DataSpider Servista 4.3
導入ガイド

2021/04/20

株式会社セゾン情報システムズ
HULFTビジネスユニット

HULFT

目次

はじめに

■ DataSpider Servista 導入フロー

本資料では、以下のフローに従ってDataSpider Servista（以下、DSS）の導入を解説します。

1 冗長化構成の選定 → 2 サイジングの実施 → 3 導入構成の選定 → 4 リポジトリDB有無の選定 → 5 導入作業

DSSでは、導入を解説するドキュメント「インストールガイド」を標準で用意しています。本資料と合わせてご参照ください。

DataSpider Servista 4.3 インストールガイド
(DSS43_InstallGuide_ja.pdf)

１．冗長化構成の選定

冗長化構成の制約事項

本項では、DSSの冗長化構成の構築に必要な情報をまとめています。

■ 冗長化構成の制約事項

- DSSをHA (High Availability) クラスタ構成に組み込むことによってコールドスタンバイに対応します。
- フェイル発生時にアクティブ側で処理中だったデータ連携トランザクションはフェールオーバーされません。
- **フェイルオーバーを想定した連携データリカバリ設計は個別に必要**です。
  また、DSSが使用するディレクトリはどちらのノードでも同じ構成にしなければなりません。

DSSでは、クラスタリング構成を解説するドキュメント「クラスタリングガイド」を用意しています。本資料と合わせてご参照ください。

DataSpider Servista 4.3 クラスタリングガイド
(DSS43_ClusterConfigurationGuide_ja.pdf)

冗長化構成の要素

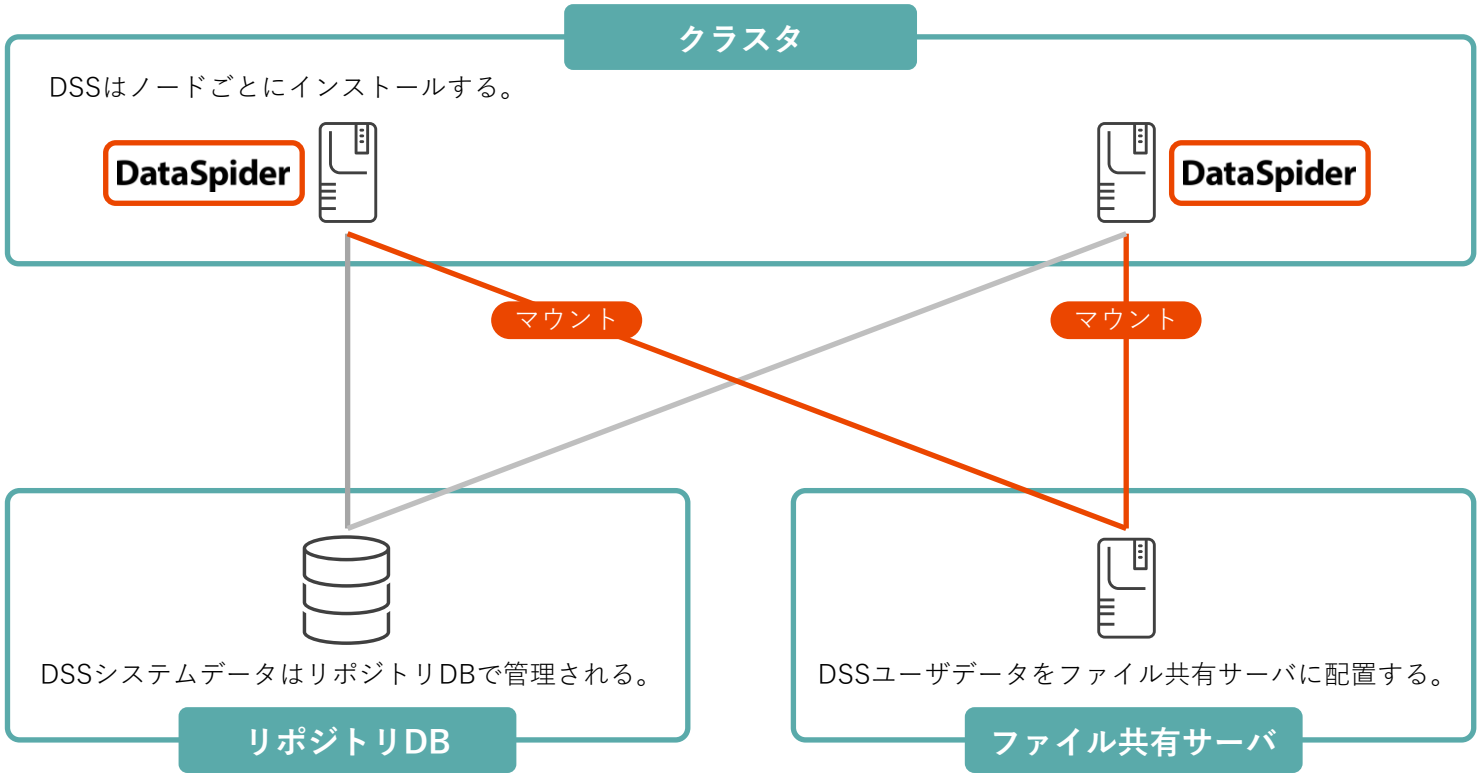
DSSの冗長化を実現する要素は以下のとおりです。

要素	No.	選択肢	説明	備考
DSSインストール先	1	ノード別	ノードごとにDSSをインストールします。	• ノード間のデータ同期が必要 • No.2より性能速度が速い
	2	ファイル共有サーバ	ファイル共有サーバにDSSをインストールし、ノードから参照します。	• ノード間のデータ同期が必要ない • No.1より性能速度が落ちる
リポジトリDB	3	有り	リポジトリDBにデータを保存します。	• リポジトリDBの機能を使える • ノード間のデータ同期が必要ない • No.4より性能速度が落ちる
	4	無し	ローカルにデータを保存します。	• リポジトリDBの機能を使えない • ノード間のデータ同期が必要 • No.3より性能速度が速い
ファイル共有サーバ	5	使用する	ファイル共有サーバにデータを保存します。	• ノード間のデータの同期が必要ない • No.6より性能速度が落ちる
	6	使用しない	ノードごとにデータを保存します。	• ノード間のデータの同期が必要 • No.5より性能速度が速い

リポジトリDBのメリット・デメリットについては、本資料「リポジトリDBとは」の項をご参照ください。

本資料における推奨構成

本資料では、DSS冗長化における推奨構成を、
「**ノード別にインストール＋リポジトリ有り＋ファイル共有サーバを併用**」とします。



ポイント

クラスタノード上でデータは置かず、ファイル共有サーバにデータを置くことで、可用性と保守性、性能速度を高めることが可能です。

2．サイジングの実施

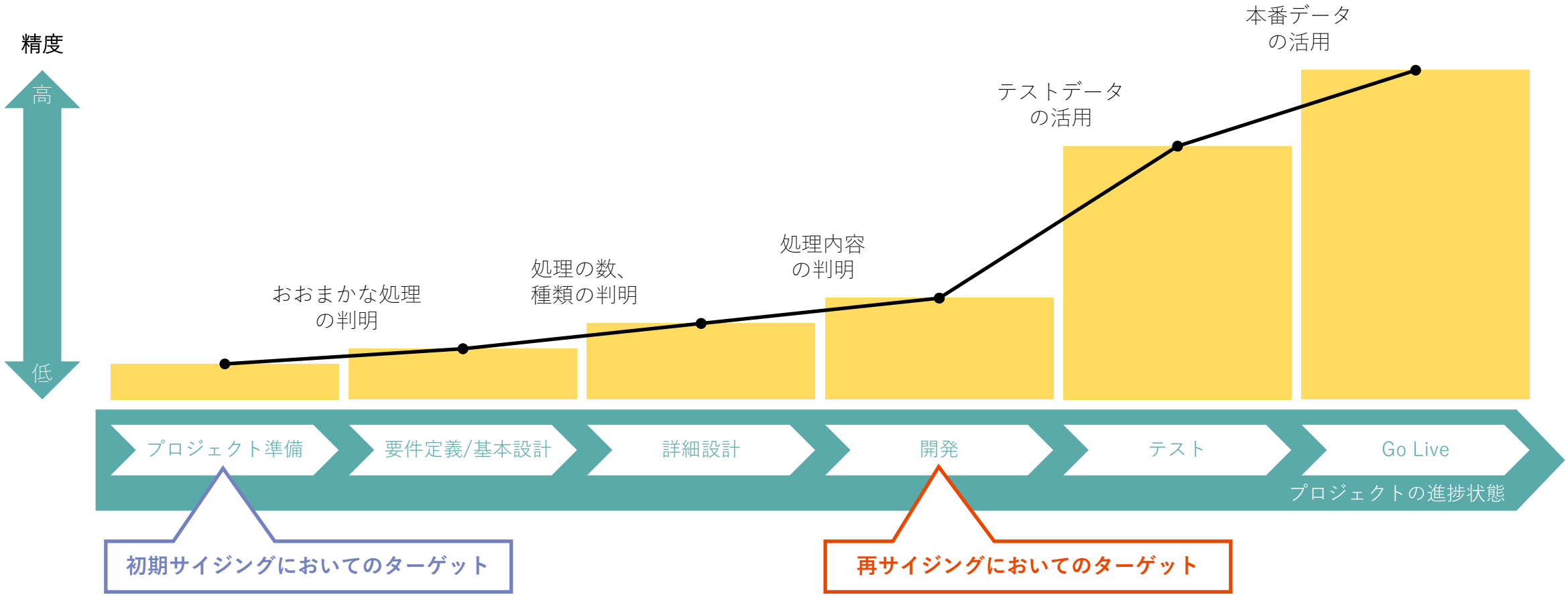
DataSpider Servista のサイジング

本項では、システム全体のキャパシティ・プランニングにおいて、DSSを運用するハードウェアを選定する際の指針と方法を参考情報としてまとめています。

システム構築する際に適切なハードウェアを選定するため、ベンチマーク結果や実例などから必要なリソースを検討することをサイジングの目的としています。

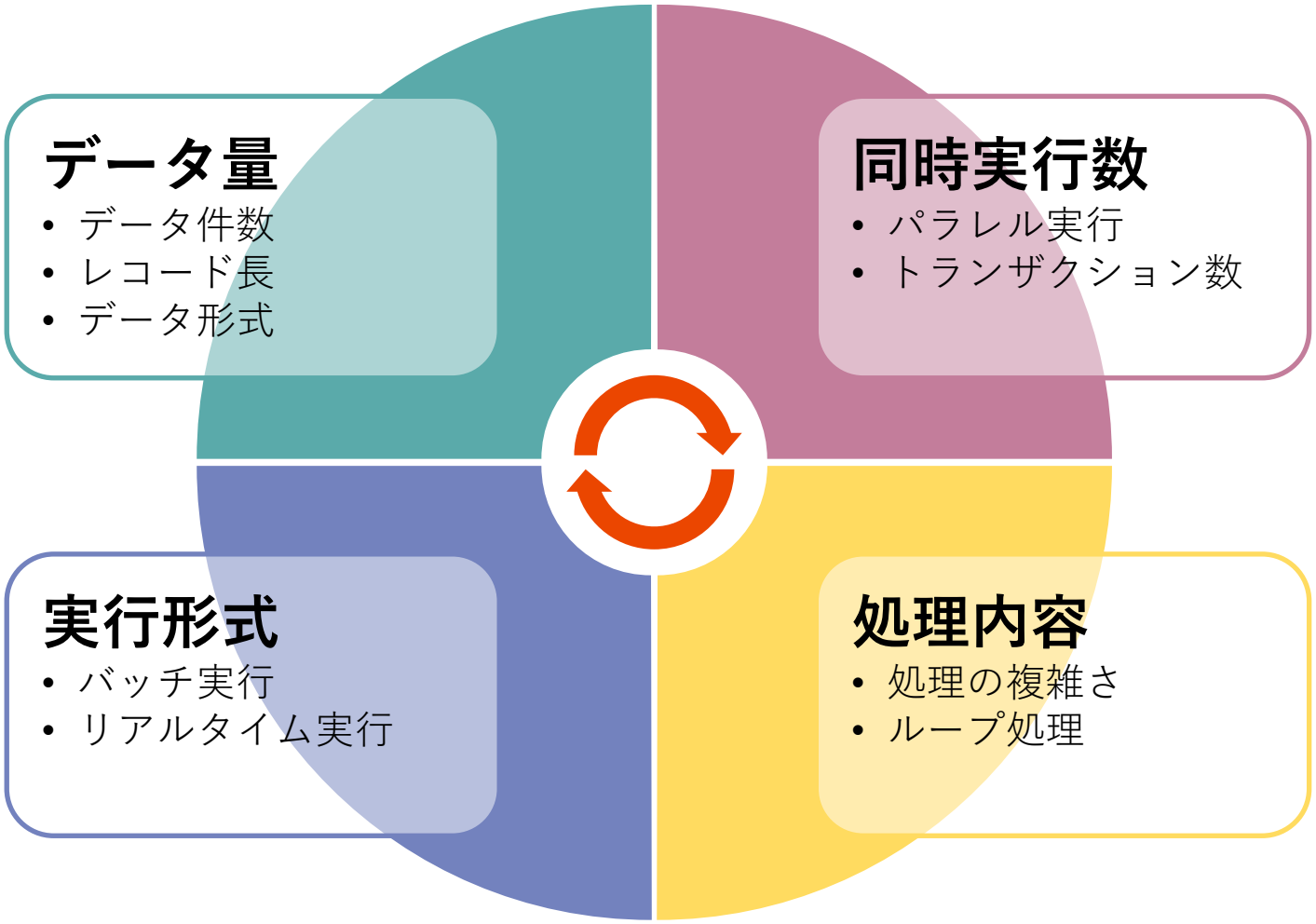
サイジングのタイミング

サイジングはプロジェクトの進捗状態によって、精度に大きな差がでます。
初期段階でのサイジングは精度に問題があり、ある程度の精度のぶれを考慮する必要があります。



スケーラビリティ（拡張性）の要素

DSSのサイジングにおける、スケーラビリティの要素は以下のとおりです。
これらの要素を確定させていくことで、サイジングの精度を高めていくことができます。



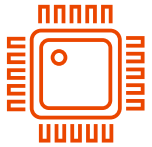
DSSサイジングの前提

本資料では、DSSのサイジングを行うにあたり、以下の内容を前提としています。

- **リソース消費は線形比例する**
  データ量が2倍になれば消費メモリも2倍になる。
- **パフォーマンスはハードウェア（HW）の種類には依存しない**
  特定のHWに依存した問題は発生しない。（製造メーカなど）
  サーバ、デスクトップなど種類にも依存しない。
- **予測不可能な現象は無視をする**
  予測不可能な現象は発生しない。
- **100%の精度は実現できない**
  あくまで目安であることを意識する。

サイジングの要素

DSSにおけるサイジングの要素は以下のとおりです。



CPU

性能、64bit or 32bit 、個数、 コア数

メモリ

メモリ容量、64bit or 32bit（※CPUに依存）、利用可能領域

ディスク

ディスク容量、性能、バックアップ

CPUサイジングの考え方

DSSにおけるCPUサイジングのポイントは以下のとおりです。

- **クロック数だけでは性能は計れない**
  メーカー・種類・世代・クロック数・コア数・アーキテクチャによって性能は異なる。

- **必ずしも「CPUの性能 = 処理能力」ではない**
  ただし、処理能力の指標にはなる。

ベンチマークをベースとした
相対的な性能比較による評価が有効

メモリサイジングの考え方

DSSにおけるメモリサイジングのポイントは以下のとおりです。

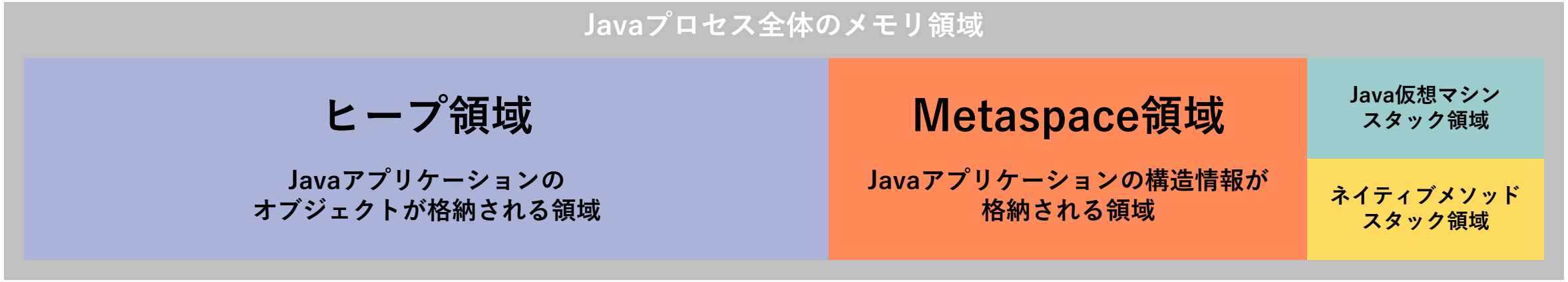
- **メモリが足りなくなると運用に支障がでる**
  OutOfMemoryエラーの発生、OSの動作が不安定になる。

- **メモリ消費量は処理によって変動する**
  データ量、データ形式、処理の複雑さによって消費されるメモリ量は大きく変動する。

運用に支障がでない値（エラーが発生しない）を
選定することを目標にする

メモリ領域について

メモ



- **ヒープ領域**
  DataSpiderServer上での処理に使用される。
  以下の場合に大幅に消費される傾向がある。
  ・スクリプト上での大容量データの読み取り時
  ・サイズの大きなプロジェクトファイルを読み込み時

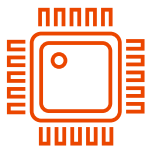
- **Metaspace領域**
  DataSpiderServerが使用するJavaのクラスやメソッド情報に使用される。
  以下の情報が多い際に大幅に消費される傾向がある。
  ・すべてのサービス上に配置されたMapperの保存情報
  ・デザイナから実行されたスクリプトが属するプロジェクト内のすべての Mapper の保存情報

DSSに割り当てできるメモリサイズ

DSSは、インストールするOSに合わせて以下の2種類を用意しています。

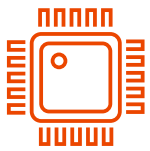
- 32bit (x86) 版 … 32bit/64bit OSにインストール可能
- 64bit (x64) 版 … 64bit OSにインストール可能

32bit版DSSと64bit版DSSでは**割当できるメモリの最大値が異なる**ため、注意が必要です。



32bit版DSS

OSの制限により1DSSに割当できるメモリは2GBまで
→ JVMが使用する部分もあるため、実質1.4～1.6GB程度



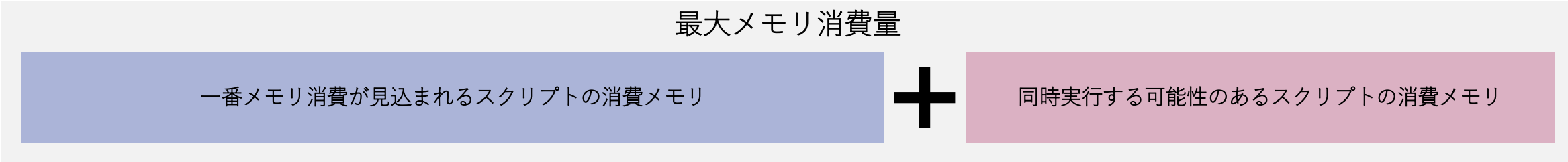
64bit版DSS

実質、搭載されている物理メモリまで割当可能
※OSの種類やバージョンによって制限を受ける場合があります。

メモリサイジングの方法

DSSでは、データの容量・形式、処理の複雑さによって消費するメモリ容量は大きく変動します。また、CPUとは異なり、メモリ容量が不足するとOutOfMemoryエラーが発生し、処理が停止する恐れがあります。メモリ不足が発生しない値をターゲットにサイジングします。
メモリサイジングの方法としては、最もメモリ消費が見込まれるスクリプトから最大メモリ消費量を考えるようにします。

1. **一番メモリ消費が見込まれるスクリプトの消費メモリを計測する**
2. **1.のスクリプトと同時実行する可能性のあるスクリプトの消費メモリを計測する**
3. **1.と2.の消費量の合計を算出する**



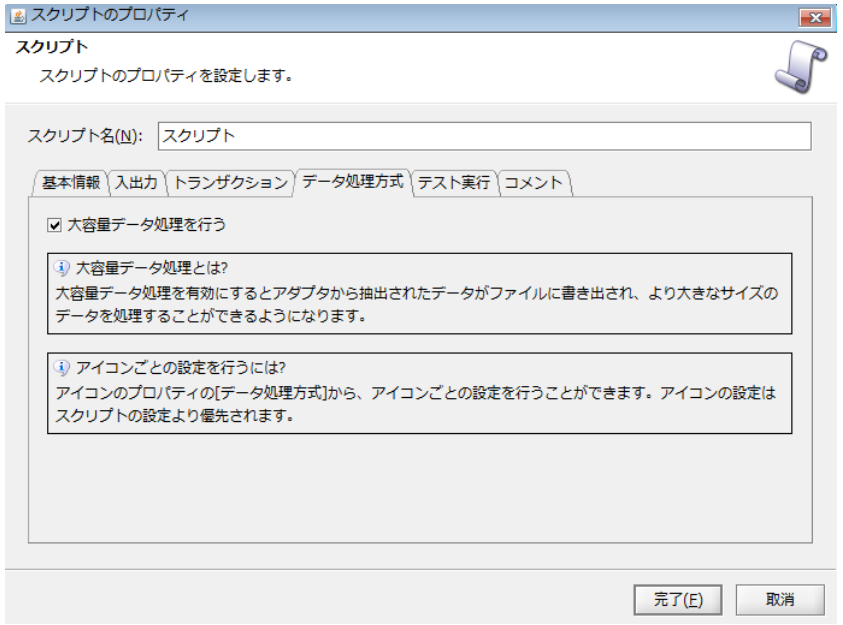
ポイント **DSSにおいて、オンメモリで高速な処理を実現するために、メモリは消費量の総和とすることでスループットを向上できます。**

大容量データ処理設定について

メモ

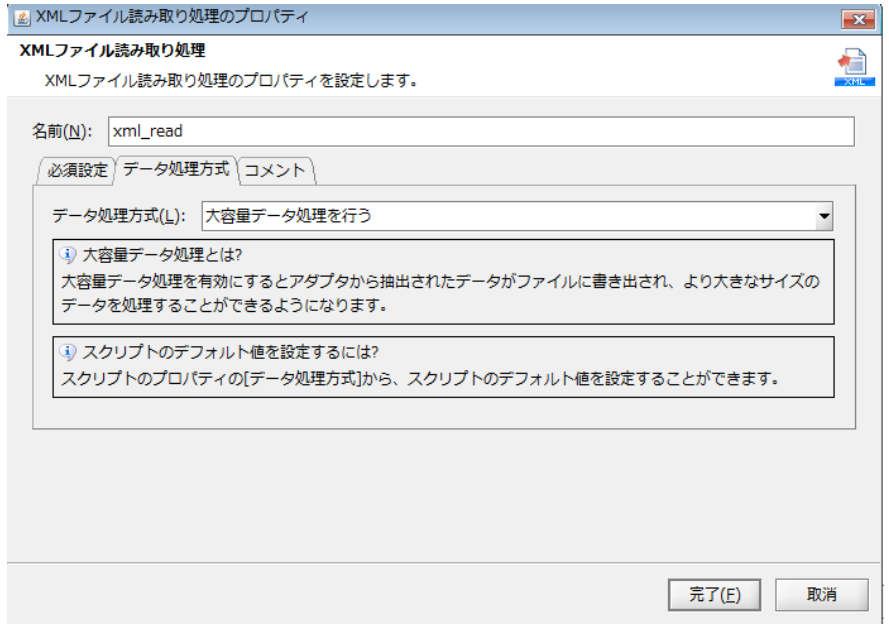
DSSに割り当て可能なメモリサイズに上限があり、ヒープサイズを十分に確保できない場合でも処理を停止させないためには、スクリプトで大容量データ処理設定を有効にすることで対処できます。
ただしディスクに書き出しながら処理を行うため、ファイルのI/Oが多発し、実行速度の低下が発生する場合があります。
大容量データ処理機構の設定は、スクリプトの全体設定として有効にするか、コンポーネントごとに個別に有効にするかを選択できます。

スクリプト全体設定



or

コンポーネント個別設定



ディスクサイジングの考え方

DSSにおけるディスクサイジングの考え方は以下のとおりです。

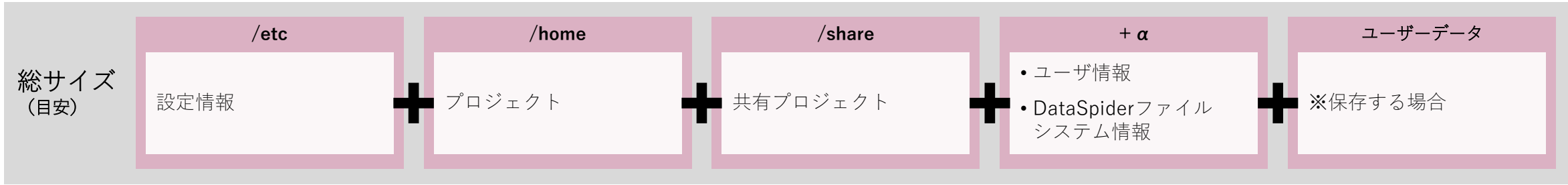
必要なディスク容量	格納データ	考慮すべきポイント
	**動作領域** ● 最小構成時の必要空き容量など	各プラットフォームにより異なる。 （「インストールガイド」参照）
	DataSpider ファイルシステム 主なディレクトリ ● 設定ファイル (/etc) ● プロジェクト (/home) ● 共有プロジェクト (/share) ● ログ (/logs) ● 一時利用フォルダ (/tmp) ・ ・ ・	サイズに影響が大きいのは… ● プロジェクト (/home,/share…) ・プロジェクトのサイズ -処理に依存(アイコンの数・種類) -バージョン数に依存 ・リポジトリDB有無 （次項「リポジトリDBの表領域サイズ」参照） ● ログ(/logs) ・設定、処理内容に依存 （ヘルプ「ログガイド参照」） ● 一時利用フォルダ(/tmp) ・大容量データ処理の使用有無
	＋α ● データ（保存する場合）	データを保存する場合、 必要に応じて空き容量を確保する。

リポジトリDBの表領域サイズ

DSSのリポジトリDBには、ユーザ情報、プロジェクトの他、DataSpiderファイルシステム情報（※１）、DSSの設定情報（※２）などが格納されます。

- **格納対象となるディレクトリ**
  /etc、/home、/shareディレクトリ以下

（※１）ディレクトリ構造やファイル構成、アクセス権、実データを含む
（※２）グローバルリソース、トリガーなどの各種設定ファイル



サイズに大きな影響を与えるのは、/home、/share以下にあるプロジェクトです。プロジェクトサイズは、基本的にスクリプトで使用しているアイコンの種類や数に依存します。また、各種Mapperで、どのようなロジックアイコンを使用するかによってもサイズは異なるため、要件による影響が大きく、標準的な目安を示すことは難しいといえます。

実環境で必要とされるサイズについては、要件に応じたスクリプトを実際に作成したあと、プロジェクトをダウンロードしたサイズを目安としていただくことを推奨します。

３．導入構成の選定

導入環境の選定

■ 単一環境と複合環境

DSSでは、単一環境（1つのDataSpiderServer）で開発～本番までを行う方式と、環境を複数（開発～検証～本番など）に分けて使用する複合環境の2種類の方式を選択できます。

■ クラウド環境とオンプレミス環境

DataSpiderServerをクラウド環境とオンプレミス環境、どちらに配置するかを選択します。

本資料では、**クラウド環境に配置する際の注意点**について説明します。

単一環境と複合環境のメリット・デメリット

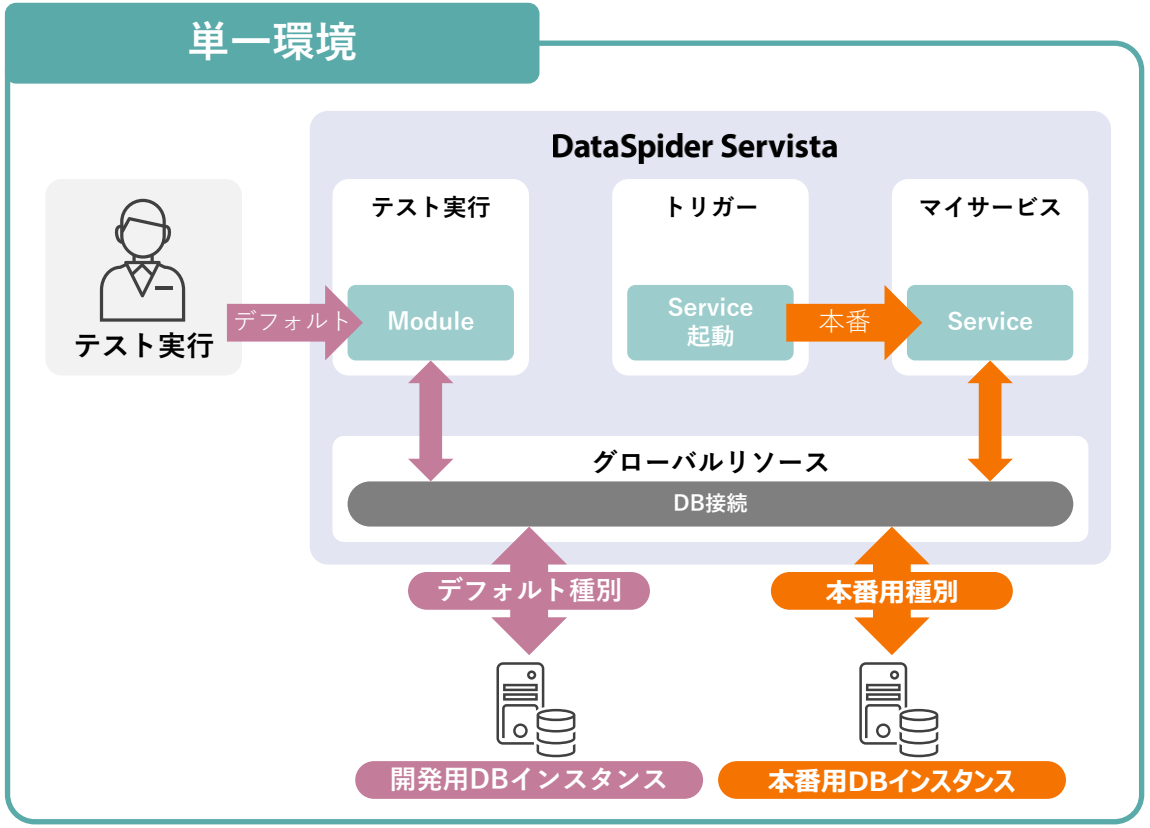
単一環境と複合環境のそれぞれのメリット、デメリットは以下のとおりです。

	メリット	デメリット
単一環境	● サーバ1台で済むため、サーバ自体の運用負荷を軽減できる。	● DSSの運用が複雑になる。 ● 本番稼働後、追加開発や検証・再現調査などが、本番の処理に影響を与える恐れがある。
複合環境	● DSSの運用がシンプルになる。 ● 本番と切り離して追加開発・検証・再現調査を実施することが可能。	● 環境ごとに複数のサーバ、ライセンスを用意する必要がある。

ポイント **DSSで作成したスクリプトに充分なテスト環境を確保し、DSSの運用負荷を下げるために、複合環境を推奨します。**

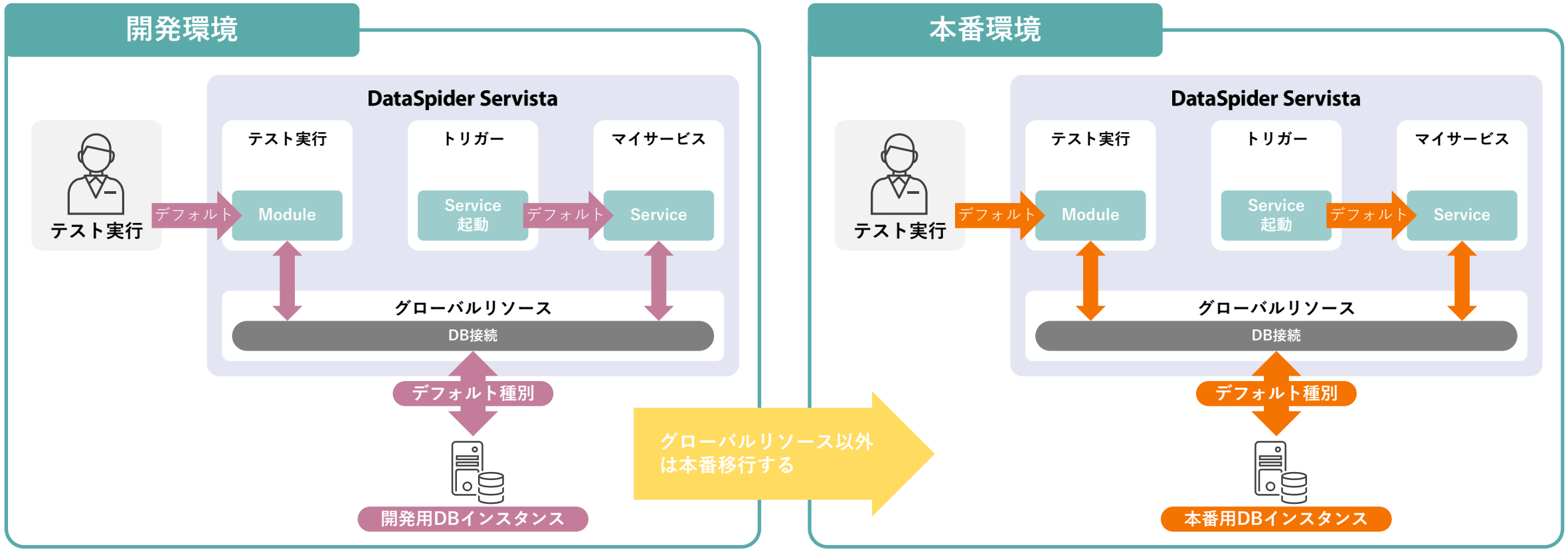
単一環境の運用についてのヒント

開発から本番までサーバ1台で行う場合には、「種別」機能を活用することで、運用を行いやすくなります。以下の図は、スクリプトの実行までを開発フェーズとし、サービスの実行（トリガー実行）を本番フェーズとして2種類の種別を使い分けるという例になります。3フェーズ（開発>検証>本番）の場合は、3種類の種別が必要となるため、トリガー設定およびサービスを検証用と本番用と2元管理するなどの工夫が必要になります。



複合環境の運用についてのヒント

DataSpiderServerを独立した複数の環境構成で運用する場合には、個々の環境に同一名称のグローバルリソース設定を静的に設置させ、グローバルリソースは環境個々に管理することをお奨めします。こうすることで、スクリプトに変更を加えず、各フェーズの移行が可能になります。（「種別」を使用するとフェーズ（開発>検証>本番）を経るたび設定を手動で変更する必要があるためです。）この場合、グローバルリソースは個々の環境で個別に管理し、移行対象から外すというルールになります。



クラウド環境配置時の注意点

DataSpiderServerをクラウドに配置する場合、以下の問題が発生する場合があります。

問題	対応策
X Window SystemのないLinux環境のため、GUI方式のインストーラを利用することが出来ない。	● CUI方式のインストーラを用意しています。 CUI方式によるインストールを実施してください。 ※CUIインストールに対応しているインストーラは、「Linux x64 版」のみです。
オンプレミス環境にインストールしたStudio for Desktopから接続した場合、Studioの一部操作が遅くなる。（マイプロジェクトの起動やデバッグ実行など）	● クラウド環境がWindowsの場合、クラウド環境にStudio for Desktopをインストールして、リモートデスクトップで接続してください。 ● Studio for Webを使用してください。

また、DataSpider Servistaを仮想環境で使用する場合、以下のすべての条件を満たす必要があります。

- DataSpider Servista の稼働OS が、「サポートプラットフォーム」に含まれるOS である
- DataSpider Servista の稼働OS が、仮想環境がサポートしているOS である
- 仮想環境が、ホスト型、またはハイパーバイザー型である

上記条件をすべて満たした場合、サポート対象となります。インストールガイドで確認のうえ、サポート対象である場合のみインストールが可能です。

ただし、特定の仮想環境に依存する問題が発生した場合、DataSpider Servista の対応ではなく、仮想境の変更を含めた対応が必要となる可能性があります。

４．リポジトリDB有無の選定

リポジトリDBとは

リポジトリDBは、RDB（リレーショナルデータベース）内に設定したリポジトリ領域にて、サービスやユーザ情報、各種設定データを管理する機構です。リポジトリDBが未設定（リポジトリDB無し）の場合、ユーザ管理機構やファイルのアクセスコントロールの機能を使用することはできません。

以下にリポジトリDBを使用するメリット、デメリットを記載します。

	メリット	デメリット
リポジトリDB 有り	● ユーザ、グループが作成可能 ● チーム開発機能が使用可能 ● アクセスコントロール機能が使用可能 ● アプリケーションとデータの分離が可能	● リポジトリ用DBの準備 ● リポジトリ用DBのメンテナンス要 （バックアップ、バージョンアップなど） ● ローカルアクセスに比べて遅い （リポジトリDBが遠隔地にあるとより顕著）
リポジトリDB 無し	● シンプルな運用が可能 ● ローカル（OSファイルシステム）アクセスのため早い	● ユーザ、グループが作成不可 （rootユーザ/rootグループのみ） ● アクセスコントロール機能が使用不可

ポイント

DSSを用いてチーム（複数人）で開発をする場合は、リポジトリDBの使用を推奨します。
1～2人などの少人数でDSSを利用する場合は、リポジトリDBの使用は推奨しません。

リポジトリDB有りで使用可能な機能一覧

リポジトリDB有りで使用可能な機能と、リポジトリDB無しの場合の代替方法は、以下のとおりです。

機能	リポジトリDB無しの場合の代替方法
ユーザ、グループの追加	開発ルールなどの運用で回避
アクセスコントロールの設定	開発ルールなどの運用で回避
チーム開発	開発者ごとにプロジェクトを作成して振り分け、管理者が適宜統合
拡張ローカルファイルシステム、 データベースファイルシステムの使用	代替方法なし
ファイル、ディレクトリのロック	開発ルールなどの運用で回避
プロジェクトの共有	開発者ごとにプロジェクトを作成

注意事項

■リポジトリDBに関する注意事項

- リポジトリDB有りから無しへの移行はできません。（無しから有りは可能です。）
- リポジトリDBは、1つのDataSpiderServerにつき、1つ用意してください。
  複数のDataSpiderServerで同一のリポジトリDBに接続することはできません。
- リポジトリDBとして使用するデータベースのインスタンスは、リポジトリDB専用として使用してください。

■リポジトリDBの設定方法

- リポジトリDBの設定は、以下の2とおりの方法があります。

DSSのインストール時にインストーラで設定する　**or**　DSSのインストール後、**DataSpider Studio**のコントロールパネルで設定する

ポイント

DSSのインストール時にリポジトリDBを利用するか定められない場合は、
インストール後、リポジトリDBを設定することが可能です。

5．導入作業

DSSインストール前の準備

- **クラスタソフトの準備**
  DSSを監視するクラスタソフトを準備します。
  ※冗長化構成をしない場合には省略可
- **リポジトリDBの準備**
  リポジトリDBとして使用するデータベースに専用のインスタンスを作成します。
  ※リポジトリDBを使用しない場合は省略可
  ※DataSpiderServerのインストール後に設定することも可能
- **DSSライセンスファイルの準備**
  インストール時に指定するDSSのライセンスファイルを準備します。
- **DSSインストール先環境の準備**
  DSSをクラウド環境に構築する場合は、クラウドインスタンスにDSSがサポートするOSを準備します。
  DSSをオンプレミス環境に構築する場合は、実筐体上もしくは仮想環境上にDSSがサポートするOSを準備します。
- **セキュリティ対策の検討**
  DataSpiderServer や SAP EICS、JDBC Proxy Server など、外部アクセスを受け付けるアプリケーションについて、セキュリティ対策を検討します。

アダプタの事前設定

DataSpiderServerのインストール時に、アダプタのインストールも同時に行われます。
ただし、一部アダプタについては別途ライブラリのインストールや事前設定が必要になるケースがあります。
使用するアダプタが該当していないかを精査する必要があります。

ライブラリのインストールなど事前設定が必要なアダプタ

No.	アダプタ名
1	DB2 アダプタ
2	MySQL アダプタ
3	Oracle アダプタ
4	SQL Server アダプタ
5	JDBC アダプタ
6	NeoCore アダプタ
7	IBM Domino アダプタ
8	IBM Notes アダプタ
9	SAP アダプタ
10	Tableau アダプタ

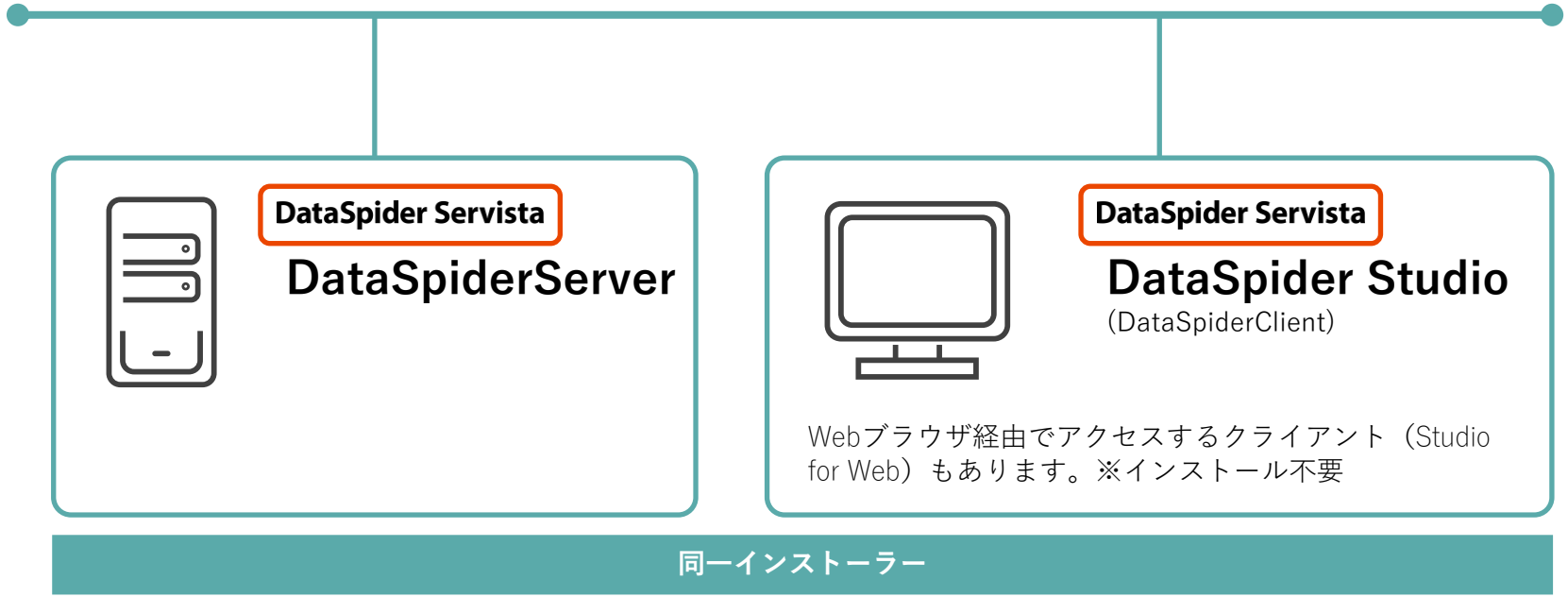
No.	アダプタ名
11	List Creator アダプタ
12	JMS アダプタ
13	Amazon Web Services アダプタ
14	Microsoft Azure アダプタ

ポイント **使用するアダプタが上記一覧に含まれている場合は、インストールガイドを参照のうえ、事前設定内容を確認してください。**

DataSpiderServerとDataSpider Studio

DSSはクライアント・サーバモデルで動作しています。

DataSpiderServerの主な役割は、処理の実行です。またDSSでは、クライアントのことを主にDataSpider Studio（以下、Studio）と呼びます。Studioの主な役割は、統合開発環境およびDataSpiderServerに実行指示を行うインターフェースです。



ポイント

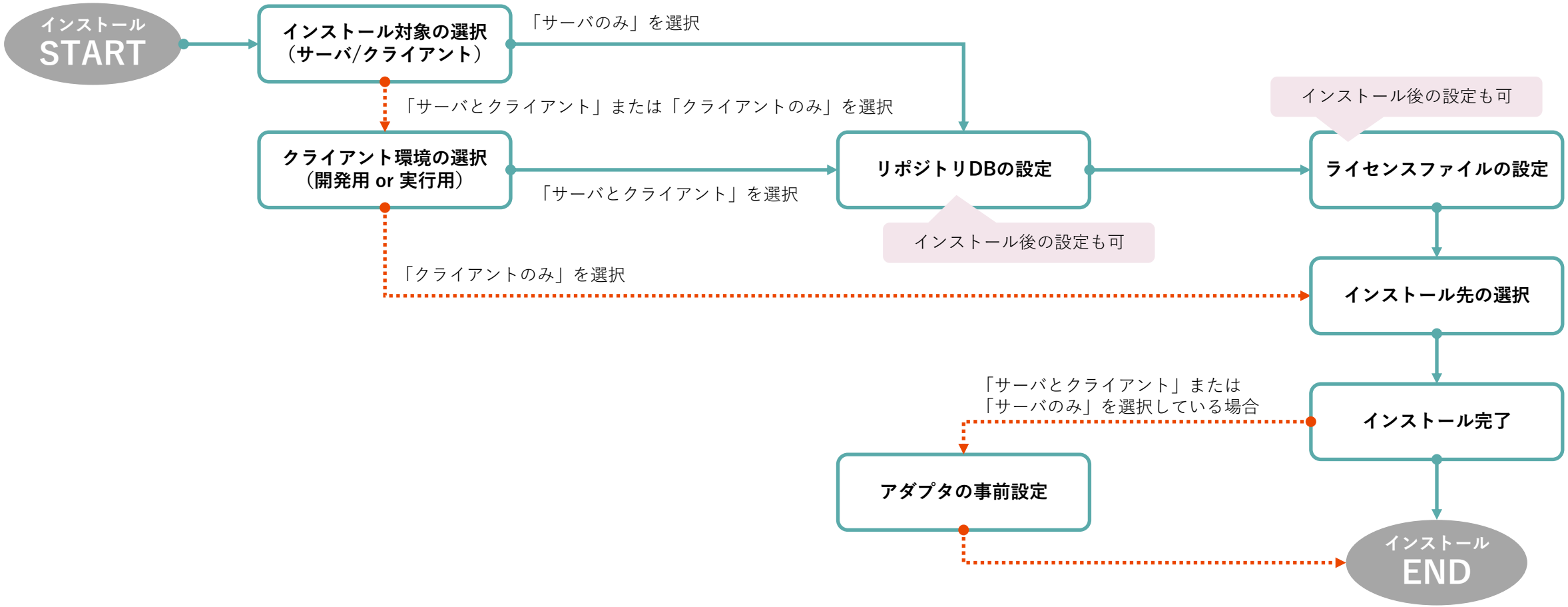
インストーラは同一です。環境が同じ場合は一度のインストールでOK。
環境が別の場合は、それぞれインストールが必要です。

セキュリティ対策

DataSpider Servista をより安全にご使用いただくためには、十分なセキュリティ対策が不可欠となります。
DataSpiderServer、SAP EICS、JDBC Proxy Server など、外部からのアクセスを受け付けるアプリケーションに対しては、十分なセキュリティ対策の実施を強く推奨いたします。
ここでは代表的なセキュリティ脅威と、その対策の一例をご紹介します。

ウィルス対策	ウイルス対策アプリケーションを利用して、ウイルス感染などのセキュリティ脅威からサーバを保護します。 • ウイルス定義ファイルを常に最新版に更新する • 定期的にウイルススキャンを実行し、ウイルスを検知・駆除する
ネットワーク セキュリティ	外部からの不正なアクセスによる、情報漏えいや改ざんといった攻撃からサーバを保護します。 • サーバにアクセス可能なIPアドレスを制限する • IDS/IPSを設置し、不正な通信を検知・遮断する • Webアプリケーションファイアウォールを設置して、脆弱性を悪用した攻撃から防御する
ログの 収集・監視	予兆検知によりセキュリティ事故を未然に防ぎ、発生した場合の事後調査を円滑にします。 • ログ監視ツールを導入し、パケットログやイベントログ（syslog）、Tomcatのログファイルなどを監視する • DSS のアクセスログ機能を有効にし、監査証跡を残す
運用面での セキュリティ	適切な運用により攻撃者からの不正行為を抑止し、サーバを保護します。 • OS や DSS の管理者権限を不必要に与えない • 定期的にパスワードを変更する

DSSインストール・フロー



ポイント インストール作業はインストールガイドをご参照のうえ、行ってください。

HULFT

Move knowledge. Move markets.