東邦大学

生命の科学で未来をつなぐ

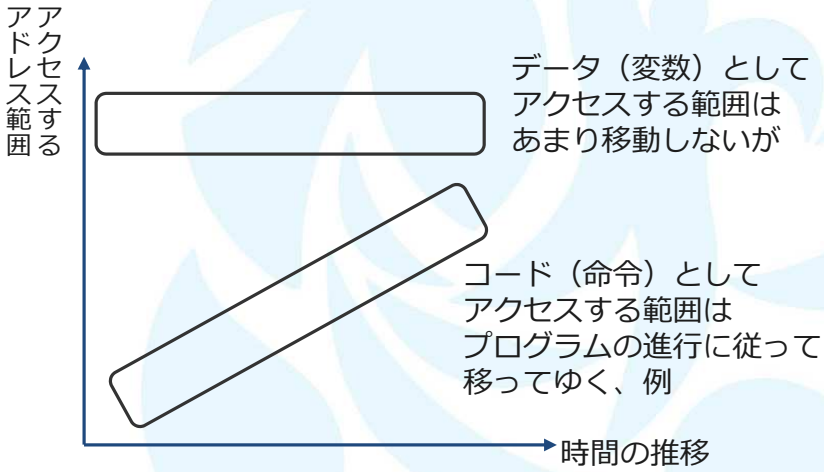
デマンドページングの性能５ ワーキングセット・まとめ

ワーキングセットの考え方

- プログラムが動いていく中で
  ある時間範囲 [t-w から t まで] の中で
  プログラムが参照するメモリ範囲（ページ範囲）

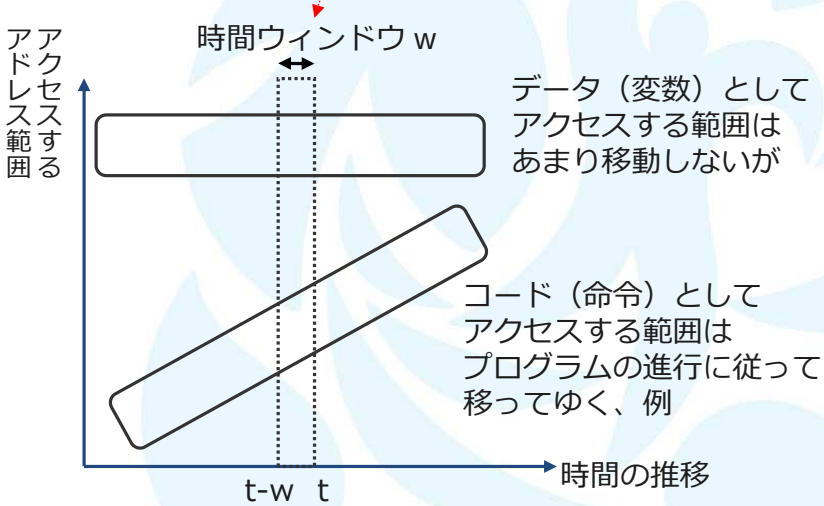
ワーキングセットの考え方

- プログラムが動いていく中で
  ある時間範囲 [t-w から t まで] の中で
  プログラムが参照するメモリ範囲（ページ範囲）
- 例えば



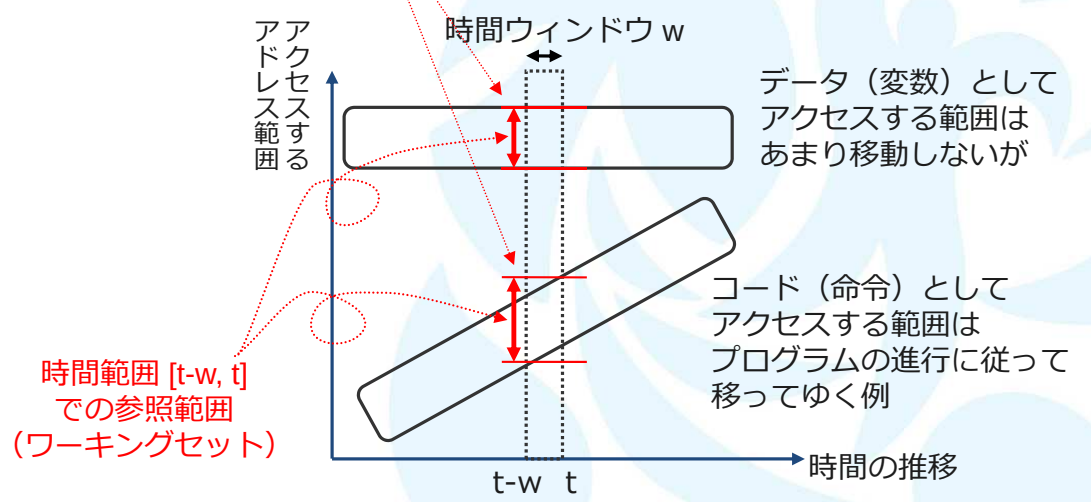
ワーキングセットの考え方

- プログラムが動いていく中で
  ある時間範囲 [t-w から t まで] の中で
  プログラムが参照するメモリ範囲（ページ範囲）
- 例えば

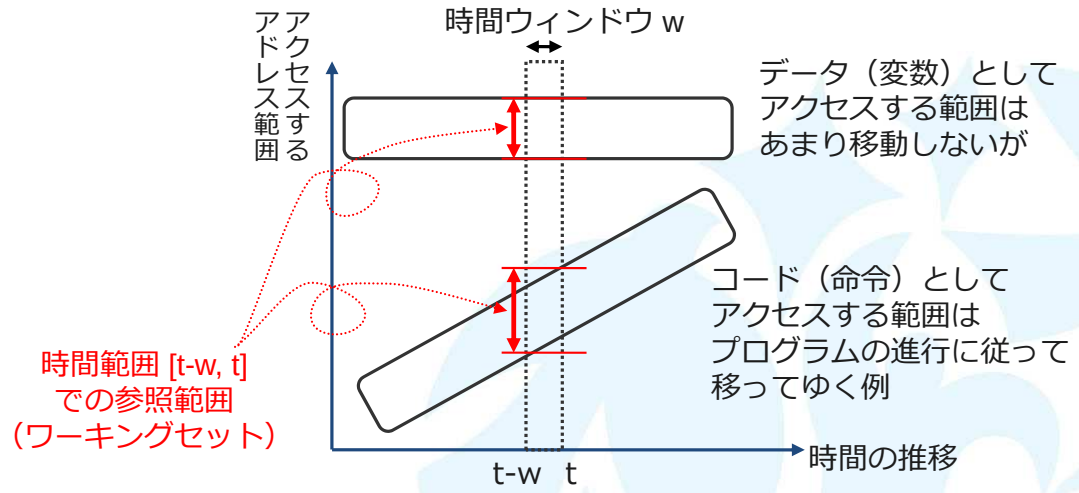


ワーキングセットの考え方

- プログラムが動いていく中で
  ある時間範囲 [t-w から t まで] の中で
  プログラムが参照するメモリ範囲（ページ範囲）
- 例えば



ワーキングセットの考え方



- ワーキングセットの部分が
  物理メモリ中に収まっていれば
  その時間帯内でページ置換えは起こらないはず
- なるべくワーキングセットが収まるのがよい

東邦大学

ここまでのまとめ

- デマンドページングは、ミス率が小さい時（ヒット率が大きい時）に実用的になるが

ここまでのまとめ

- デマンドページングは、ミス率が小さい時（ヒット率が大きい時）に実用的になるが
- ミス率は参照の局所性が成立つことによって小さくなる
  - 命令読出しの場合、データ参照の場合

ここまでのまとめ

- デマンドページングは、ミス率が小さい時（ヒット率が大きい時）に実用的になるが
- ミス率は参照の局所性が成立つことによって小さくなる
  - 命令読出しの場合、データ参照の場合
- 実測によって実用になることが示されている

ここまでのまとめ

- デマンドページングは、ミス率が小さい時（ヒット率が大きい時）に実用的になるが
- ミス率は参照の局所性が成立つことによって小さくなる
  - 命令読出しの場合、データ参照の場合
- 実測によって実用になることが示されている
- 現に広く実用になってきた・なっている
  - メインフレームOS、Linux、Windowsなど

東邦大学

（次の話題）　ページアウト

- もし物理メモリが満杯だったら
  どれかを追出す　（置き換えることになる）
- 追出すページの選択方法
  （＝置換えアルゴリズム）
  ⇒　性能に影響　（ここが議論になる）
- （工夫）欲しい時に直ぐ空きページがあるように
  常に一定以上の空きページをプールする
  - 常時見張っていて、不足したら追出しをするような
    常駐サーバー（ページアウトデーモン）を置く

デマンドページングの
性能がなぜ得られるのか
理解できましたか？