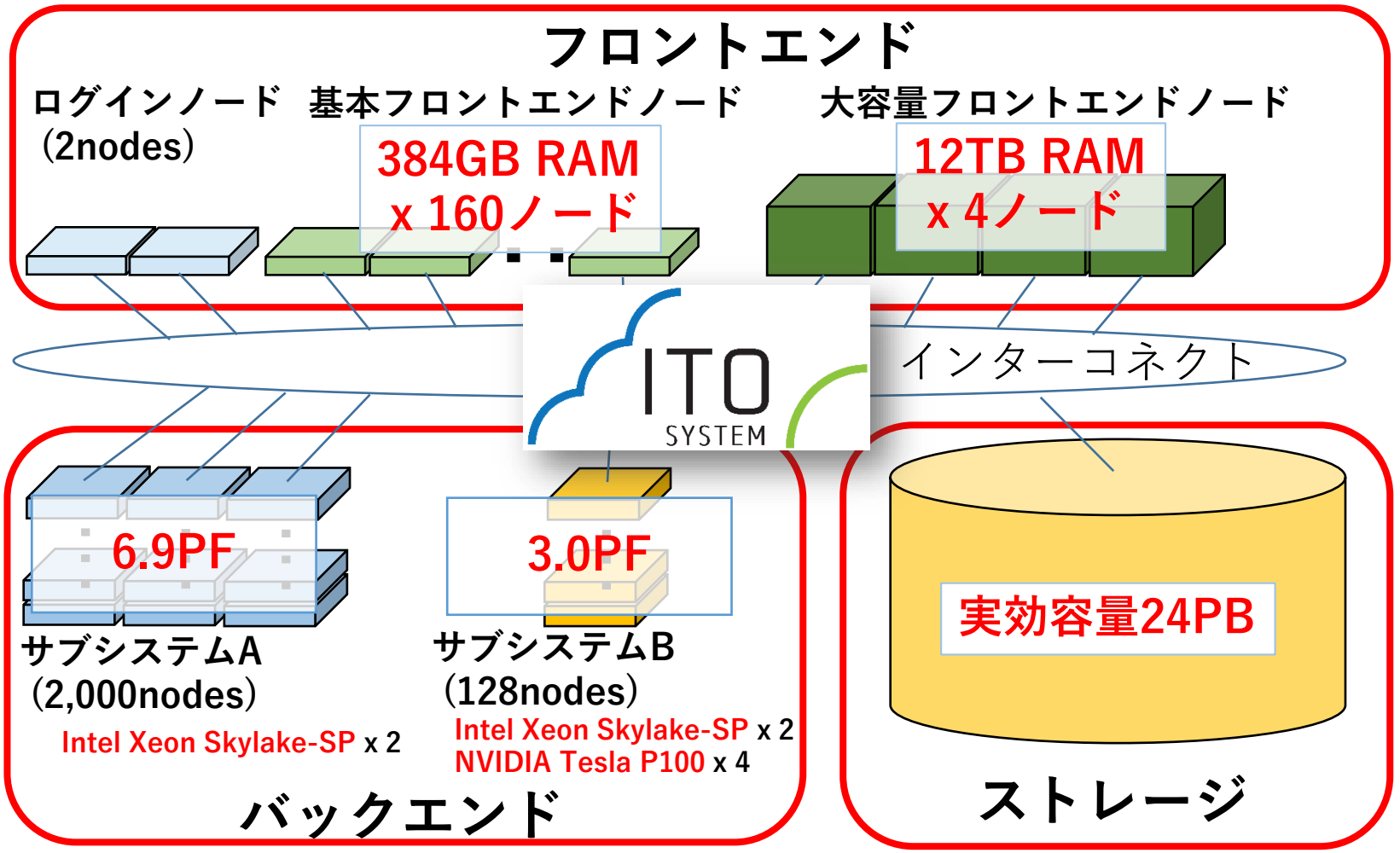
九州大学スーパーコンピュータシステム ITO におけるジョブスケジューリングの課題

2018年 9月 18日

PCクラスタコンソーシアム
南里 豪志（九州大学）

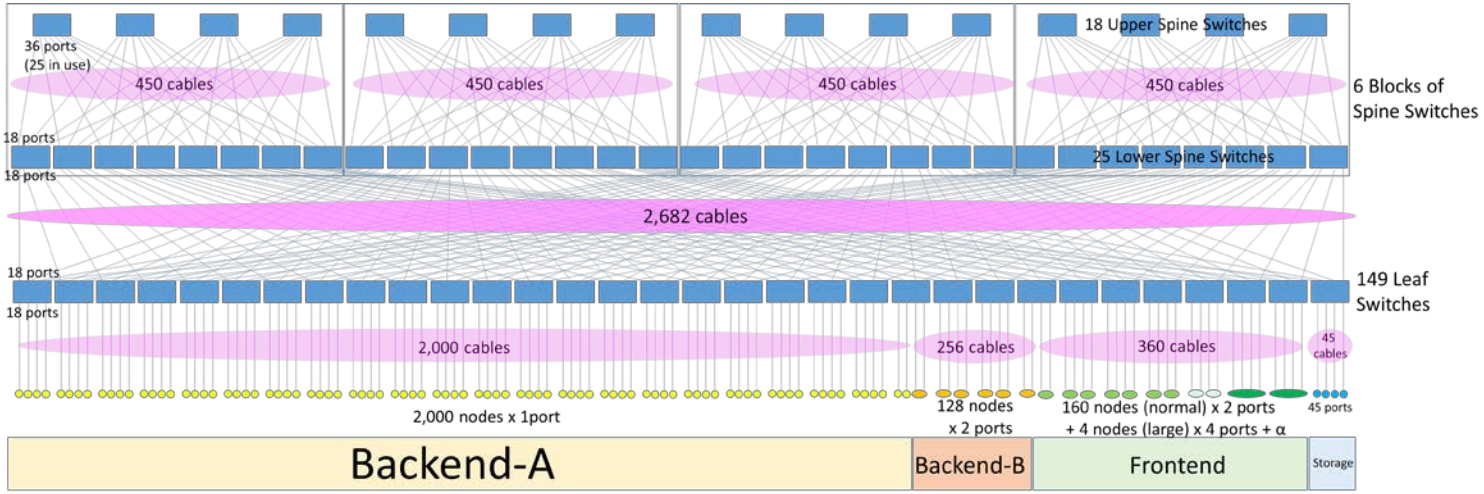
スーパーコンピュータシステム ITOの概要



ジョブスケジューラから見たハードウェアの特徴

- インタラクティヴジョブが不要
  - 基本 / 大容量フロントエンド: 対話型、Web予約
  - サブシステム A / B: 非対話型、バッチ

- ノード配置による通信性能への影響は軽微(のはず)
  - Full Bisection Bandwidth の Fat Treeトポロジー



基本的なバッチシステム運用方針

- グループ単位のジョブ管理
  - グループ内のアカウントで
    使用資源量
    優先度
    等を共有
- 富士通製 Technical Computing Suiteによる運用
- **定額利用負担金**
  - 非･従量課金

利用負担金表（バッチ利用分、月額）

利用タイプ	サブシステムA	サブシステムB
共有タイプ	4ノード当たり 2,960円	1ノード当たり 2,100円
ノード固定タイプ	4ノード当たり 23,600円	1ノード当たり 17,000円

ノード固定タイプ： 占有的な利用タイプ。詳細は後述。

利用負担金定額制

- 「同時利用可能資源量」に応じた課金

- 例）サブシステム A共有タイプ 4ノード（144コア）
  - 合計 144コアまで、同時に利用可能
    - 144コアジョブ x 1
    - 72コアジョブ x 2
    - 36コアジョブ x 2 + 72コアジョブ x 1
      etc.

利用負担金定額制の利点と欠点

- 利点：使用量によらず、負担金額が一定
  - 利用グループの予算計画が容易

- 欠点：基本的に、使ったもの勝ち

▪「利便性」、「稼働率」を損なわずに
▪「公平性」を実現する

ようなスケジューリング手法が
（従量性よりも強く）求められる

ITOにおける主なスケジューリング手法

- フェアシェアによる優先度管理
  - 公平性
- ノード固定タイプの提供
  - 利便性
- 短時間ジョブキューの提供
- バックフィルスケジューリングの適用
  - 稼働率
- 実行開始時刻の保証
  - 利便性

フェアシェアによる優先度管理

- 利用グループごとに、
  フェアシェア値（＝ 優先度）
  を設定
  - 原則、全グループで初期値は同じ

- 使用した資源量（コア時間積）に応じてフェアシェア値減算
  ⇒ **各グループへの資源割り当て均等化**

- ただし、月ごとに初期値に回復
  ⇒ **高利用グループが過度に待たされる状況を回避**

ノード固定タイプの提供

- ノードを特定のグループに優先割り当て
  ＝　混雑状況によらず、
  「**1時間以内**」
  に、そのグループのジョブが流れ始めることを保証

- 利用負担金：**共有タイプの約 8倍**
  - 共有タイプでは、平均 8グループでノードを共有している、と想定

- 各グループへの割り当てノード数調整
  - 毎年度末に次年度分の利用希望受付
  - **過去の利用実績をもとに調整**

短時間ジョブキューの提供

- ノード固定タイプのノードの稼働率：**低い**
  - 当該グループの利用状況に依存

- 1時間以内に終了する**共有タイプ**ジョブキューを用意

- **ノード固定タイプの空きノードを有効活用**
  - 実行終了時に、ノード固定タイプのジョブが無ければ、
    続けて短時間ジョブ実行

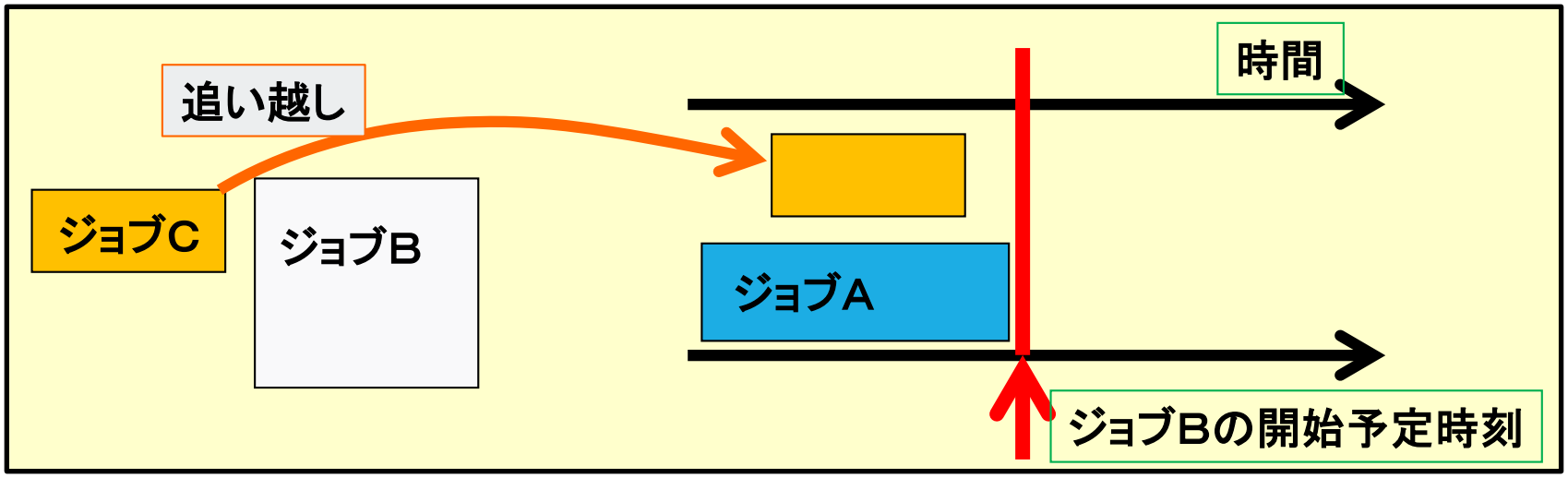
現在のジョブキュー（抜粋）

キュー名	ノード数	最大計算時間
ito-q-dbg	1/4	1時間
ito-m-dbg	16	1時間
ito-xxl-dbg	256	1時間
ito-q	1/4	168時間
ito-m	16	24時間
ito-xxl	256	6時間

短時間ジョブキュー

バックフィルスケジューリング

- 他のジョブを遅らせない範囲で、ジョブの追い越しを認める
  - 例）ジョブCを Bより先に実行しても、Bの実行開始時刻に影響なし



ジョブ開始予想時刻の変動

- ジョブ投入時：

```
JOB_ID      JOB_NAME    MD ST  USER      START_DATE
1520147     test-om3.0  NM QUE k70043a   (09/12 20:30)
```

ジョブ開始予想時刻

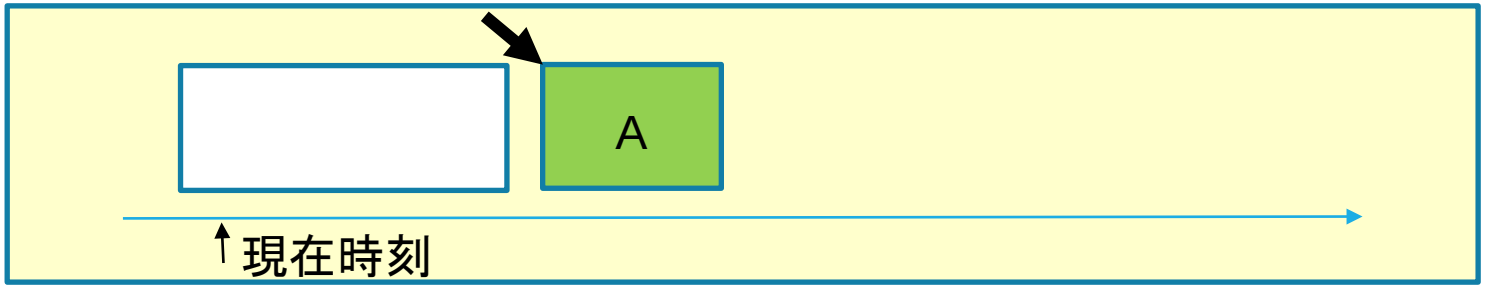
- 1時間後

```
JOB_ID      JOB_NAME    MD ST  USER      START_DATE
1520147     test-om3.0  NM QUE k70043a   (09/12 21:30)
```

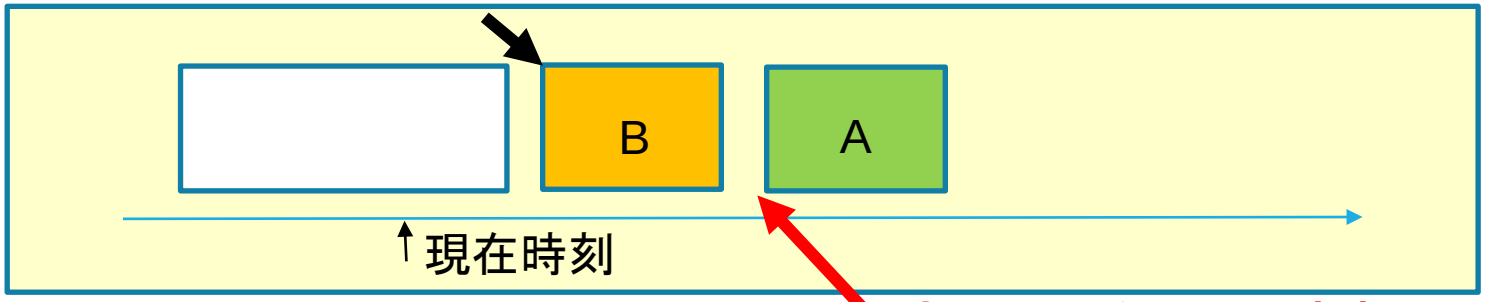
変動の原因

- フェアシェアとバックフィルの併用による
- 例）

グループA（フェアシェア値 低）がジョブ投入



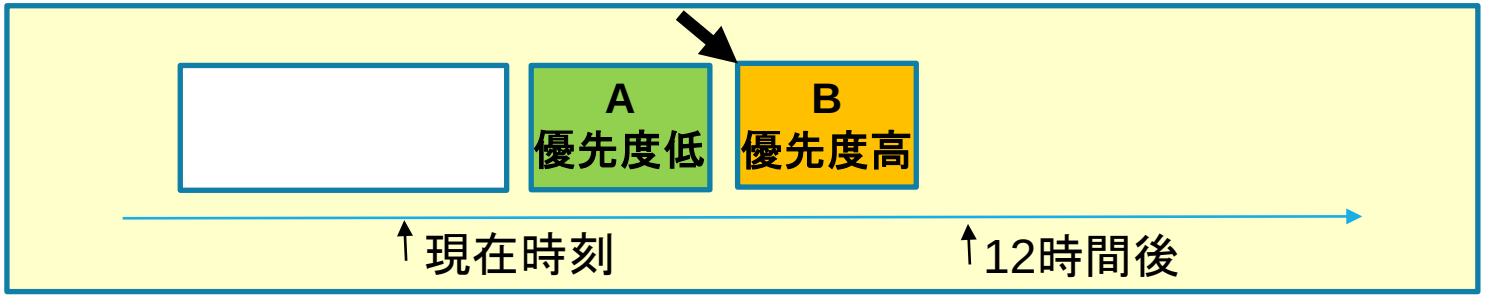
グループB（フェアシェア値 高）がジョブ投入



対応策

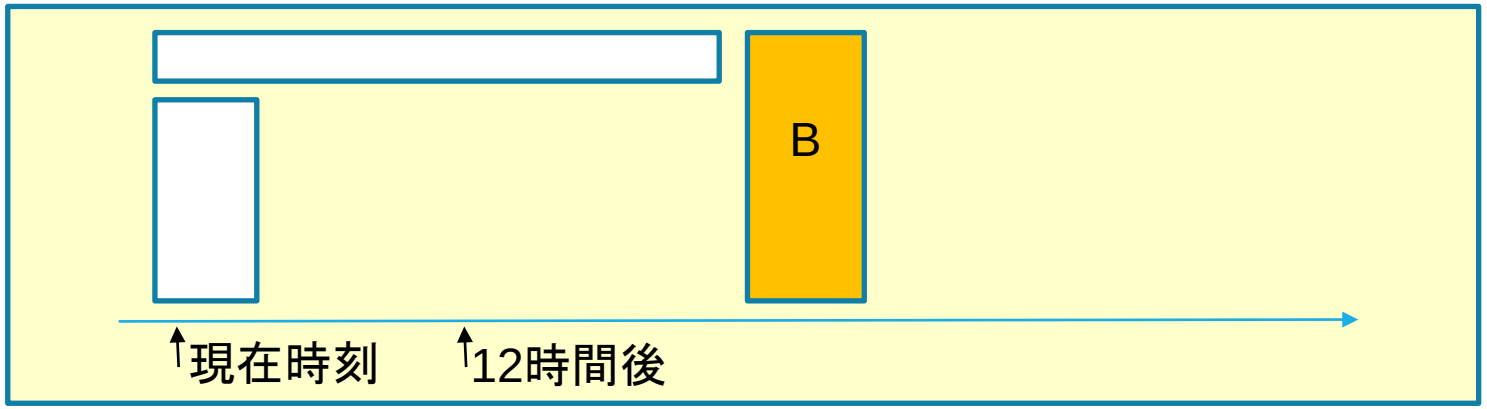
- 「いっそのこと、開始予測時刻を表示しない」
  という手もあるが。。。

- 開始時刻保証：
  開始予測時刻が現在時刻の**12時間以内**となったジョブは、
  フェアシェア値によらず遅らせない

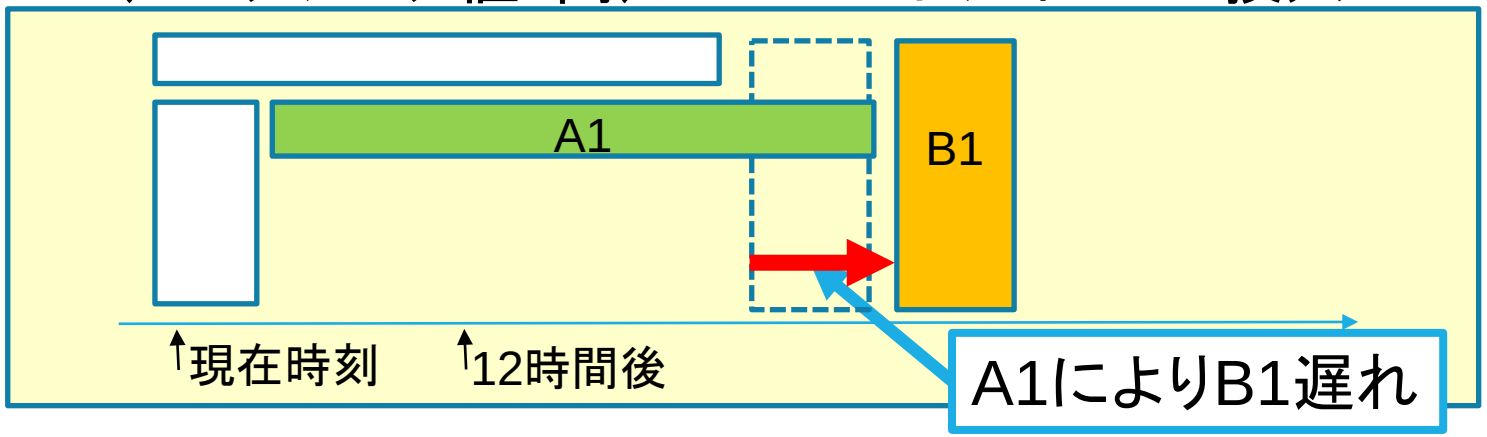


現在の課題：大規模ジョブの想定外の遅れ

- 例）フェアシェア値　A > B > C で、64ノードのシステム利用
  1. B（フェアシェア値：中）が 64ノードジョブ B1 投入



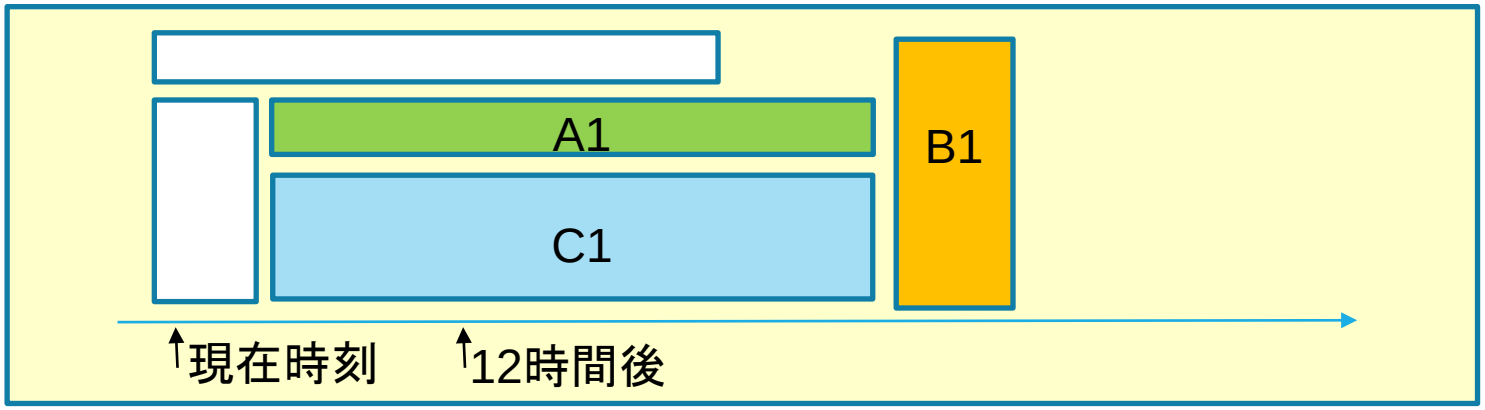
  2. A（フェアシェア値：高）が 16ノードジョブ A1 投入



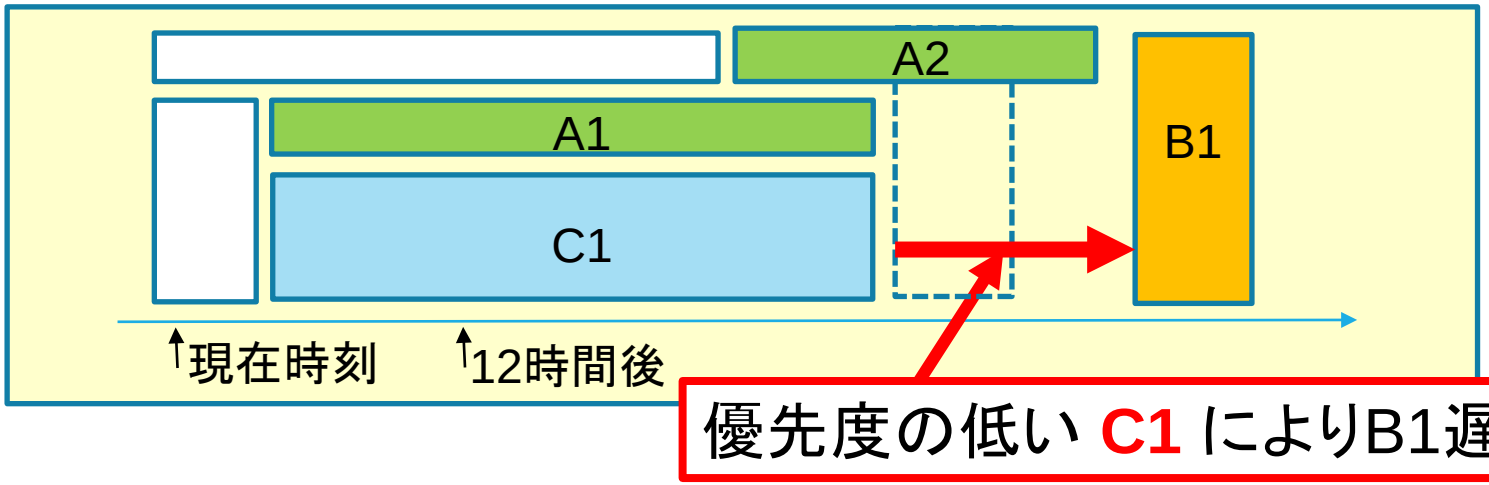
つづく

大規模ジョブの想定外の遅れ(つづき)

3. C(フェアシェア値:低)が 32ノードジョブ C1 投入

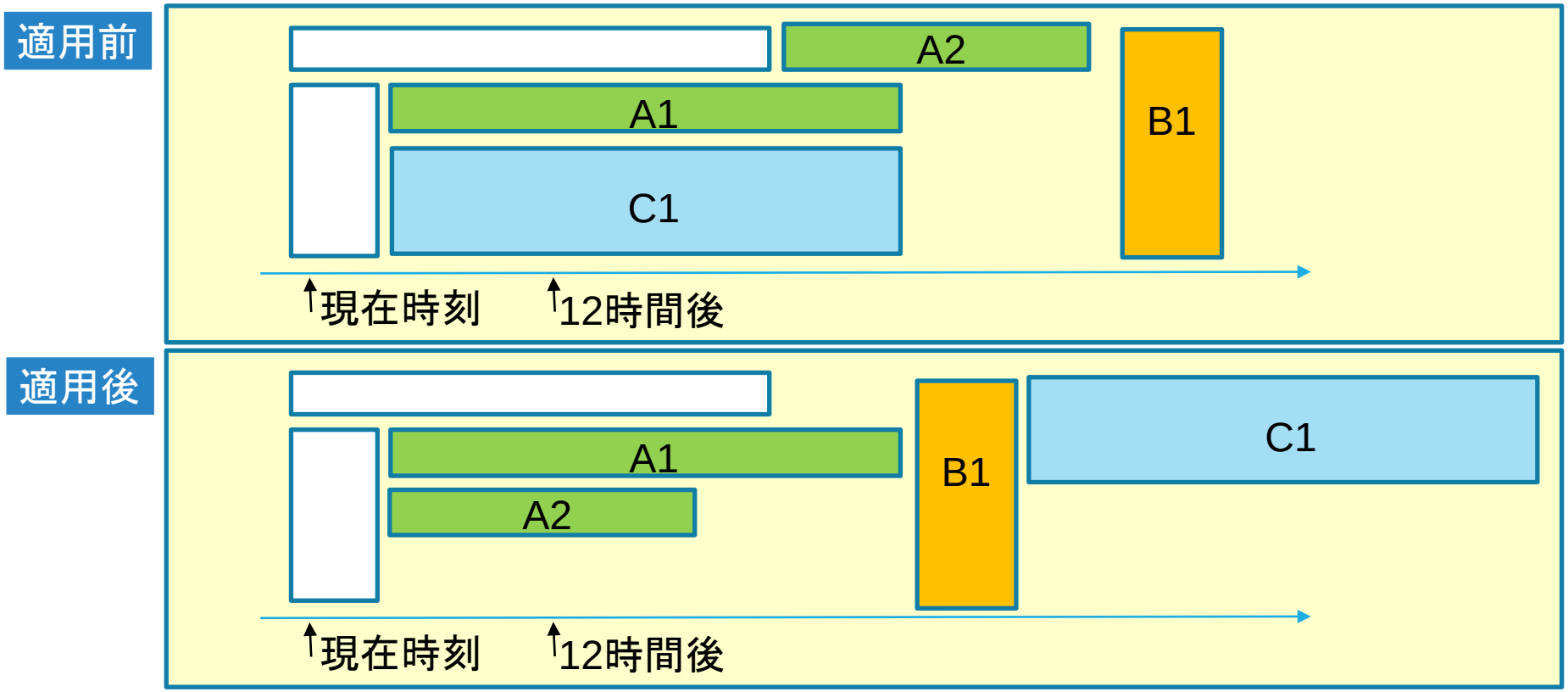


4. A(フェアシェア値:高)が 16ノードジョブ A2 投入



対策：検討中

- 案1）小規模（32ノード以下）のジョブは12時間以内でも
  フェアシェア値に応じて実行開始を遅らせる



- 案2）大規模（64ノード以上）のジョブの優先度を高くする
  - 小規模ジョブの実行が過度に阻害されないように、注意が必要

まとめと今後の課題

- ITOのジョブスケジューリング
  - 定額利用制で「公平性」、「利便性」、「稼働率」の実現方法を模索中：
    フェアシェア、ノード固定タイプ、短時間ジョブキュー、バックフィル、
    実行開始時刻の保証
  - フェアシェア、バックフィル、実行開始時刻保証の併用による問題：
    大規模ジョブの想定外の遅れ

- 今後の課題： 新しいスケジューリング技術の要求
  - 省電力運用： 電力キャッピング、デッドラインスケジューリング、等
  - コンテナ利用： Singularity等
  - パブリッククラウドとの連携

  - 詳細は、パネルで