

QC-HPC Co-Scheduler シミュレーション

進捗整理

胡 曜

東京大学情報基盤センター

2025年1月17日更新

QC-HPC Co-Schedulerシミュレーションについて

- シミュレーションで想定したこと
- QC-HPCスケジューリング規則
- QCジョブスケジューリングの動作手順
- ジョブの再スケジューリング
- いくつかのユースケース
- シミュレーションのデモ

シミュレーションで想定したこと

- 複数HPC、複数QC
- 通常HPCジョブ、QC-HPC連成ジョブは同じキューで流れる（混載キュー）
- ジョブはHPC側から投入
 - ジョブタイプ（HPC, QC-HPC）、HPC占有ノード数、予測実行時間、開始時間、優先度を指定

QC1かQC2を指定
する必要がある

QC-HPCスケジューリング規則

- QCジョブは、HPCジョブよりも優先してスケジュールする
- QCジョブの実行順は、**優先度順→到着順**に基づいて決める
- QCジョブを実行するために、実行中のHPCジョブを一時止める可能性がある
 - 終了直前のHPCジョブは止めない
 - 止められたHPCジョブは、以降の時点で再開する

QCジョブスケジューリングの動作手順

- ACCEPT状態として投入し、HPC側からQC-HPCコスケジューラ側へQC資源占有状況（セマフォ）を問い合わせ、占有された時間帯を避ける
- HPCノードの占有状況を問い合わせ、予測実行時間内の空きノードを確保する（ジョブの状態をQUEUEDにする）
 - HPC側の空きノードが不足している場合、HPCジョブを一時中止するか再スケジュールする
- スケジューリング期間に入れないなど失敗した場合、ジョブの状態をHOLDにする

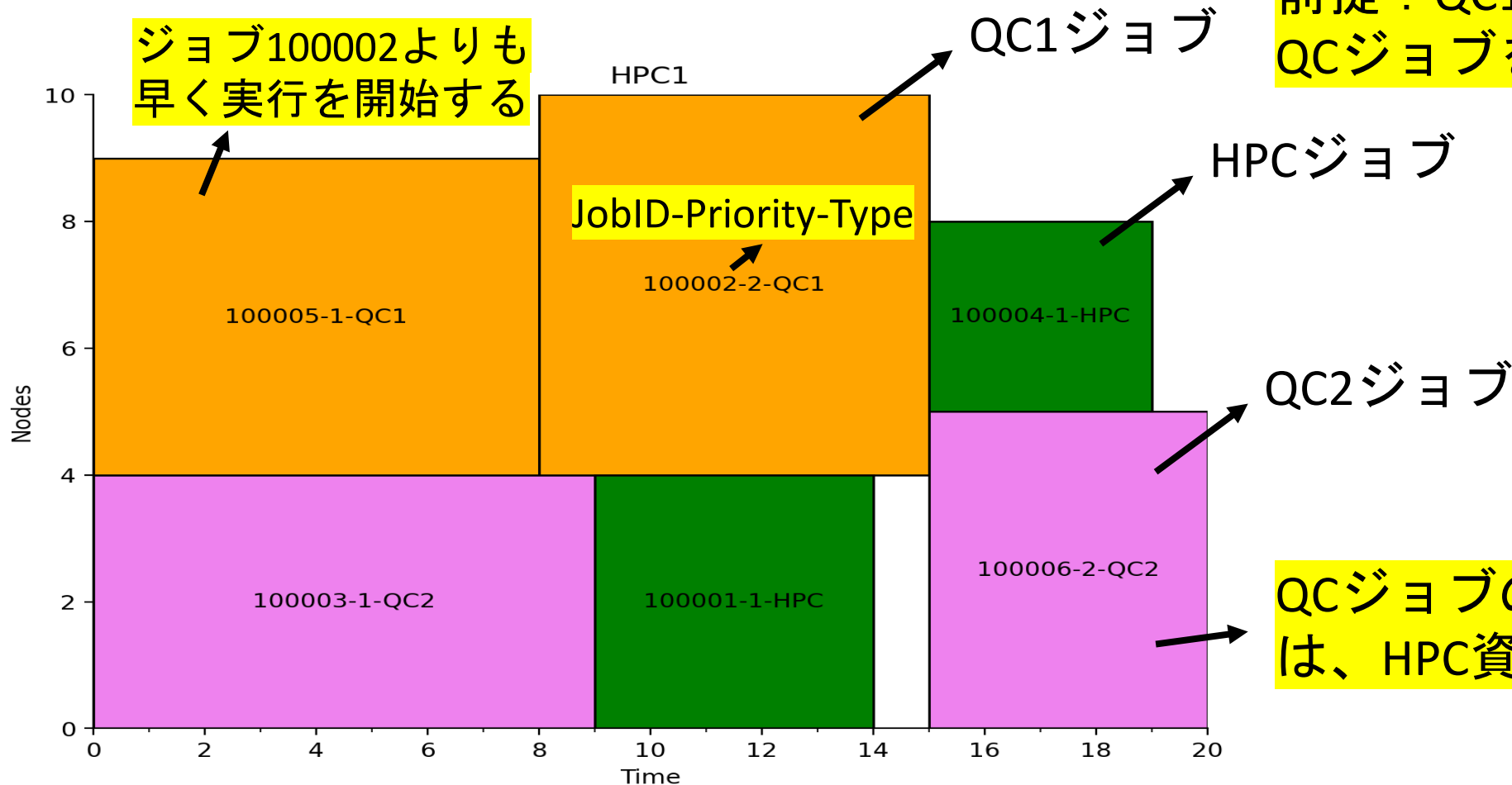
Todo : ジョブの状態遷移は実機での動作と一致させるようにする

ジョブの再スケジューリング

再スケジューリングなしの場合、止められたジョブを復活させる時点が大幅に遅れる恐れがある

- 以下の場合に、ある時点以降のジョブを再スケジュールすることを考える
 - 一時停止のHPCジョブをできるだけ早く再開するために、他HPCジョブを再スケジュールする
 - 高優先度のQCジョブをスケジュールする時、低優先度のQCジョブを再スケジュールする
 - スケジュール済み（QUEUED状態）のジョブを削除する場合、他ジョブを再スケジュールする

ユースケース1：1つのHPC（HPC1）と、2つのQC（QC1、QC2）

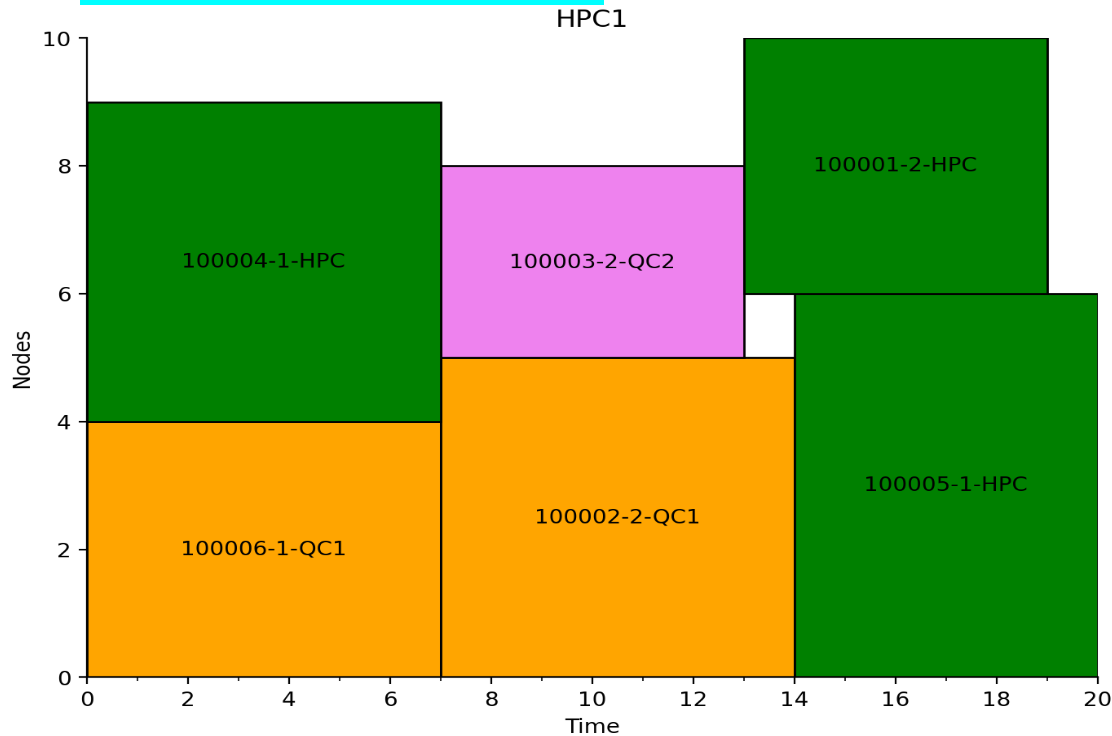


前提：QC1かQC2は同一時点でQCジョブを1つだけ実行する

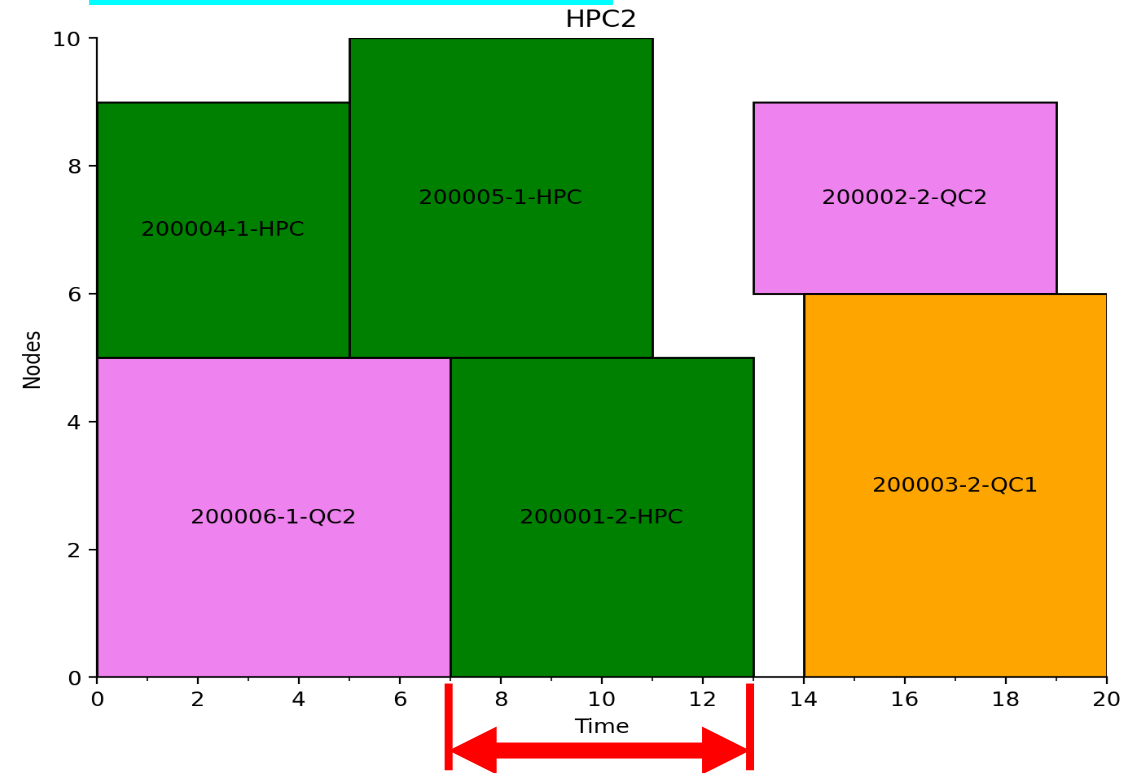
QCジョブのスケジューリングは、HPC資源の確保後となる

ユースケース2：2つのHPC（HPC1、HPC2）と、2つのQC（QC1、QC2）

HPC1スケジュール

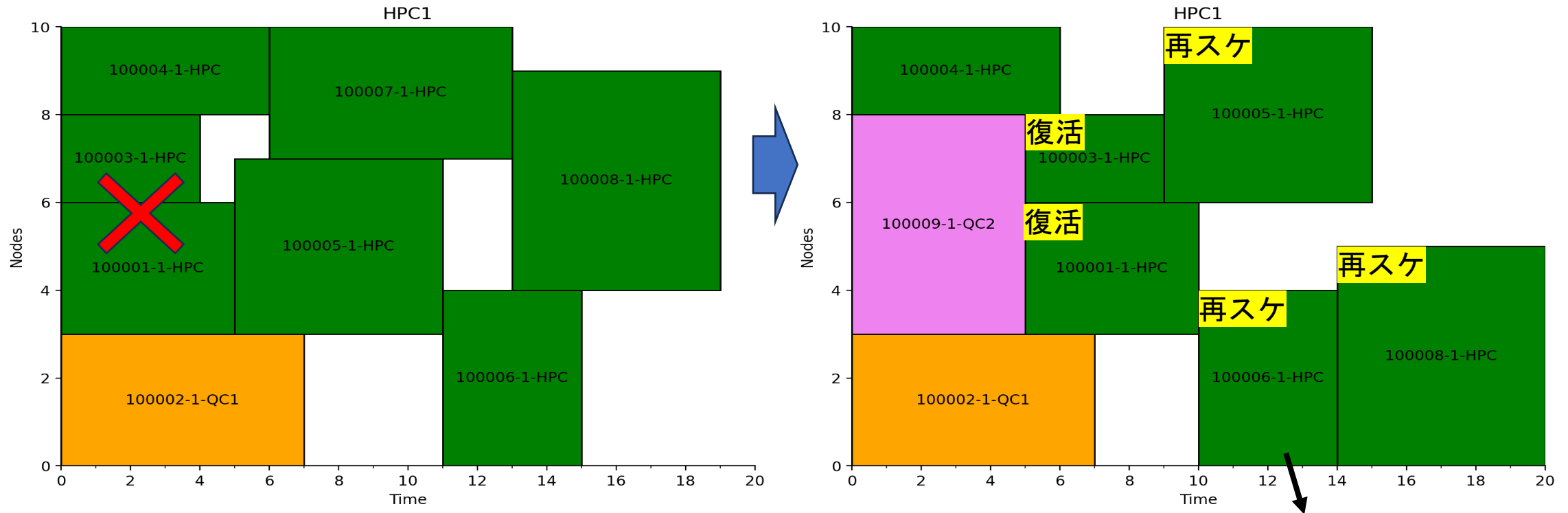


HPC2スケジュール



QC資源占有中のため、QCジョブのスケジューリングは不可

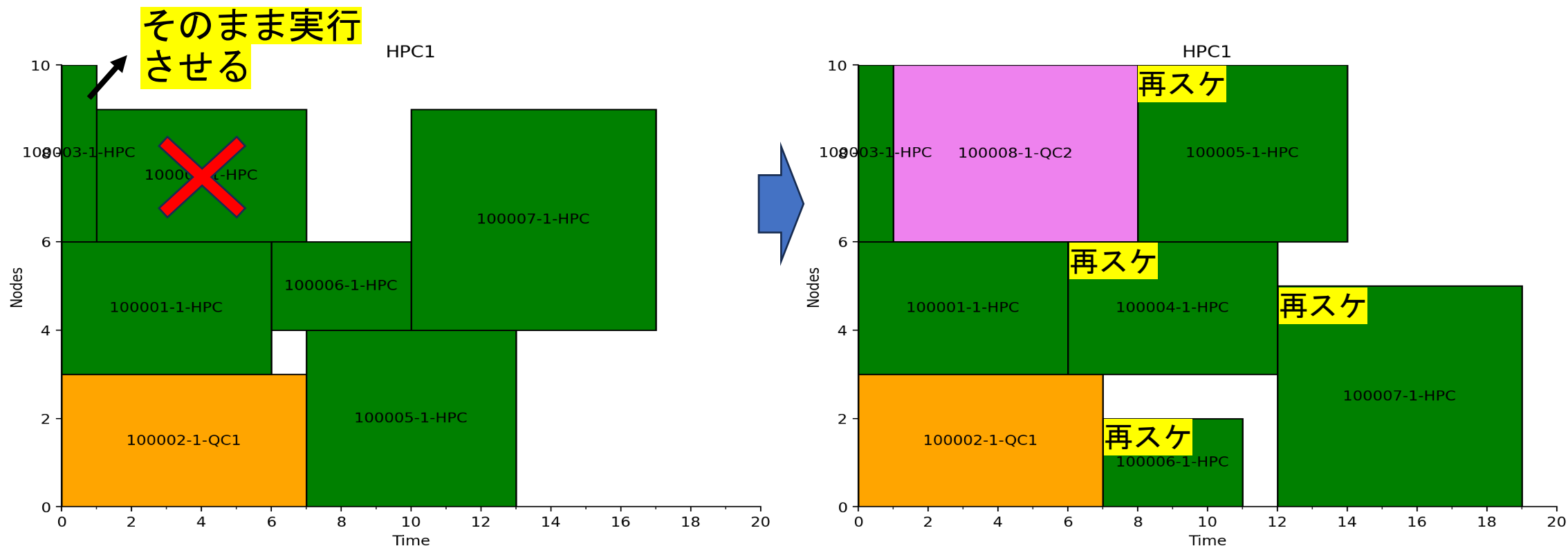
ユースケース3：追加で投入されたQCジョブを実行するために実行中のHPCジョブを一時停止する



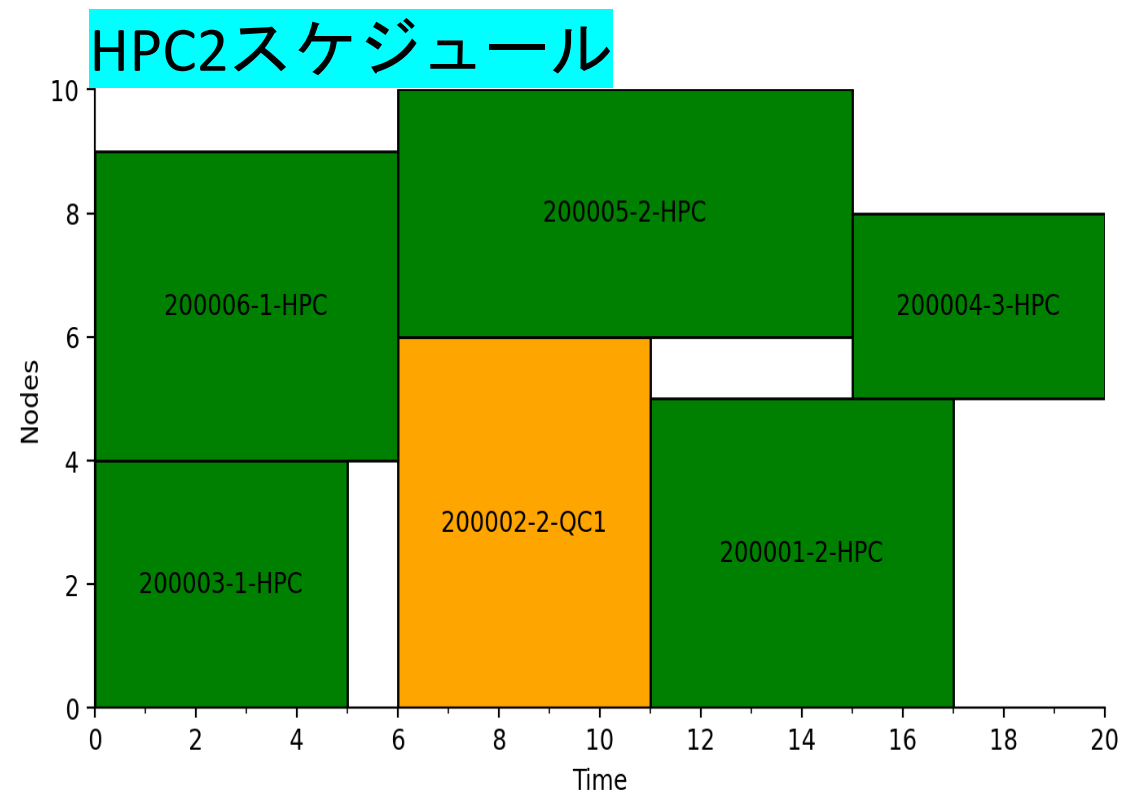
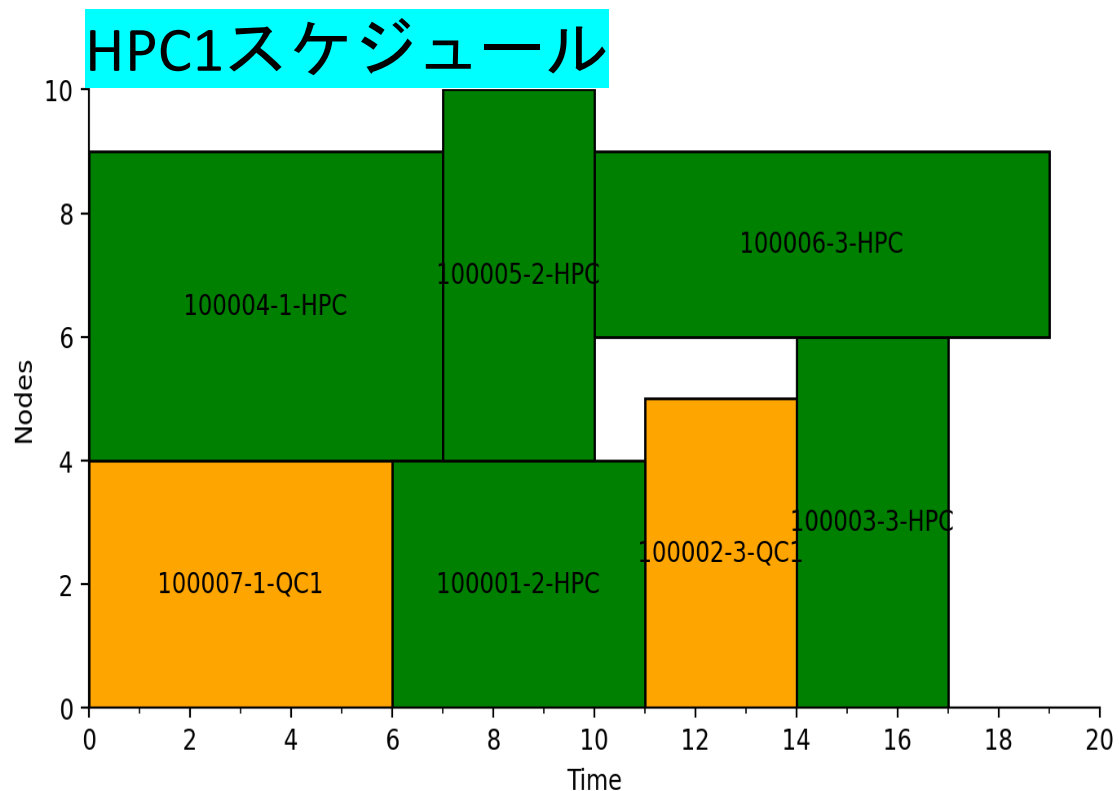
HPCジョブはどっちを止めるのが重要 (Todo)

再スケジューリング結果、前倒しで実行を開始するケースもある

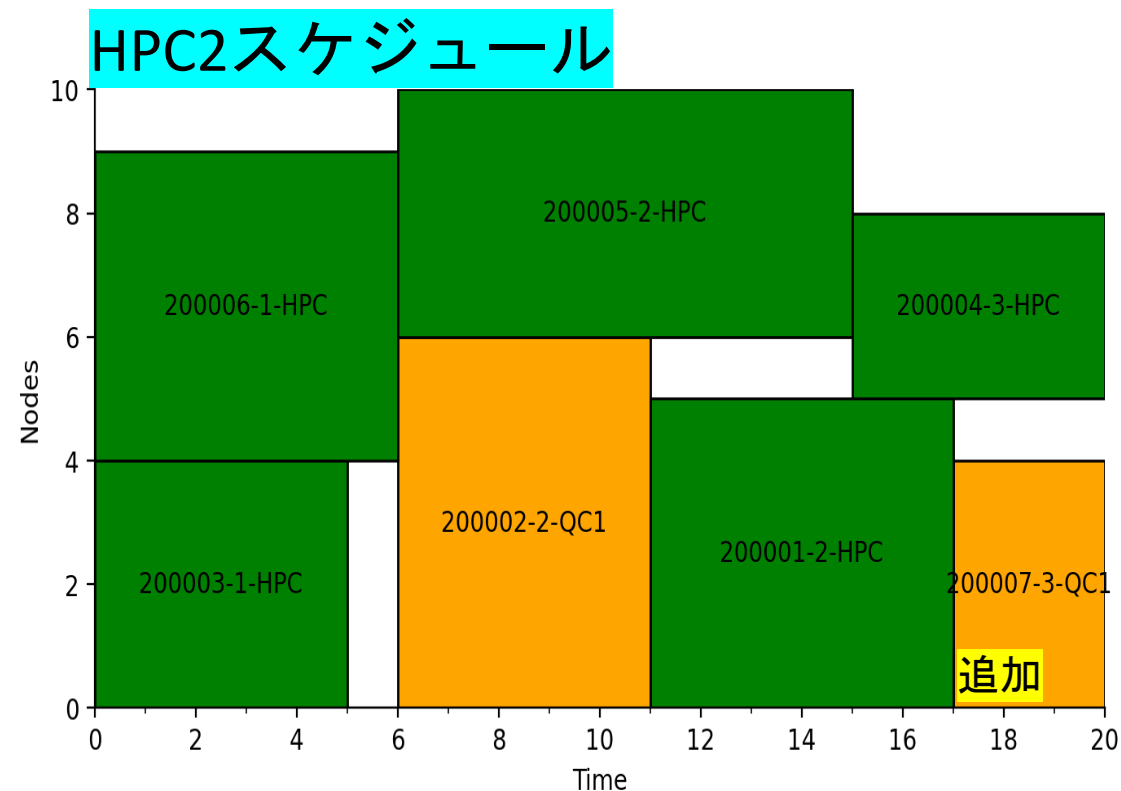
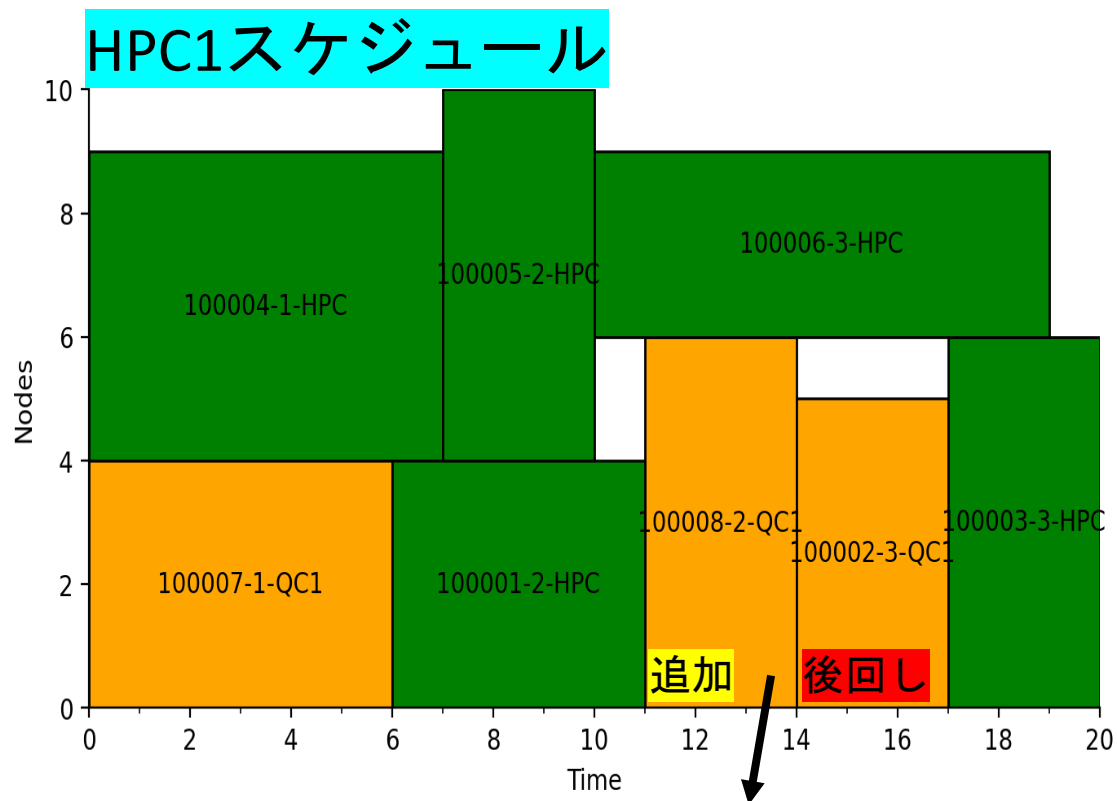
ユースケース4：終了直前のHPCジョブを止めず、後のHPCジョブを再スケジュールする



ユースケース5：QCジョブの追加（前）

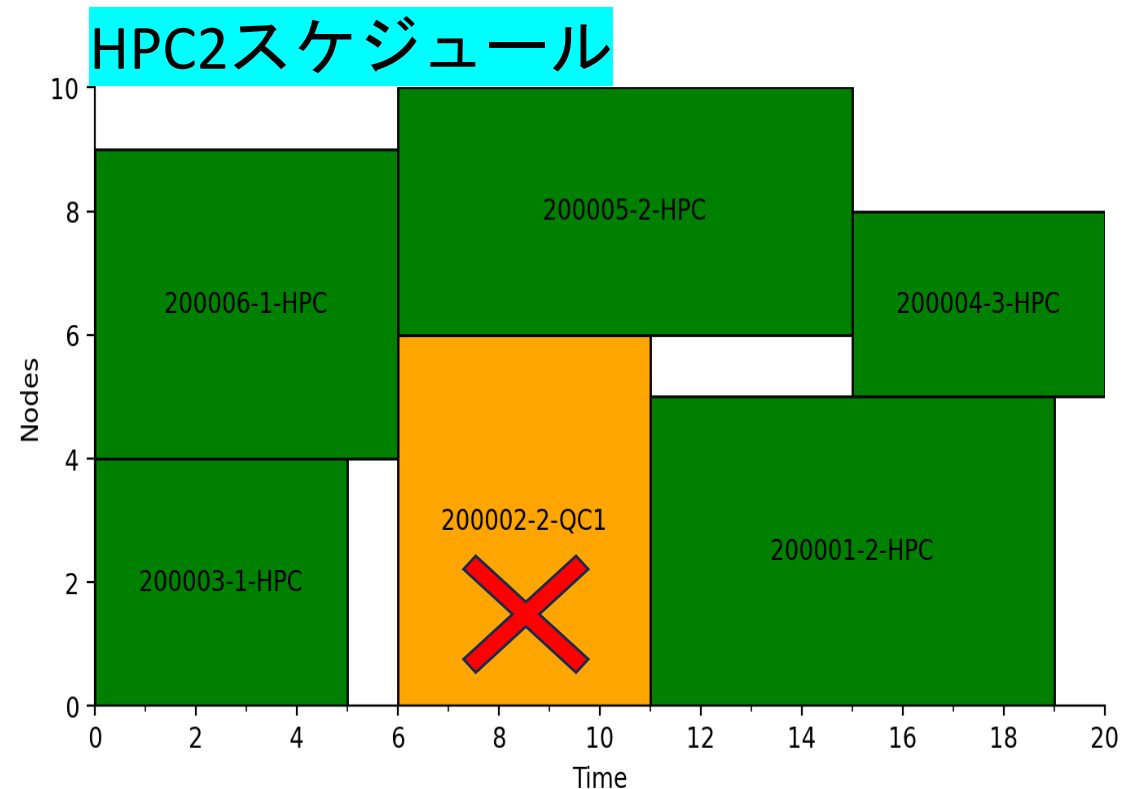
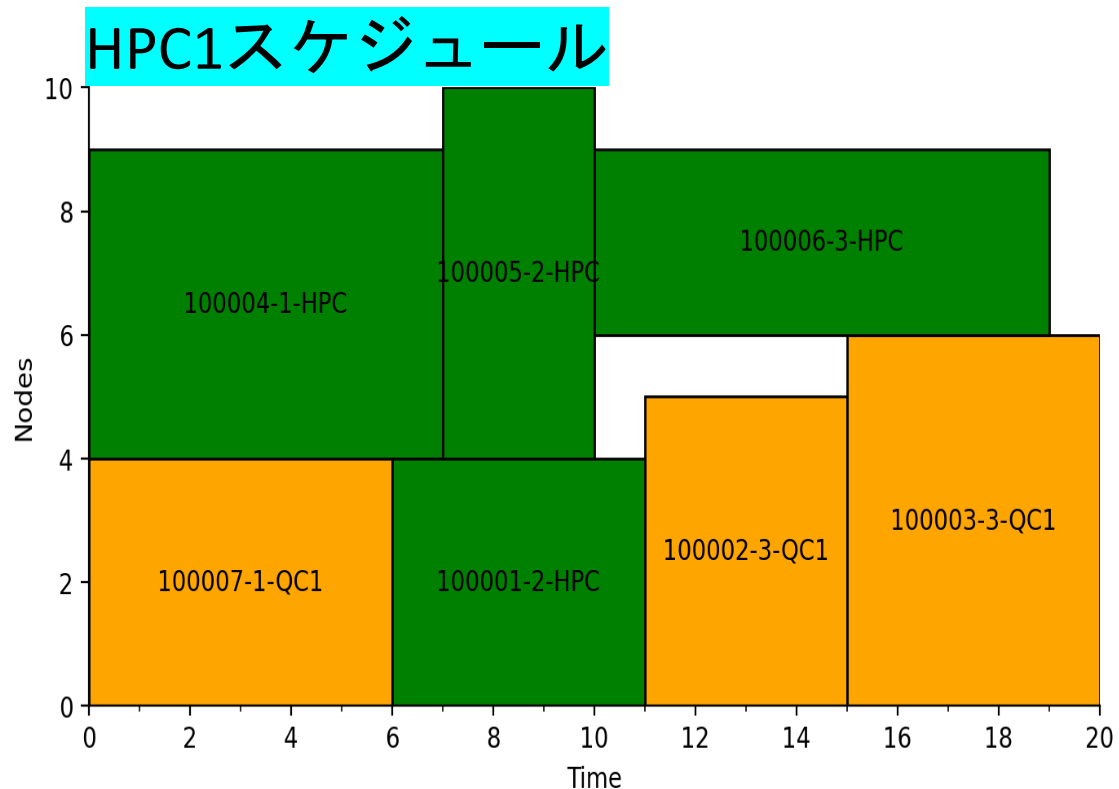


ユースケース5：QCジョブの追加（後）



優先度はジョブ100002よりも高いので、先にスケジュールする

ユースケース6：QCジョブの削除（前）



ユースケース6：QCジョブの削除（後）

