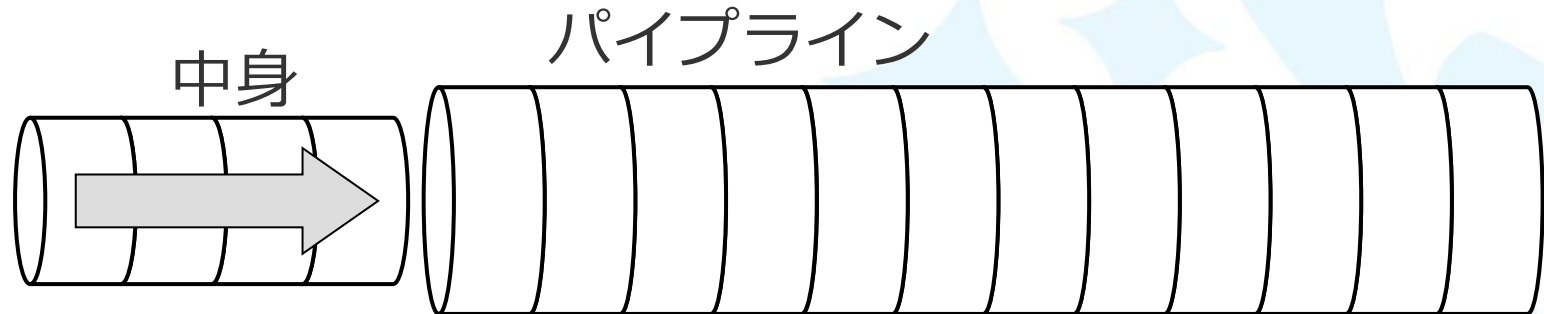


まず原理 ～ パイプラインとは



中身をパイプに押し込む
段々に先に進む

まず原理 ～ パイプラインとは



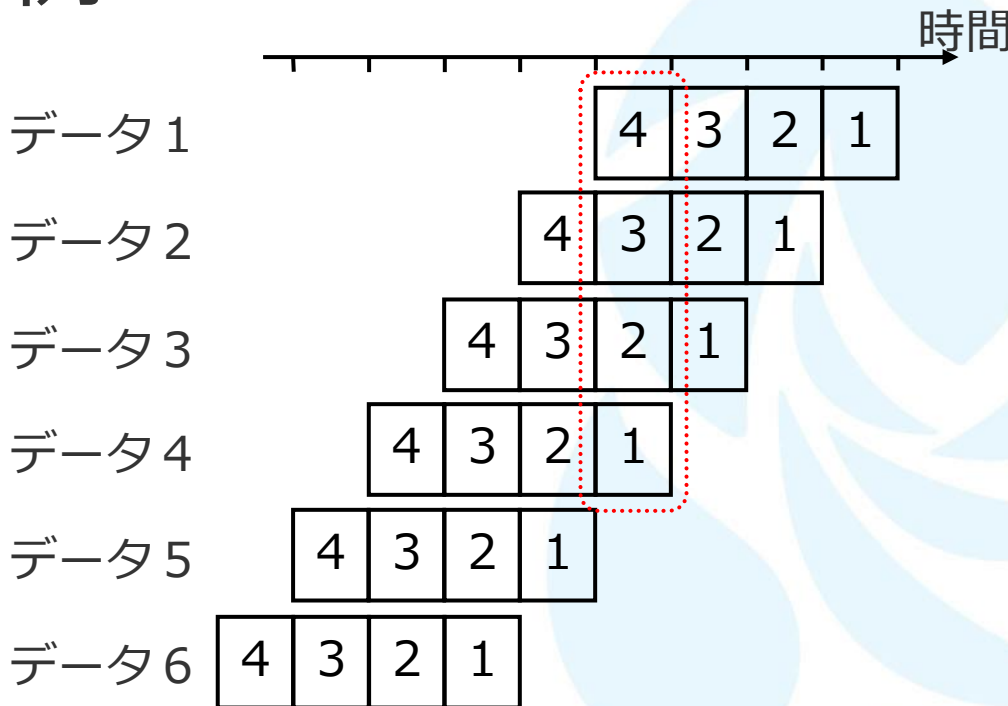
中身をパイプに押し込む
段々に先に進む

パイプは同じ長さのステージに分かれる
中身は1単位時間ごとに1ステージ進む

まず原理 ～ パイプラインとは

パイプは同じ長さのステージに分かれる
中身は1単位時間ごとに1ステージ進む

例



到着するデータは4段で
処理される（1～4）

左図時刻では、

データ1はステージ4

データ2はステージ3

データ3はステージ2

データ4はステージ1

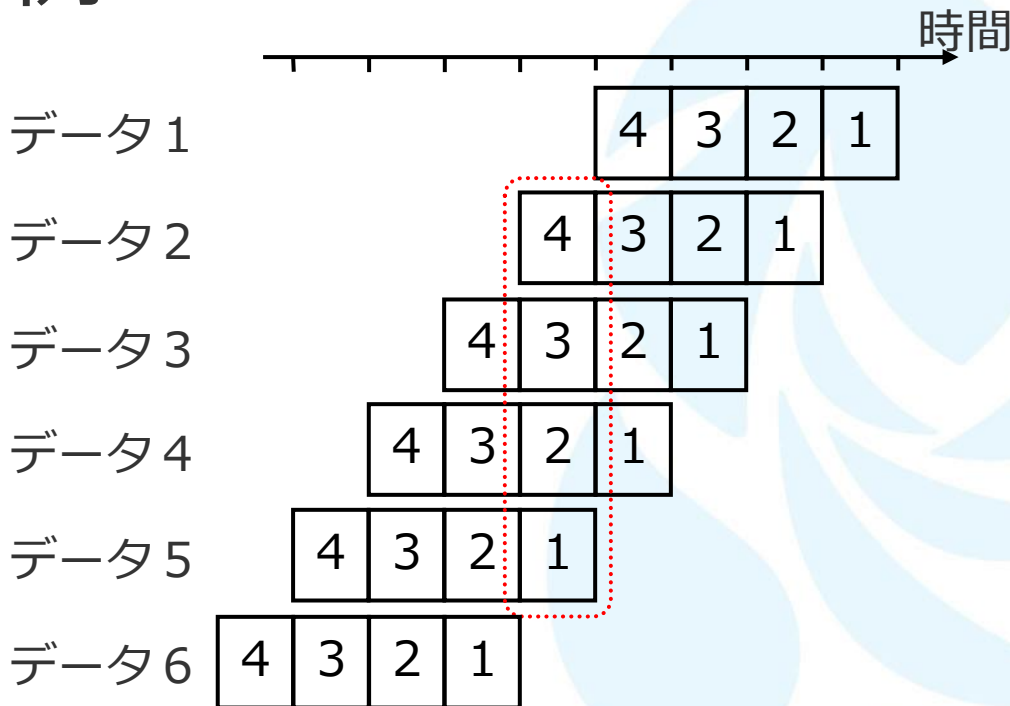
で処理されている



まず原理 ～ パイプラインとは

パイプは同じ長さのステージに分かれる
中身は1単位時間ごとに1ステージ進む

例



到着するデータは4段で
処理される（1～4）

左図時刻では、

データ2はステージ4

データ3はステージ3

データ4はステージ2

データ5はステージ1

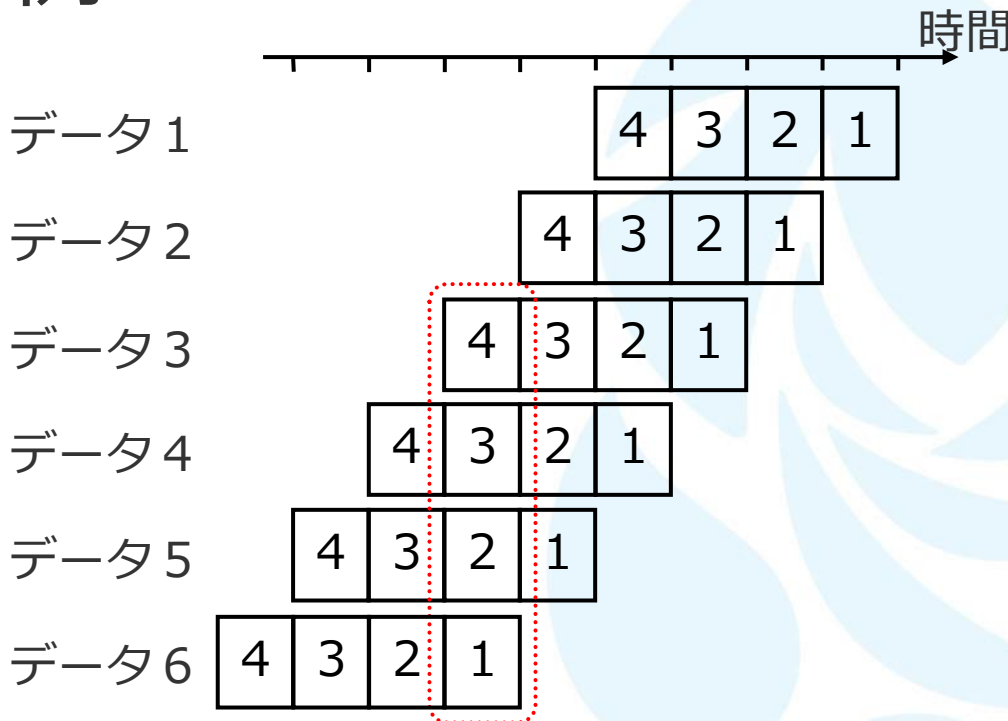
で処理されている



まず原理 ～ パイプラインとは

パイプは同じ長さのステージに分かれる
中身は1単位時間ごとに1ステージ進む

例



到着するデータは4段で
処理される（1～4）

左図時刻では、
データ3はステージ4
データ4はステージ3
データ5はステージ2
データ6はステージ1
で処理されている

まず原理 ～ パイプラインとは

このように、パイプラインでは
それぞれのステージが並行して動作する



まず原理 ～ パイプラインとは

このように、パイプラインでは
それぞれのステージが並行して動作する

⇒ 処理が早くなる

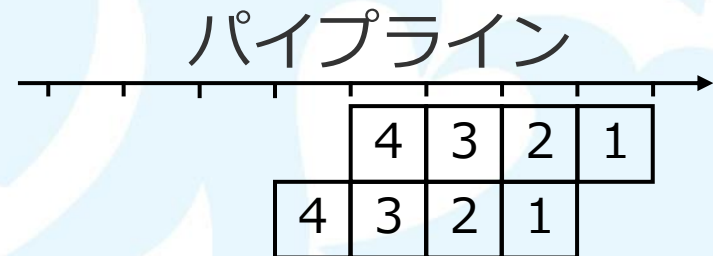
まず原理 ～ パイプラインとは

このように、パイプラインでは
それぞれのステージが並行して動作する

⇒ 処理が早くなる



2データで8クロック



2データで5クロック

どれだけ高速化するか
Sステージ、Nデータの所要時間は？



どれだけ高速化するか

Sステージ、Nデータの所要時間は？



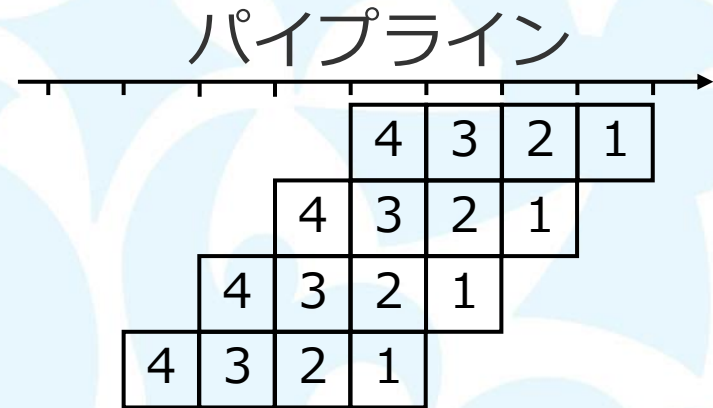
Nデータで $N \times S$ クロック

どれだけ高速化するか

Sステージ、Nデータの所要時間は？



Nデータで $N \times S$ クロック



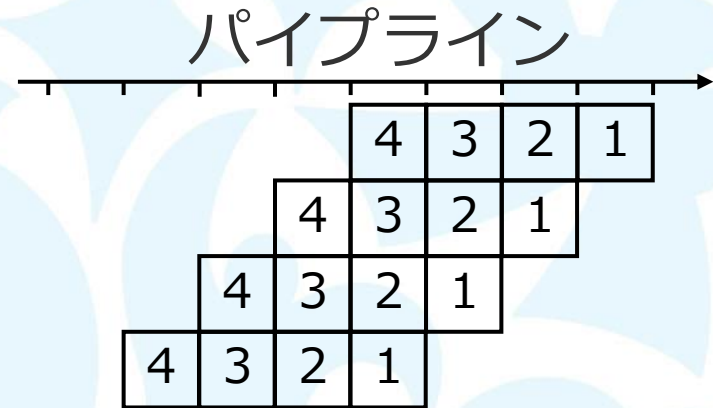
Nデータで $N + S - 1$ クロック

どれだけ高速化するか

Sステージ、Nデータの所要時間は？



Nデータで $N \times S$ クロック



Nデータで $N + S - 1$ クロック

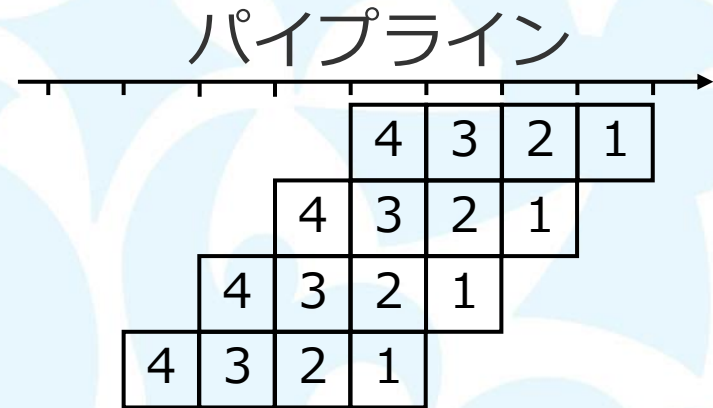
加速率 = $T(\text{直列}) / T(\text{パイプライン}) =$

どれだけ高速化するか

Sステージ、Nデータの所要時間は？



Nデータで $N \times S$ クロック



Nデータで $N + S - 1$ クロック

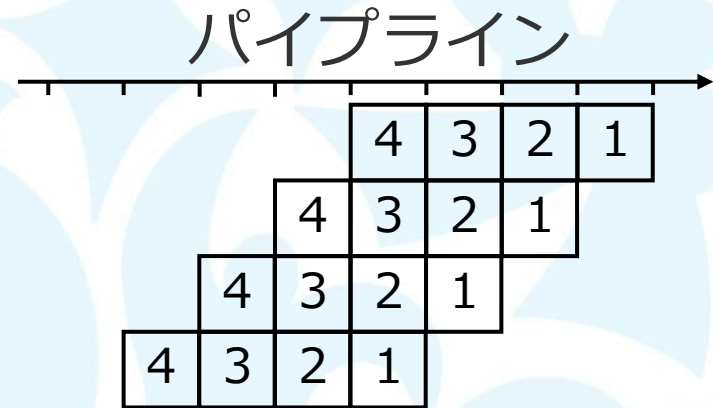
$$\text{加速率} = T(\text{直列}) / T(\text{パイプライン}) = (N \times S) / (N + S - 1)$$

どれだけ高速化するか

Sステージ、Nデータの所要時間は？



Nデータで $N \times S$ クロック



Nデータで $N + S - 1$ クロック

加速率 = $T(\text{直列}) / T(\text{パイプライン}) = (N \times S) / (N + S - 1)$

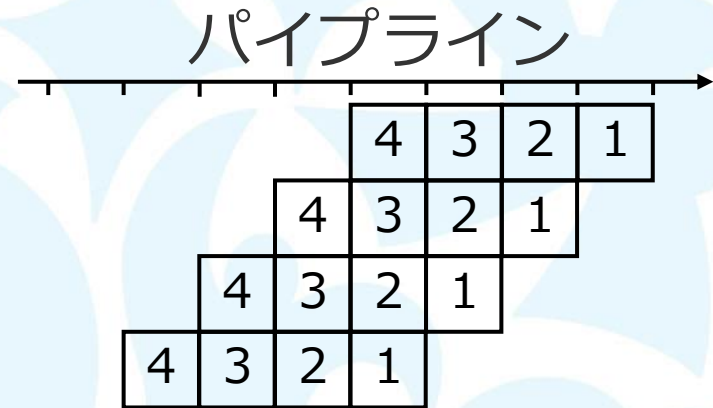
データが無限に続くとする

どれだけ高速化するか

Sステージ、Nデータの所要時間は？



Nデータで $N \times S$ クロック



Nデータで $N + S - 1$ クロック

加速率 = $T(\text{直列}) / T(\text{パイプライン}) = (N \times S) / (N + S - 1)$

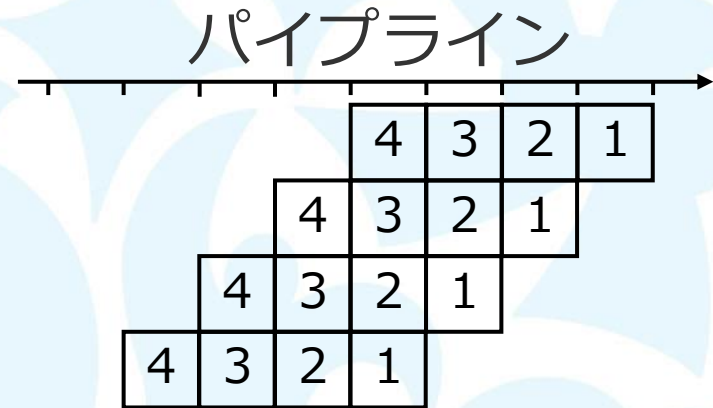
データが無限に続くとする (分母・子をNで割って)

どれだけ高速化するか

Sステージ、Nデータの所要時間は？



NデータでN×Sクロック



NデータでN+S-1クロック

$$\text{加速率} = T(\text{直列})/T(\text{パイプライン}) = (N \times S)/(N + S - 1)$$

データが無限に続くとする (分母・子をNで割って)

$$\text{加速率} = \lim_{(N \rightarrow \infty)} S/(1 + (S-1)/N) = S \Rightarrow \text{S倍}$$

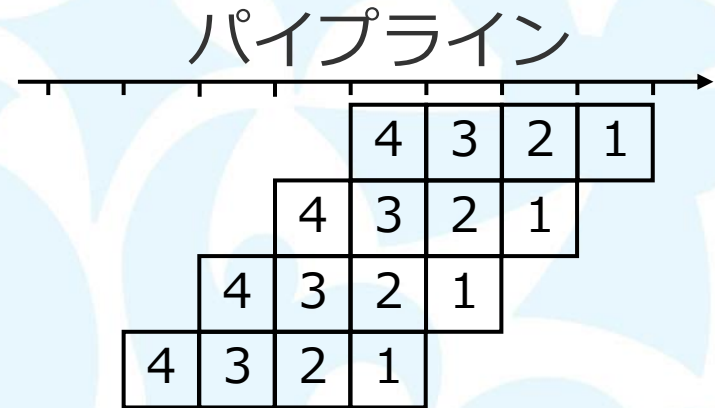


どれだけ高速化するか

Sステージ、Nデータの所要時間は？



NデータでN×Sクロック



NデータでN+S-1クロック

加速率 = $T(\text{直列})/T(\text{パイプライン}) = (N \times S)/(N + S - 1)$

データが無限に続くとする (分母・子をNで割って)

加速率 = $\lim_{(N \rightarrow \infty)} S/(1 + (S-1)/N) = S \Rightarrow S$ 倍

S (ステージ数) 倍 速くなる



実際の使われ方 2つ



実際の使われ方 2つ

使い方 1) 命令パイプライン

使い方 2) データパイプライン



命令パイプライン

1つ1つの命令の「実行サイクル」？

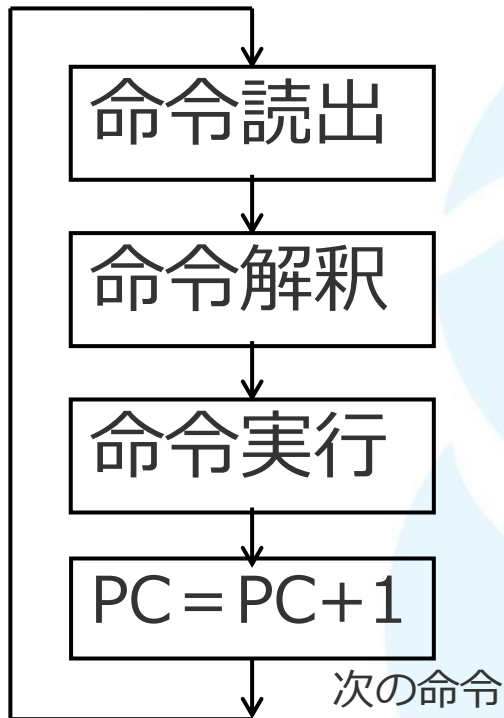
思い出してみよう



命令パイプライン

1つ1つの命令の「実行サイクル」？

思い出してみよう



1つの命令は4つのステップで実行される

これをぐるぐる回りながら次々と命令を順番に実行する

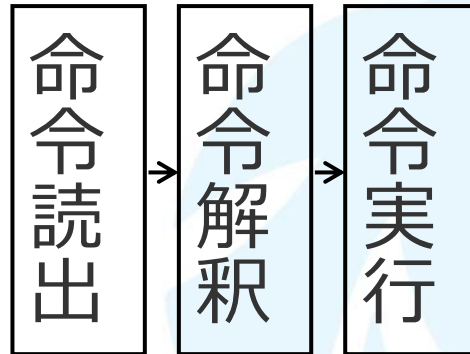
命令パイプライン

で



命令パイプライン

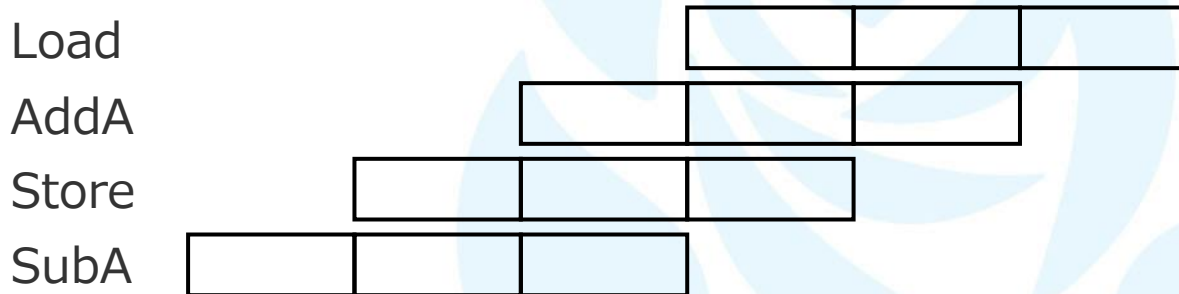
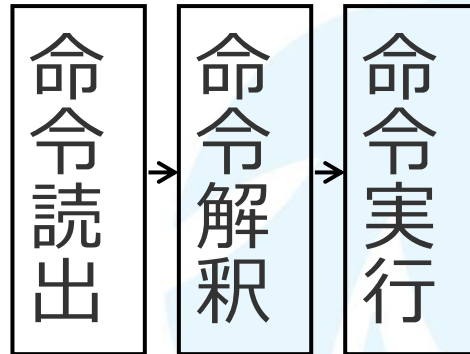
これらのステップをステージとして
パイプラインを作る



PC←PC+1は
高速なので
読出ステージ
に含める

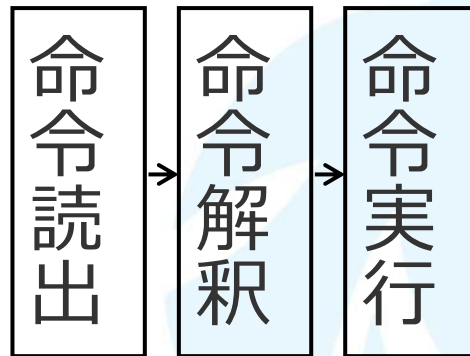
命令パイプライン

これらのステップをステージとして
パイプラインを作る



命令パイプライン

これらのステップをステージとして
パイプラインを作れる



実行ステージ ~ Load命令を実行
解釈ステージ ~ AddA命令を解釈
読出ステージ ~ Store命令を読出

Load

AddA

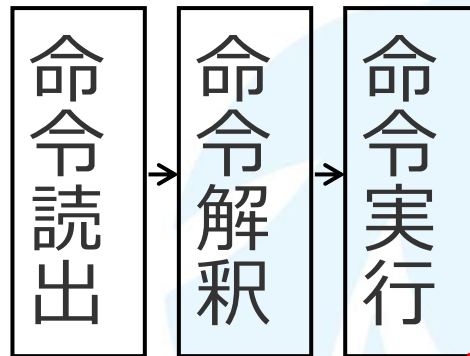
Store

SubA

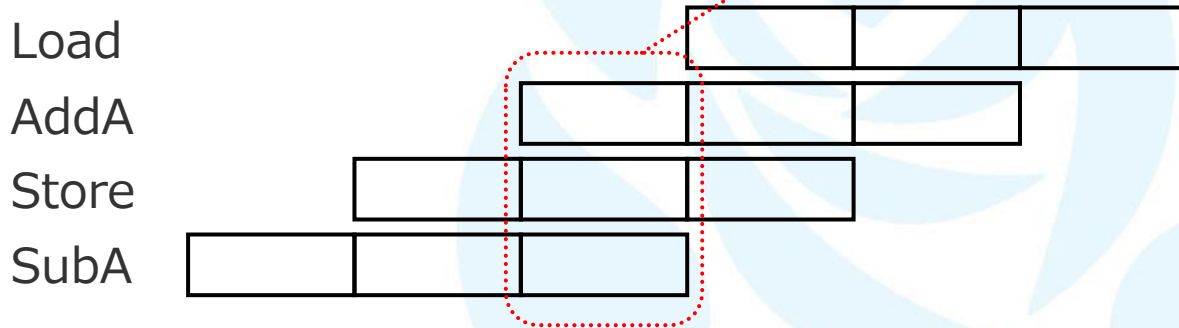


命令パイプライン

これらのステップをステージとして
パイプラインを作れる



実行ステージ ~ AddA命令を実行
解釈ステージ ~ Store命令を解釈
読出ステージ ~ SubA命令を読出



命令パイプライン

このように1つの命令の実行を
複数のステージに分割して
パイプライン化する
⇒ 命令パイプライン

命令パイプライン

このように1つの命令の実行を
複数のステージに分割して
パイプライン化する
⇒ 命令パイプライン

ステージ数を増やせば、並列度が上がり
高速化できる

Intel Core i7では16段程度まで増やしている



データパイプライン

同じ処理をしたいデータが
たくさんあるときに、パイプライン化

どういうときだろうか？



データパイプライン

同じ処理をしたいデータが
たくさんあるときに、パイプライン化
ベクトルや行列の計算
物理的なシミュレーション計算など

データパイプライン

同じ処理をしたいデータが
たくさんあるときに、パイプライン化

ベクトルや行列の計算
物理的なシミュレーション計算など
画像処理・生成など



データパイプライン

同じ処理をしたいデータが
たくさんあるときに、パイプライン化

ベクトルや行列の計算
物理的なシミュレーション計算など
画像処理・生成など

物理的なシミュレーション計算など ⇒
すべての点に対して同じ計算をする（ベクトル・行列）
昔のスパコンは「ベクトル計算機」であった
例： CDC-6600 (1964)、Cray-1 (1976)



データパイプライン

同じ処理をしたいデータが
たくさんあるときに、パイプライン化

ベクトルや行列の計算
物理的なシミュレーション計算など
画像処理・生成など

画像処理・生成など ⇒

最近のPCやグラフィックアダプタは画像処理
プロセッサ(GPU) を積んでいるが、これらはポリゴン
やピクセルごとに同一処理をするベクトル処理が
主体になっている



パイプラインの用途のまとめ

命令パイプライン

データパイプライン



パイプラインの用途のまとめ

命令パイプライン

- ・ 1つ1つの命令実行を、パイプラインに組む
- ・ 読出・解釈・実行（更にここを多段にする）

データパイプライン



パイプラインの用途のまとめ

命令パイプライン

- 1つ1つの命令実行を、パイプラインに組む
- 読出・解釈・実行（更にここを多段にする）
- PC用を含む広範なプロセッサで実用している

データパイプライン



パイプラインの用途のまとめ

命令パイプライン

- 1つ1つの命令実行を、パイプラインに組む
- 読出・解釈・実行（更にここを多段にする）
- PC用を含む広範なプロセッサで実用している

データパイプライン

- 多数のデータに対し同じ処理を行う時に有効

パイプラインの用途のまとめ

命令パイプライン

- 1つ1つの命令実行を、パイプラインに組む
- 読出・解釈・実行（更にここを多段にする）
- PC用を含む広範なプロセッサで実用している

データパイプライン

- 多数のデータに対し同じ処理を行う時に有効
- ベクトル・行列のような数値計算（物理シミュレーションなど）や

パイプラインの用途のまとめ

命令パイプライン

- 1つ1つの命令実行を、パイプラインに組む
- 読出・解釈・実行（更にここを多段にする）
- PC用を含む広範なプロセッサで実用している

データパイプライン

- 多数のデータに対し同じ処理を行う時に有効
- ベクトル・行列のような数値計算（物理シミュレーションなど）や
グラフィック処理（画像の生成や処理）・マルチメディア処理（画像・音声など）
が中心



パイプラインの用途のまとめ

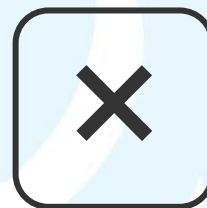
命令パイプライン

- 1つ1つの命令実行を、パイプラインに組む
- 読出・解釈・実行（更にここを多段にする）
- PC用を含む広範なプロセッサで実用している

データパイプライン

- 多数のデータに対し同じ処理を行う時に有効
- ベクトル・行列のような数値計算（物理シミュレーションなど）や
グラフィック処理（画像の生成や処理）・マルチメディア処理（画像・音声など）
が中心

パイプラインの考え方が
分かりましたか？



↓
次へ