



SCore Lightの概要

- ◆堀 敦史(Allenia Software)
- ◆長谷川 篤史(NEC情報システムズ)
- ◆竹岡 尚三(AXE)
- ◆石川 裕(東京大学)

現状認識: SCore vs. Beowulf

	SCore	Beowulf
実績	大規模PCクラスタ	小中規模PCクラスタ
PC利用形態	専用マシン想定	別用途併用可能
インストール	難(要ネットワーク知識)	簡単
機器構成変更	難(要SCore知識)	簡単
保守性	中(利用勝手に難)	考慮されていない

MPI通信ライブラリ (MPICH-SCore)
High Performance Multi-Users Environment
PM for Cluster Computing
Network

MPI通信ライブラリ (MPICH-P4, LAM/MPI)
LAN Environment
TCP/IP for Internet
Network

- Linuxとオープンソースの 組み合わせで作られたクラスタを一般にBeowulf型クラスタと呼ばれる
- 1990年半ば、JPL/CaltechのThomas Sterling率いるBeowulfプロジェクトに由来する

ユーザビリティ向上

	SCore	
PC利用形態	SCoreのディスクレス化	
インストール	SCore Lightインストールツール	
構成変更	SCore自動構成ツール	
保守性	SCore動的再構成機能	

計算ノード: PCクラスタ構成要素のうち、計算を担当するPC群を計算ノードと呼ぶ

計算ノード保守時に、
全体のシステムを停止せずに
他のコンピュータを使用

普段はWindowsシステム
夜間はSCoreクラスタ構
成要素として使用

ディスクレスSCore

簡単
インストール

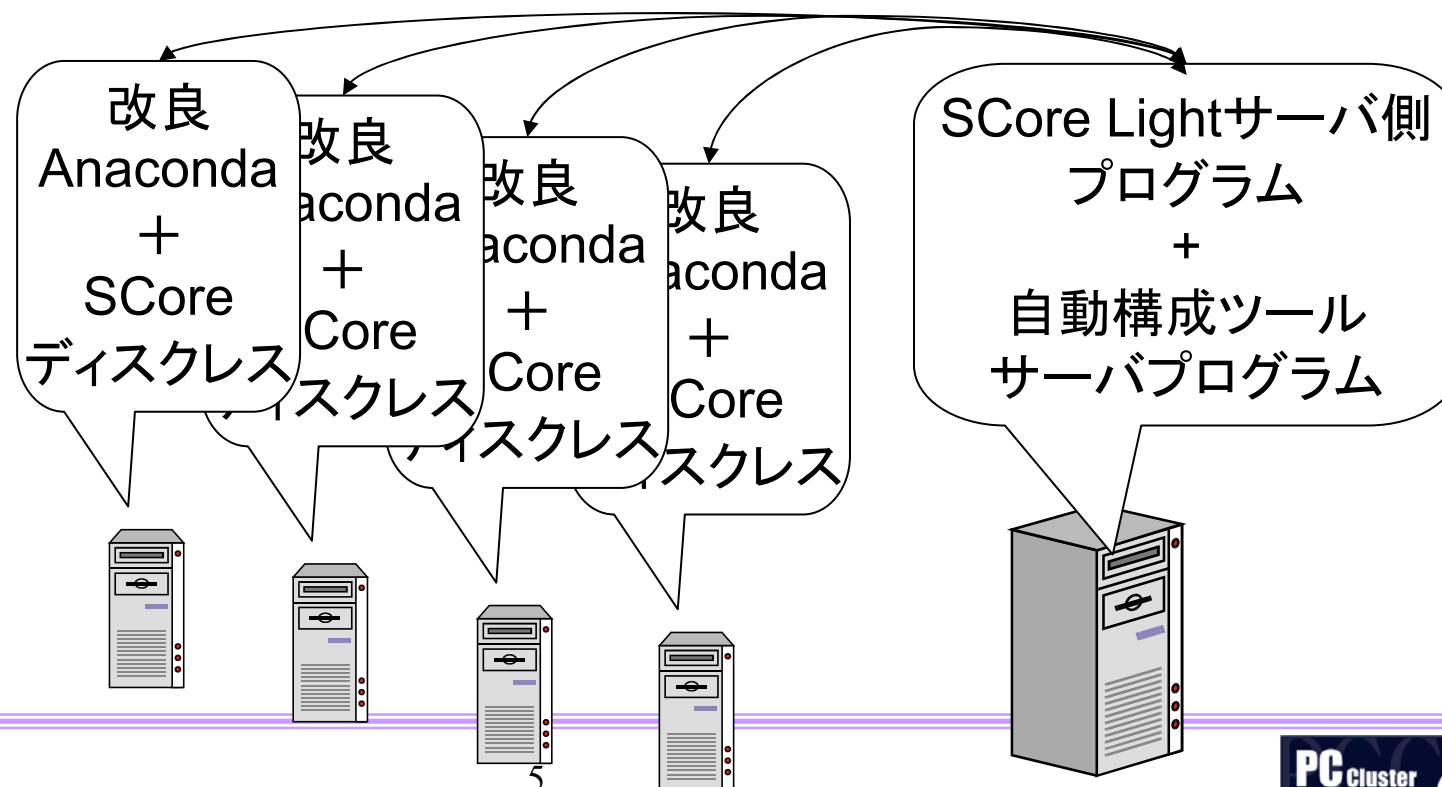
新たに計算ノードが加わる

この後の発表

- ◆ SCore Lightインストールツール
 - ◆ 石川 裕(東京大学)
- ◆ SCoreのディスクレス機能
 - ◆ 竹岡 尚三(AXE)
- ◆ SCore自動構成ツール
 - ◆ 長谷川 篤史(NEC情報システムズ)
- ◆ SCore動的再構成機能
 - ◆ 堀 敦史(Allenia Software)

SCore Lightインストールシヨンツール

- ◆ 3タイプのインストールシヨン
 - ◆ ディスクレス環境
 - ◆ Linuxインストールも含めたディスク環境
 - ◆ 既存Linux環境



ソフトウェア構成

SCoreのディスクリス化

ファイルシステム構造変更
+ SCore動的再構成機能

SCore動的再構成機能

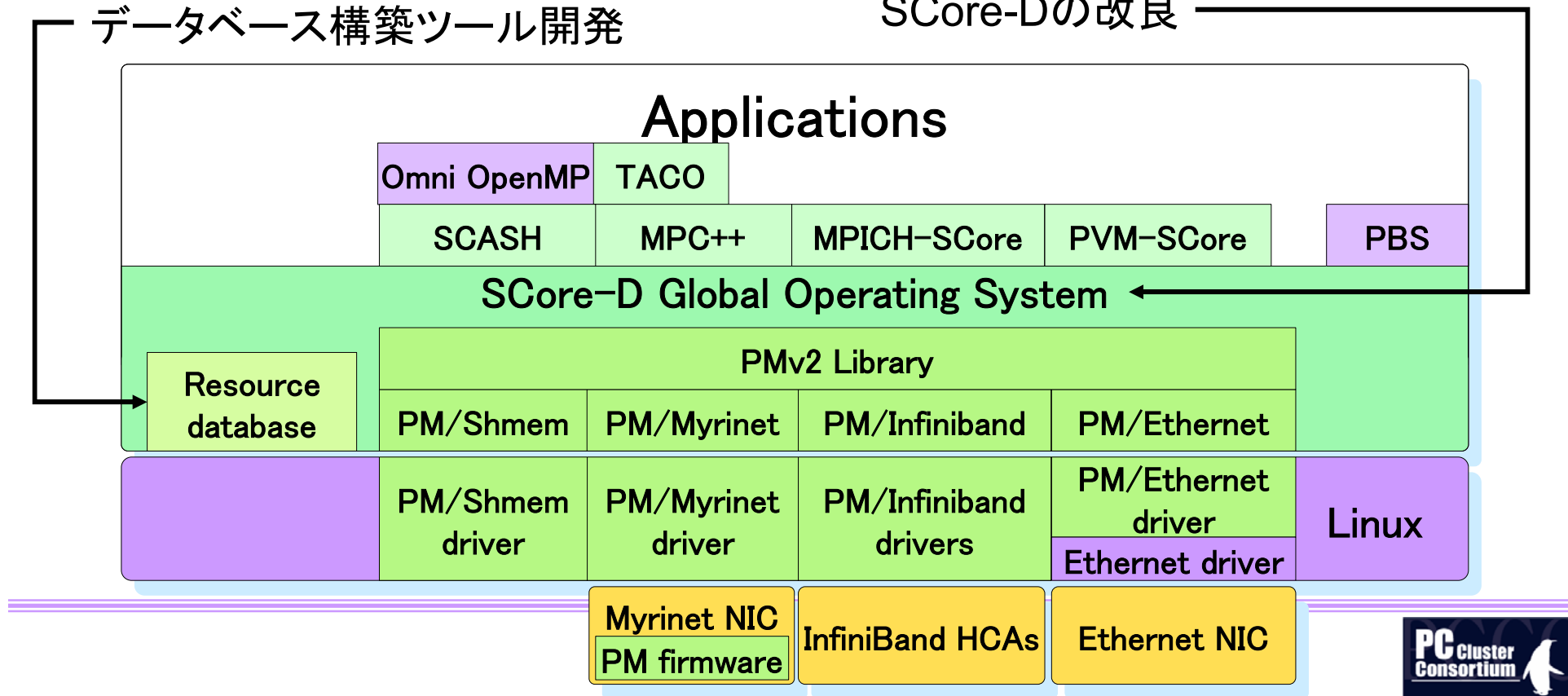
SCore Light インストールツール

改良Anaconda + SCoreディスクリス
+ ツール

SCore動的再構成機能

データベース構築ツール開発

SCore-Dの改良



*SCore*のディスクレス機能(1)

- ◆ 手軽にSCoreを試せる環境の提供
- ◆ 管理コストの軽減
- ◆ 現状
 - ◆ HDDにLinux+ScoreをインストールしなければSCoreを動かすことができない
 - ◆ 大学の研究室レベルですぐに試すことができない
 - ◆ 新規ユーザ獲得の障壁

*SCore*のディスクレス機能(2)

- ◆ Linuxディスクレス・ブート
 - ◆ ネットワークからの起動
 - ◆ CDROMからの起動
 - ◆ HDDはまったく使用しなくとも動作可能
 - ◆ NFSサーバのディスク上に各ノードのルート・ディレクトリを置ける
- ◆ 各ノードの自動コンフィギュレーション
 - ◆ NFSサーバ上のルート・ディレクトリを各ノードが適切に認識してマウント
 - ◆ 自動的にSCore計算ノードとなる

*SCore*のディスクレス機能(3)

- ◆ RPMパッケージ・インストール補助ツールの提供
- ◆ ディスクレス・ノードはファイルの多くを共有
→ 不用意にRPMをインストールすると、
共通情報が不適切に書き換えられる
- ◆ RPMパッケージを適切に変更する作業を容易にする
ツールを提供
- ◆ RPMをディスクレス・システム向けに変更して、以降そ
れを使用可能

SCore自動構成ツール(1)

◆ SCoreの現状

[機能]

- ◆ MPI, PVM, OpenMPを提供
- ◆ Checkpoint/restart
- ◆ バッチシステム(PBS)、Multi User環境

[導入状況]

- ◆ 産総研、理研等の大規模システムや、奈良先端大等の中規模システムを中心に普及
- ◆ 大学、研究機関のへの導入が中心
→ 小規模システムへの普及が課題

SCore自動構成ツール(2)

- ◆ 普及を促進するための課題
 - ◆ 大規模システム(数百ノード～)
 - ◆ 専任のSEによる手厚いサポート
 - ◆ 中規模システム(数十ノード～百数十ノード)
 - ◆ RemoteでSEがサポート
 - ◆ 小規模システム(数ノード)
 - ◆ SEのサポートなし
 - ◆ ユーザ自身が管理
 - ◆ 専門家がないので、導入時の負担が大きい/何をすればいいのかわからない
 - ◆ 設定方法がわからない
 - ◆ 障害時の対処方法がわからない

ソースを読めば直に判ること
でも、その時間が無い

SCore自動構成ツール(3)

◆ 小規模システムでは何が必要か？

- ◆ 導入時の負担大
 - ◆ インストール手順書の整備
 - ◆ クラスタの構築に特化したインストーラ
- ◆ 設定方法がわからない
 - ◆ 初心者向けのドキュメントの整備
 - ◆ 設定の自動化

→「SCore自動構成ツール」の開発

- ◆ ユーザが何も設定しなくても、クラスタとして使える環境の提供
- ◆ 計算ノードを追加して電源を入れるだけで、設定を自動的に構成

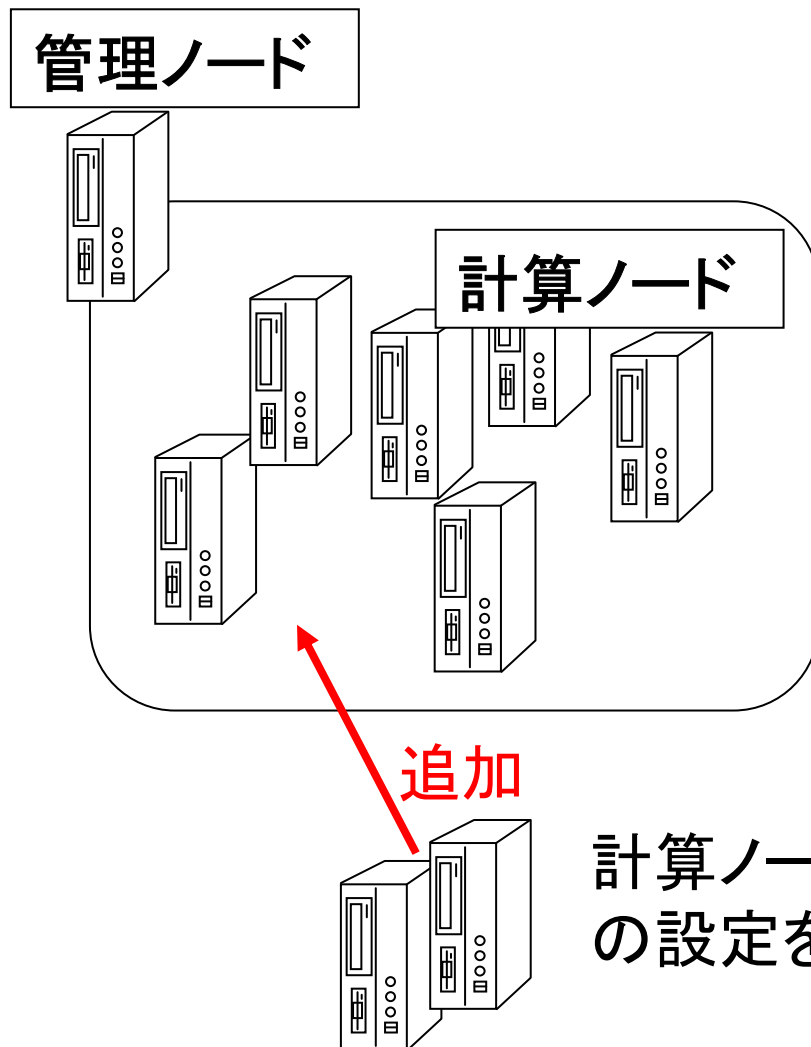
ユーザはSCoreの仕組みを知らなくても困らない

使用する事だけに専念できる

- ◆ 障害時の対処方法がわからない
 - ◆ FAQ、TIPSの整備？

SCore自動構成ツール(4)

◆ 機能の概要



Network上の計算ノードの
情報を収集して、自動的に、
クラスタの設定を生成

計算ノードが追加されたら、クラスタ
の設定を、自動的に更新

SCore自動構成ツール(5)

◆「自動構成ツール」は何をすべきか？

◆ 一般的な設定

- ◆ Network設定(DHCP,DNS?)
- ◆ ユーザアカウント(NIS,LDAP?)
- ◆ 共有ファイルシステム(NFS?)
- ◆ Networkサービス

◆ SCoreの設定

- ◆ 管理ノードの設定
 - ◆ リソース管理(計算ノードの情報)
- ◆ 計算ノードの設定
 - ◆ HWリソースの情報管理
 - ◆ デバイスの設定

サイトポリシーの問題

ユーザは使用できるが、
サイト管理者でないと設定/変更できない

勝手に設定すると問題!!

SCoreで使用する設定を、
自動的に生成してシステムに設定。

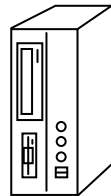
SCore自動構成ツール(6)

◆ 自動設定ツールの構造

ユーザ

- 設定更新後、管理ノードから、クラスタを使用

管理ノード



- クラスタ構成/設定ファイルを更新

HW構成情報の登録

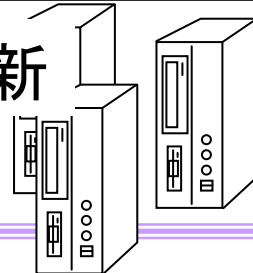
設定更新要求

計算ノード

- 計算ノードの設定を更新

- 管理ノードを検索

- 計算機の構成情報を収集



*SCore*自動構成ツール(6)

◆ まとめ

- ◆ ネットワークに繋げて、PCを立ち上げれば設定を更新、クラスタとして使用できる(パッケージのinstallは必要)

- ノードの追加/削除が容易に

- 故障したら、PC毎交換/sendbackで修理

- ◆ 設定方法を覚える必要がない

- ユーザは、使うことに専念できる

- SEがいなくてもok(?)

*SCore*動的再構成 (1)

- ◆ さんからの(執拗な)要望
「マルチユーザ運用で、ノードが故障しても、故障ノード以外でのジョブを実行を継続したい」
- ◆ 現状：一つでもノードが停止すると、クラスタ全体が停止してしまう

SCore動的再構成 (2)

- ◆ SCore の全てのジョブは SCore-D が管理
- ◆ SCore-D は互いに通信するプロセス群
- ◆ 通信の返事をすべきノードがダウンすると、返事を待っているプロセス(スレッド)が永遠に停止
- ◆ これを回避するのは非常に難しい

=>

- ◆ いきなりノードが停止するのではなく、事前に通知があった場合を対処しよう

SCore動的再構成 (3)

- ◆ マルチユーザ運用の運用上の問題
 - ◆ 一部のノードを止める場合
 - ハードおよびソフトの保守
 - ◆ マルチユーザの運用を止めなければならない
 - 大規模クラスタ(=多くのユーザ)での運用
 - 停止は、簡単ではない
- ◆ 運用を止めずに一部ノードの停止および復帰を可能にする

SCore動的再構成 (4)

- ◆ SCore-D 起動モードの追加

 - 現在：scout 経由で SCore-D を起動

 - 今後：“scored -daemon”としてホスト毎に起動する
モードを追加

- ◆ SCore-D のコンソールからノードの削除／追加を指定

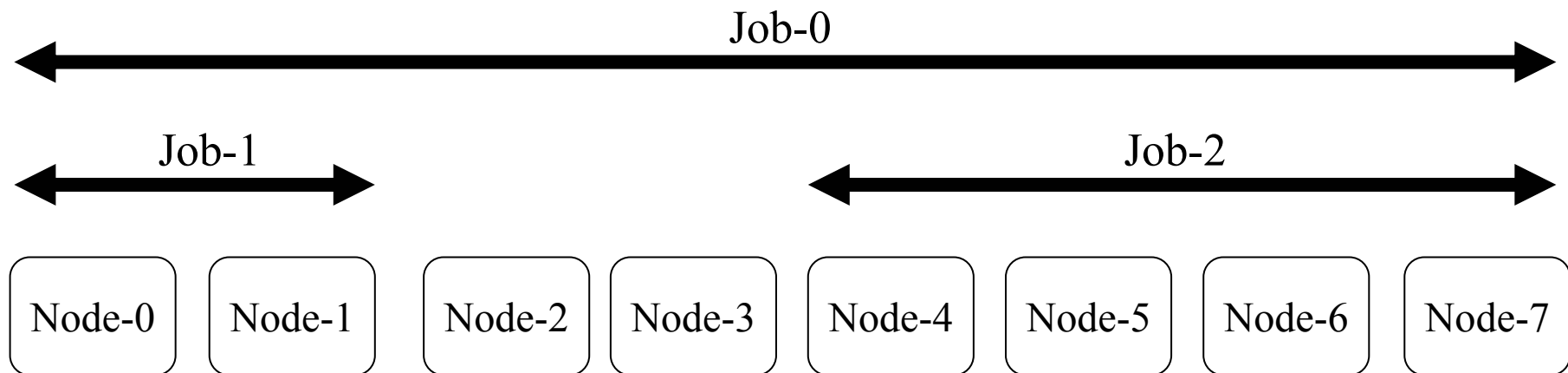
- ◆ 削除ノード上のジョブはチェックポイントが採られ、

 - ◆ 可能なら削除完了後にリスタート、

 - ◆ 可能でなければ、可能になるまでサスペンド

*SCore*動的再構成 (5)

- ◆ 動的再構成のイメージ (1)
運用中

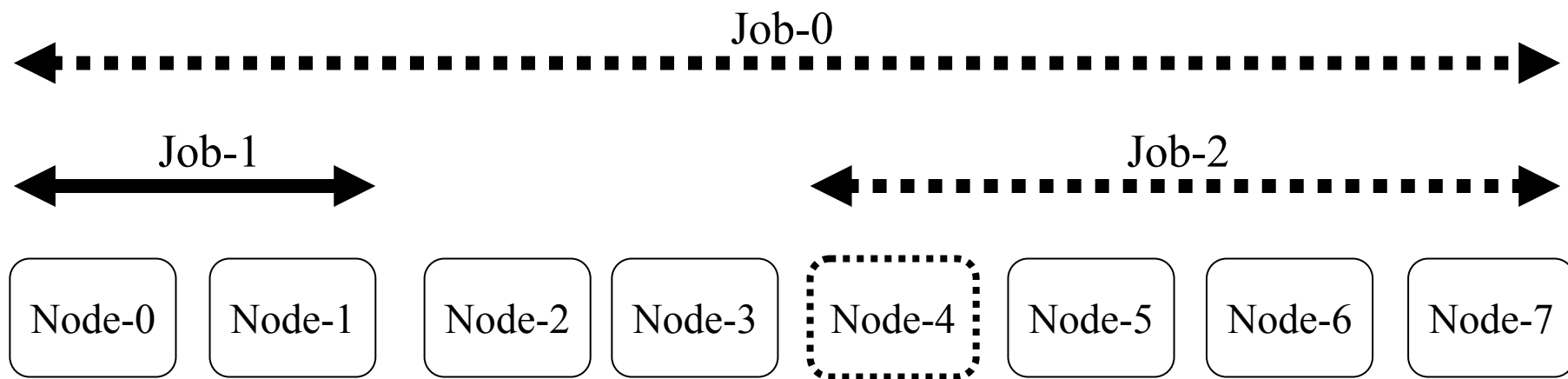


*SCore*動的再構成 (6)

◆ 動的再構成のイメージ (2)

ノード (Node-4) の削除開始

削除に伴い、Job-0 と Job-1 をチェックポイント

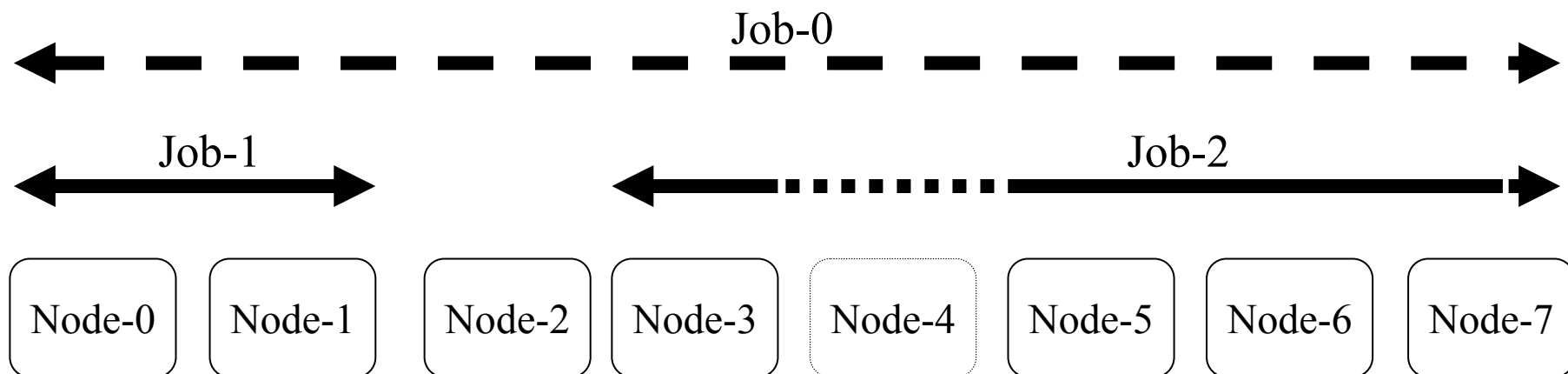


*SCore*動的再構成 (7)

◆ 動的再構成のイメージ (3)

ノード (Node-4) の削除完了

Job-2 をリスタート、Job-1 はサスペンド



*SCore*動的再構成 (8)

- ◆ 動的再構成のイメージ (4)
 - ノード (Node-4) の追加 (復帰)
 - Job-0 のリスタート

