自然言語処理における知識獲得のためのクラスタ環境の利用

京都大学 黒橋研究室
村脇 有吾

2008年6月4日
InTrigger Community Workshop

概要

- 自然言語処理における利用事例の報告
  - 大量のデータの処理
    - メモリ上で完結しない
  - 粒度が非常に粗く、プロセスレベルで独立 (EP)
- 発表の流れ
  - 知識獲得タスクの概要
  - InTrigger における実現方法
    - gxp3 を使った並列分散処理

研究目的

- 言語解析に必要な知識の自動獲得
  - 自然言語は曖昧
  - 曖昧性解消には文法以外の知識も必要
- 知識獲得の具体例
  1. 用言格フレーム
  2. 語彙知識

用言格フレームの獲得

?

クロールで 泳いでいる女の子を見た
望遠鏡で　泳いでいる女の子を見た

格フレーム

{人,子,…}が
{クロール,平泳ぎ,…}で
{海,大海,…}を泳ぐ

{人,者,…}が
{双眼鏡,望遠鏡,…}で
{姿,人,…}を見る

[Kawahara and Kurohashi, HLT2001, COLING2002, LREC2006]

語彙獲得: 背景 (1/3)

- 解析器は新聞向けにチューニング
- ウェブテキストに対して脆弱
  - 規範的でない表記法
    - 『とりあぇえず、デカいよぅ！！すごいよぅ（はぁと）』
  - 誤字・誤変換
    - 以外に安い。
  - 辞書にない語 (未知語)
    - ググる

語彙獲得: 背景 (2/3)

?

アイツの人生をググってみた。

ググる:
動詞 子音動詞ラ行

アイツの
人生を
ググってみた。

ググ:
未定義語 → 名詞扱い

アイツの
人生を
ググって
みた。

語彙獲得: 背景 (3/3)

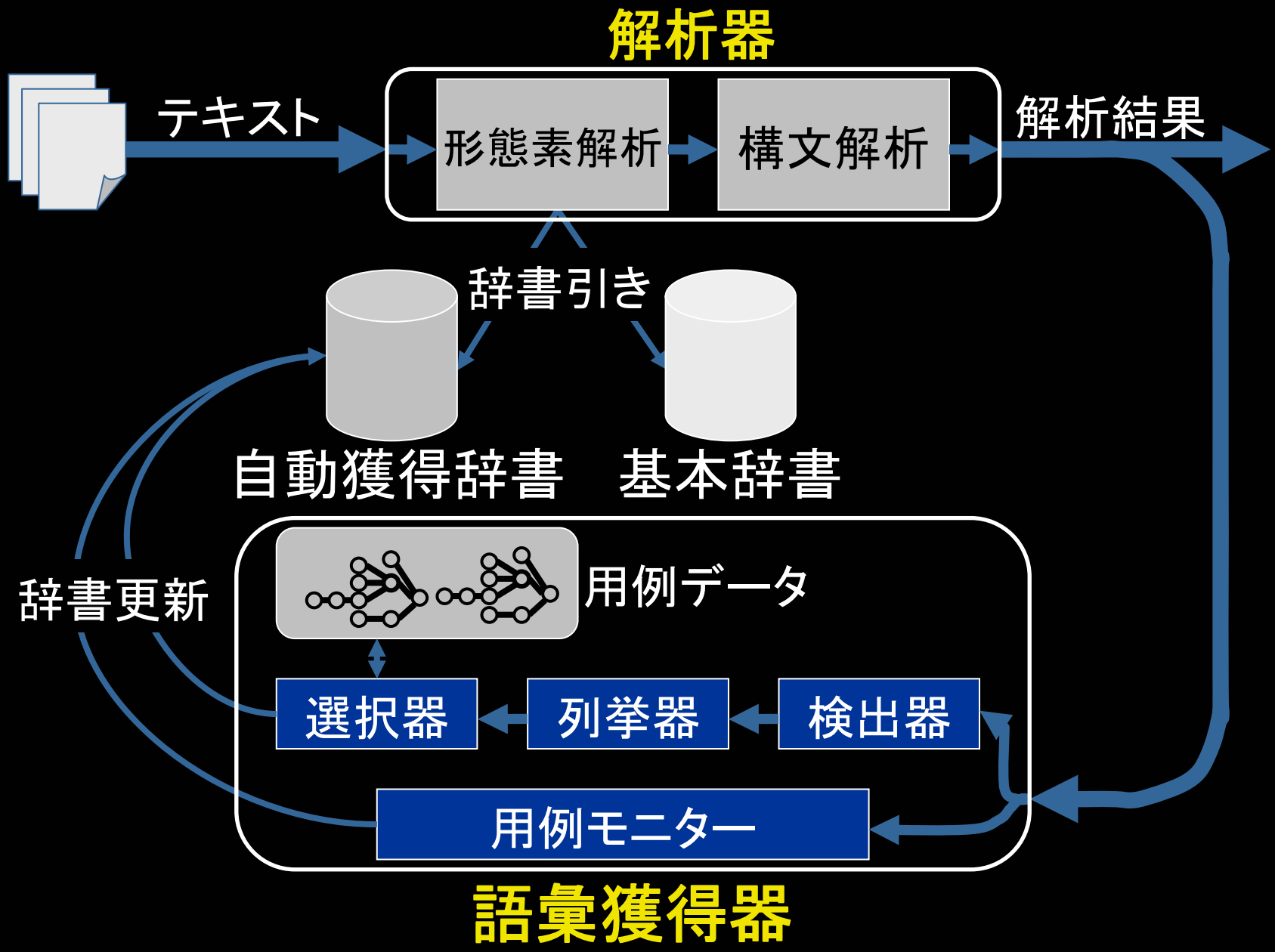
- 基本語彙 (約3万語)
  - 人手で整備済み
  - 表記ゆれ等の情報を付与
- オープンクラスの語彙
  - 計算機による自動処理
    1. ヒューリスティクスによる解析時の未知語処理
    2. テキストからの語彙獲得

語彙獲得: タスク

- 獲得すべき情報
  1. 形態素境界

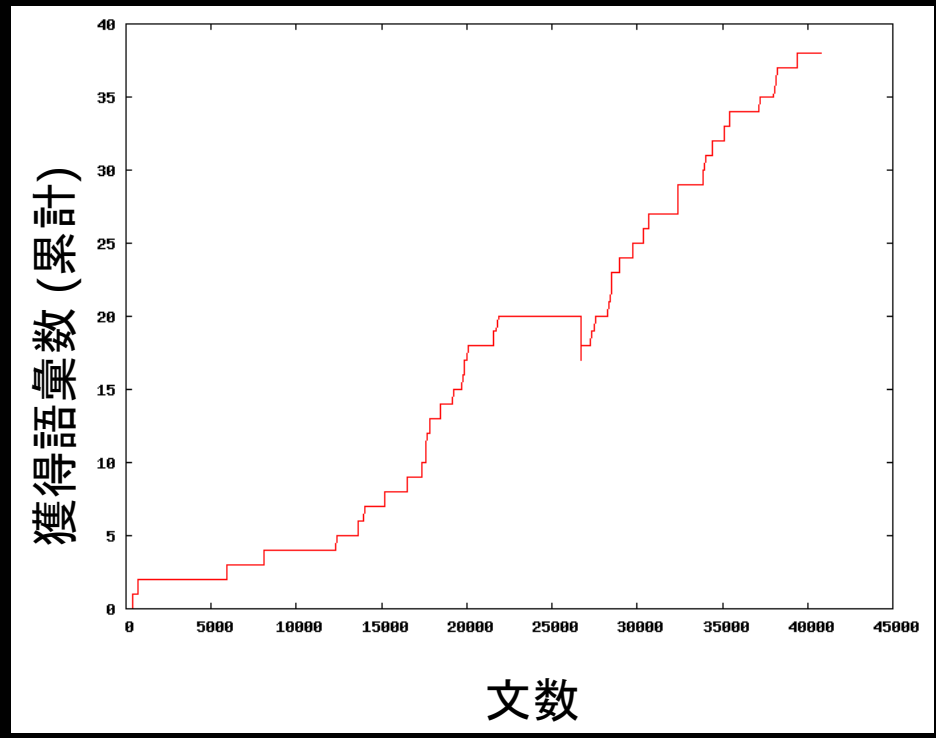
     アイツの人生をググってみた。
  2. 品詞

     ググ-る: 動詞-子音動詞ラ行
- 高精度の獲得
  - 解析に悪影響を及ぼさないように
- 比較的少数の用例からの逐次獲得


解析器
テキスト
形態素解析
構文解析
解析結果
辞書引き
自動獲得辞書
基本辞書
辞書更新
用例データ
選択器
列挙器
検出器
用例モニター
語彙獲得器

実行の経過

- 500ページ (40808文) を解析
- 38形態素を獲得
  - 獲得に要する用例数:6 (中央値)
- 再解析により8974文 (22.0%) が変化



アルゴリズム: 未知語の検出

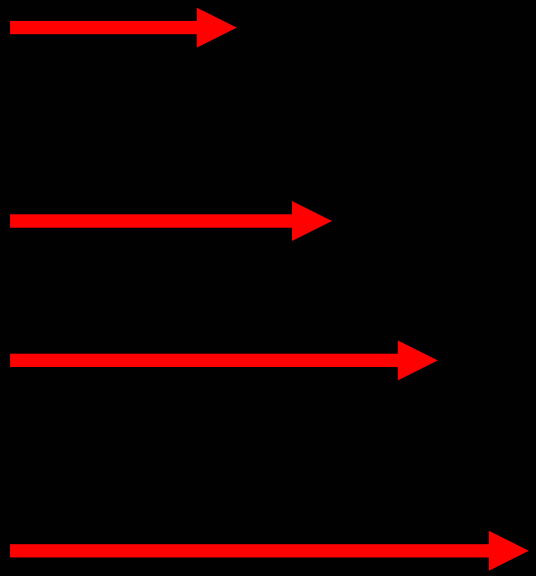
- 文の解析結果から未知語を含みそうな箇所を検出
  - この時点では語の境界は不明

アイツの人生をググってみた。

未定義語

アルゴリズム: 候補の列挙

ググってみた



品詞	活用形
子音動詞ラ行	タ系連用テ形
子音動詞ワ行	タ系連用テ形
子音動詞タ行	タ系連用テ形
母音動詞	タ系連用テ形
子音動詞マ行	基本連用形
母音動詞	タ形
子音動詞タ行	未然形

アルゴリズム: 候補の選択

- 蓄積した複数の用例を比較
- 曖昧性が解消されたら獲得

ググってみた		ググるのは		・・・
ググ	子音動詞ラ行 子音動詞ワ行 子音動詞タ行	ググ	子音動詞ラ行 母音動詞	
ググっ	母音動詞	ググる	母音動詞 普通名詞 サ変名詞 イ形容詞 ナ形容詞	
ググって	子音動詞マ行			
ググってみ	母音動詞 子音動詞タ行			

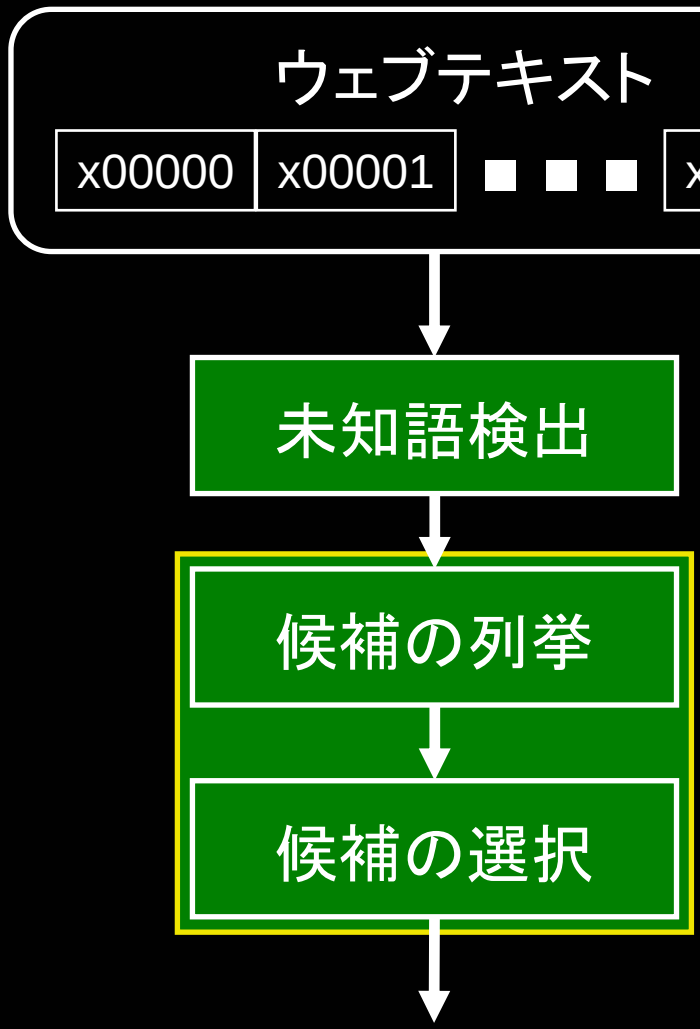
並列処理のタスク

1. 品詞の統計的振る舞いの抽出
2. 大規模テキストからの語彙獲得
   - 逐次獲得アルゴリズムの検証
   - 逐次ではなく、あらかじめ語彙を獲得しておく手法の検討

獲得例

- 難しい訓読み
  - 惹く: 動詞
  - 蒼い: 形容詞
  - 甘酸っぱい: 形容詞
- 難しい漢語
  - 憑依: 名詞
- 俗語、規範的でない表記
  - デカい: 形容詞
- 新語
  - ブログ: 名詞
  - 写メ: 名詞

語彙獲得の並列化

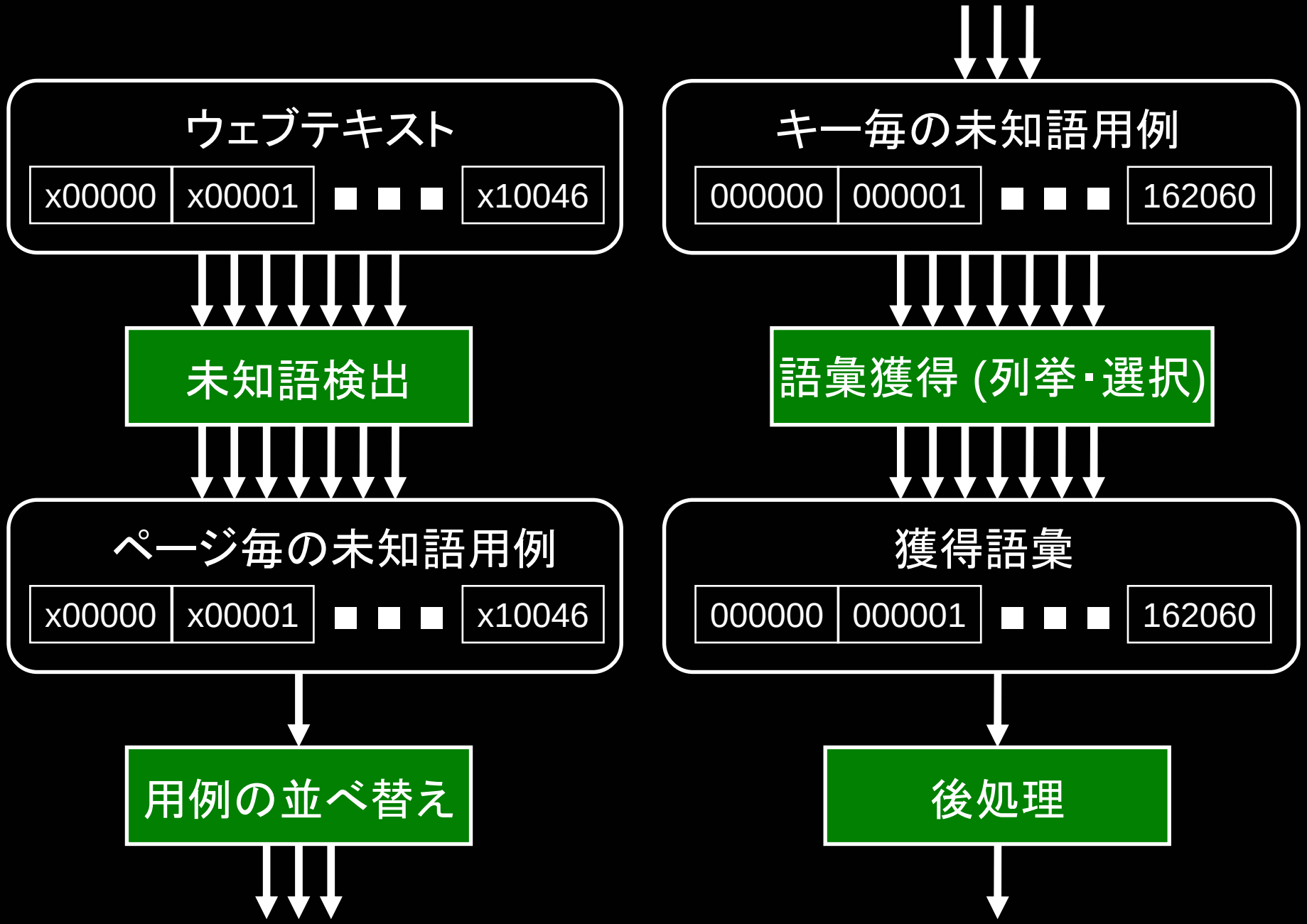


1億ページからの
逐次獲得は非現実的

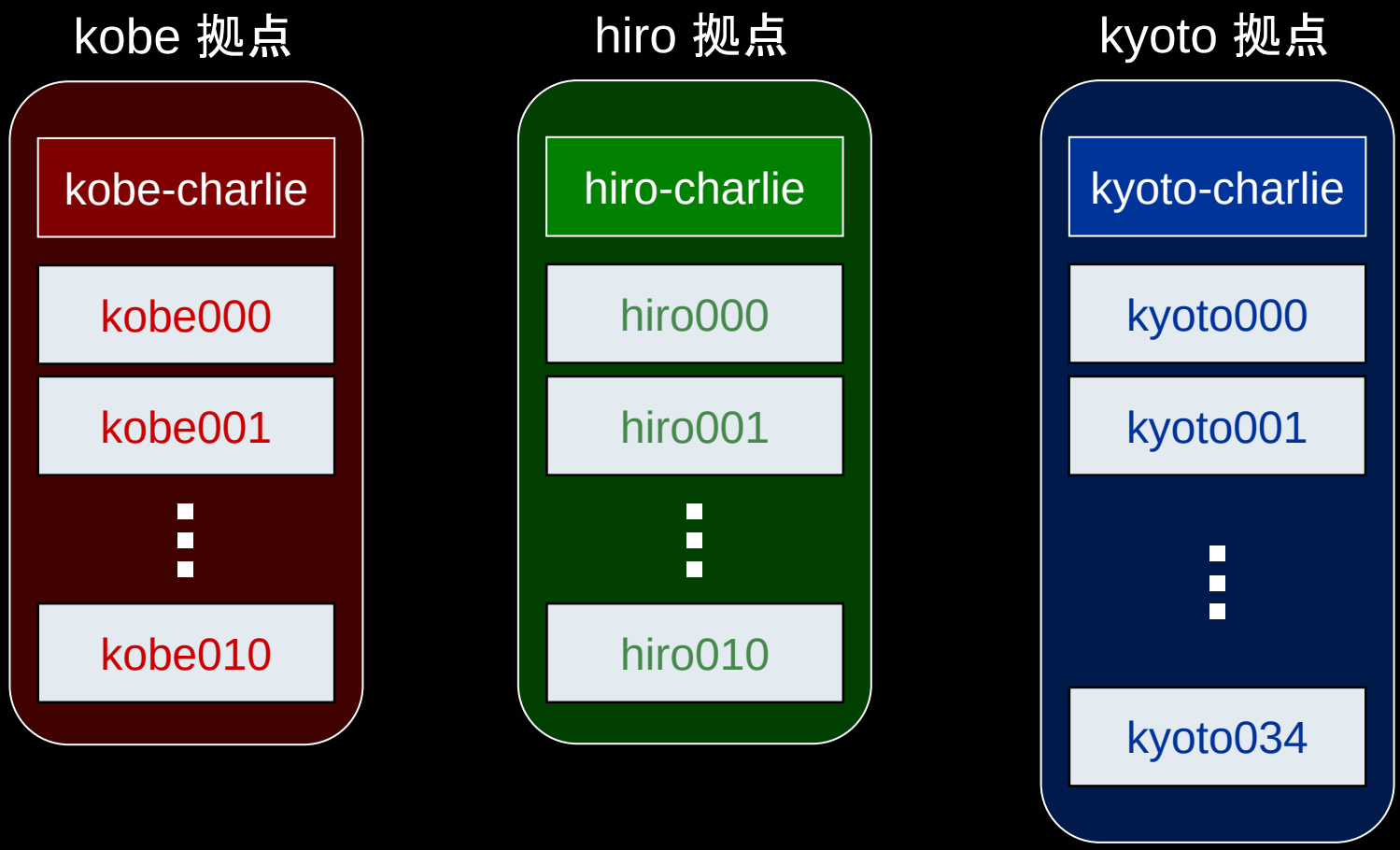
各ページからの
未知語検出を並列化

未知語の用例をキー毎に
大まかに仕分け

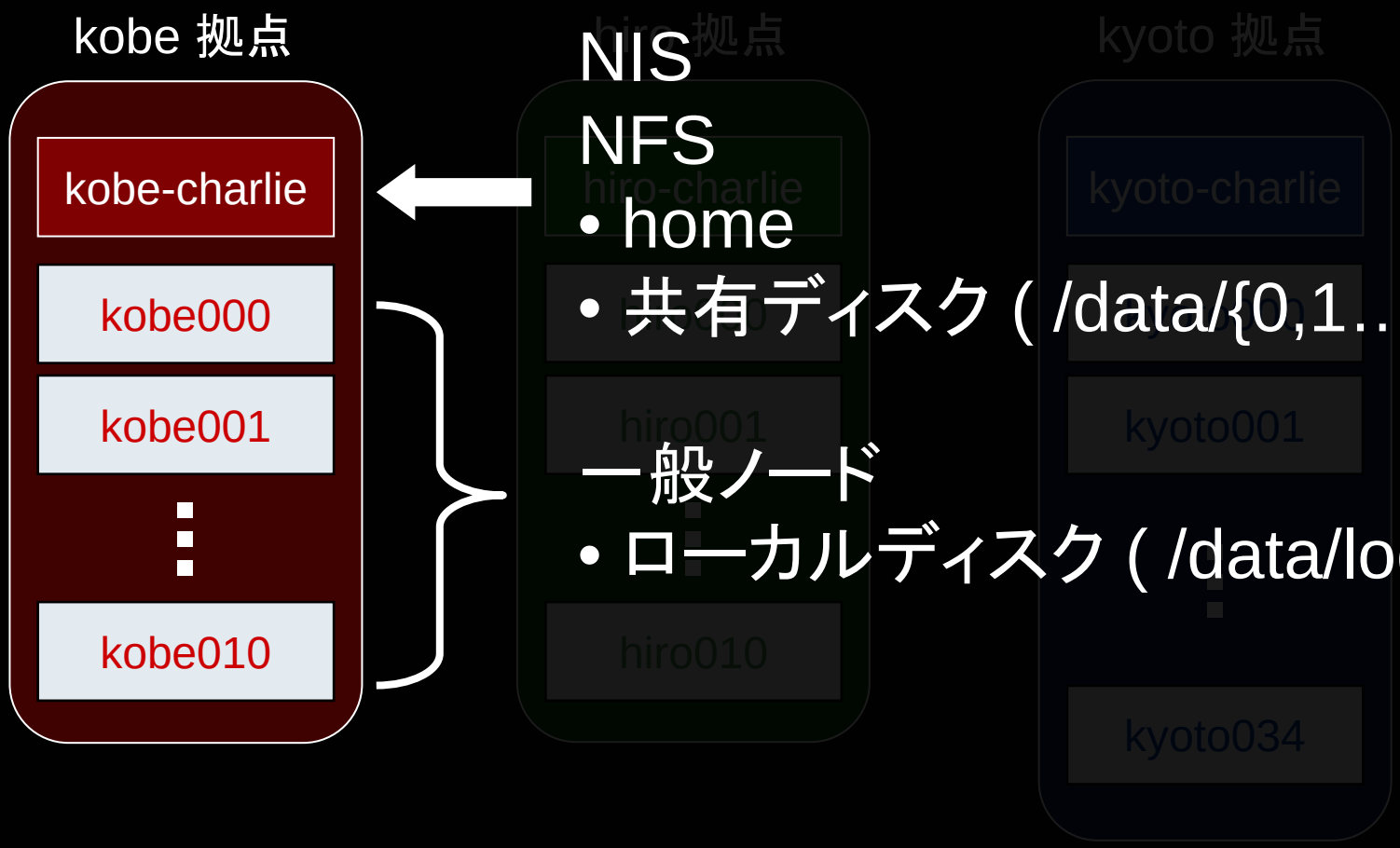
キー毎に
語彙獲得を並列化


ウェブテキスト
x00000
x00001
x10046
未知語検出
ページ毎の未知語用例
x00000
x00001
x10046
用例の並べ替え
キー毎の未知語用例
000000
000001
162060
語彙獲得 (列挙・選択)
獲得語彙
000000
000001
162060
後処理

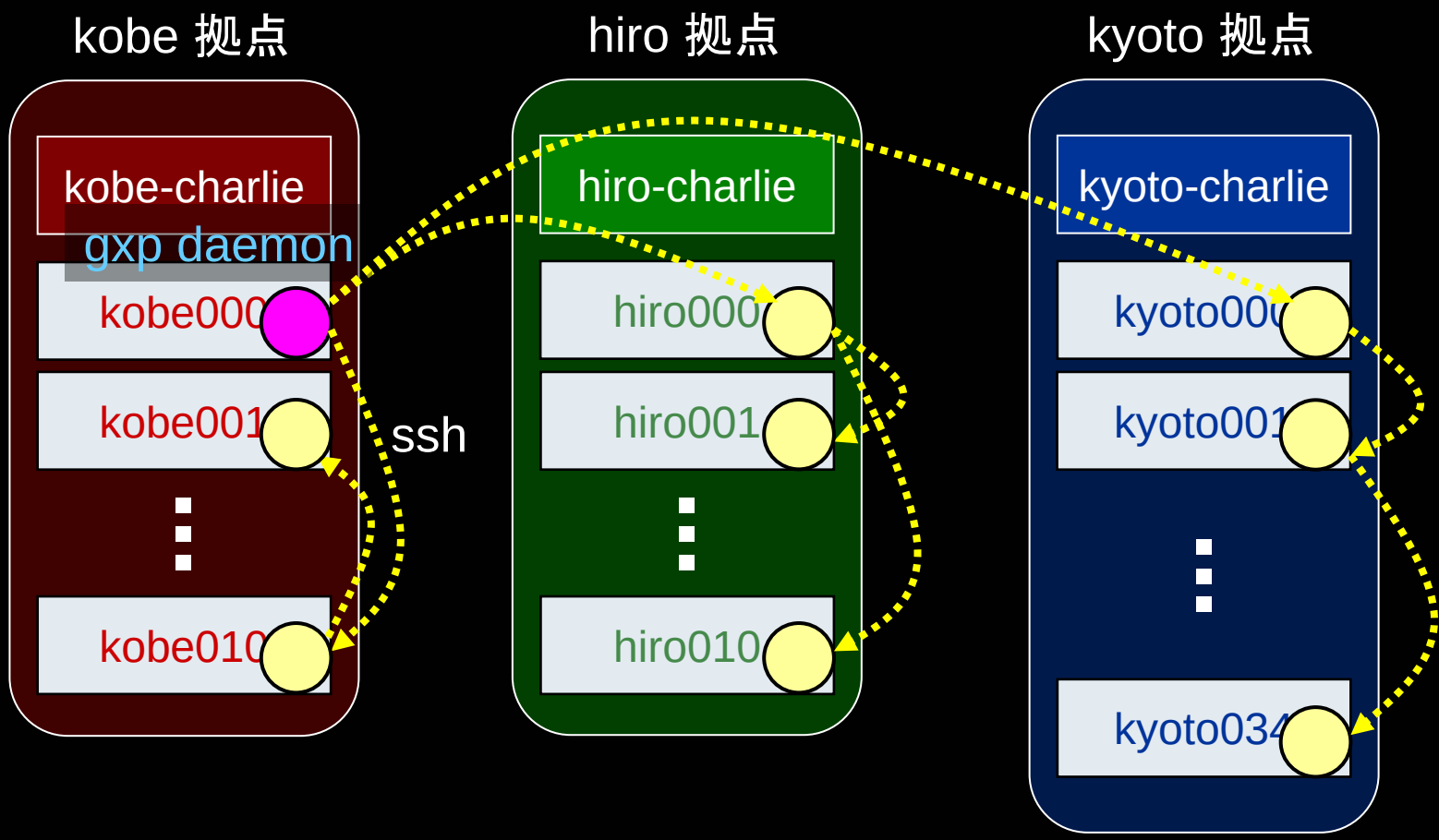
InTrigger


kobe 拠点
kobe-charlie
kobe000
kobe001
kobe010
hiro 拠点
hiro-charlie
hiro000
hiro001
hiro010
kyoto 拠点
kyoto-charlie
kyoto000
kyoto001
kyoto034

InTrigger: ノードの構成

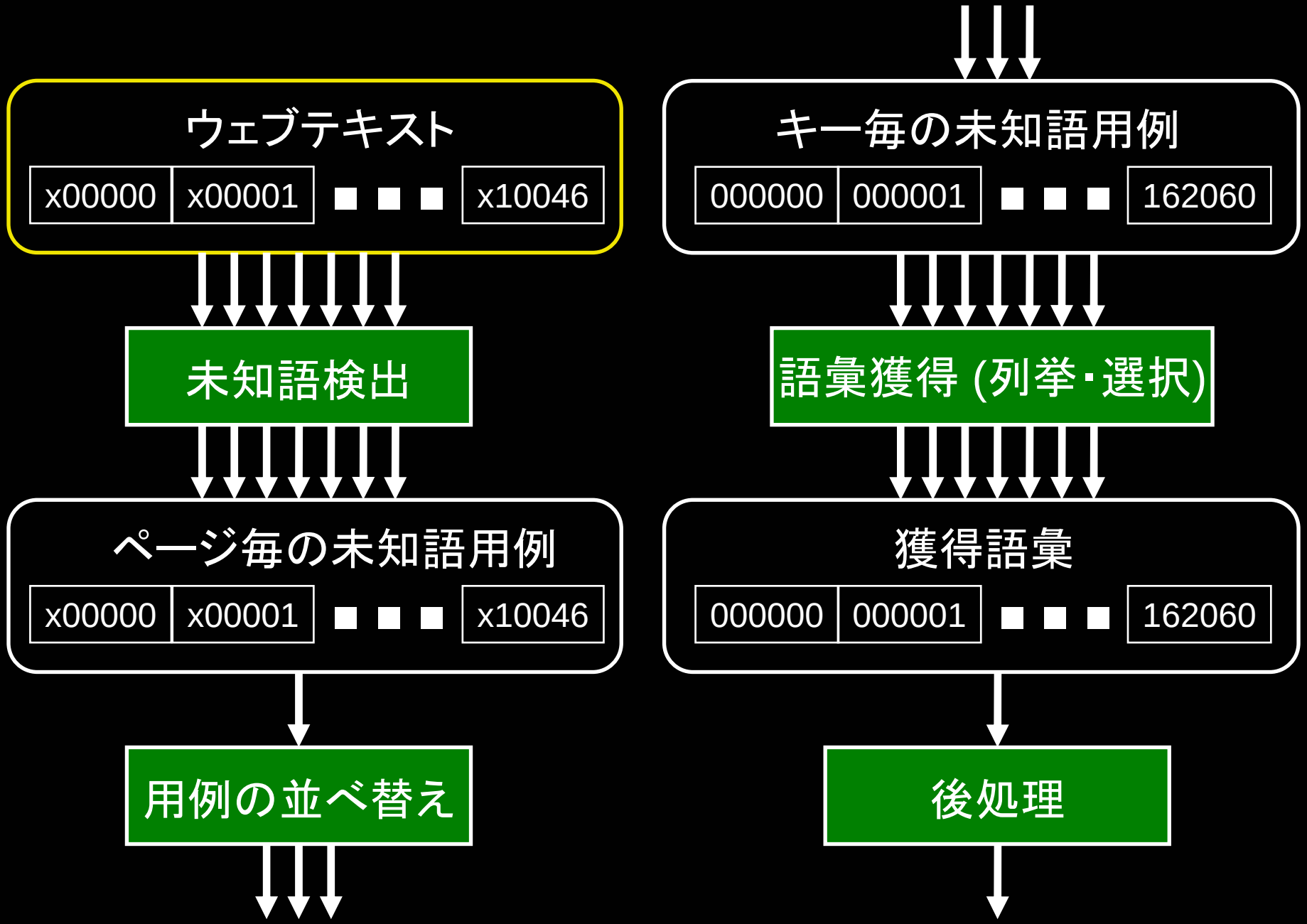


InTrigger: gxp3の利用

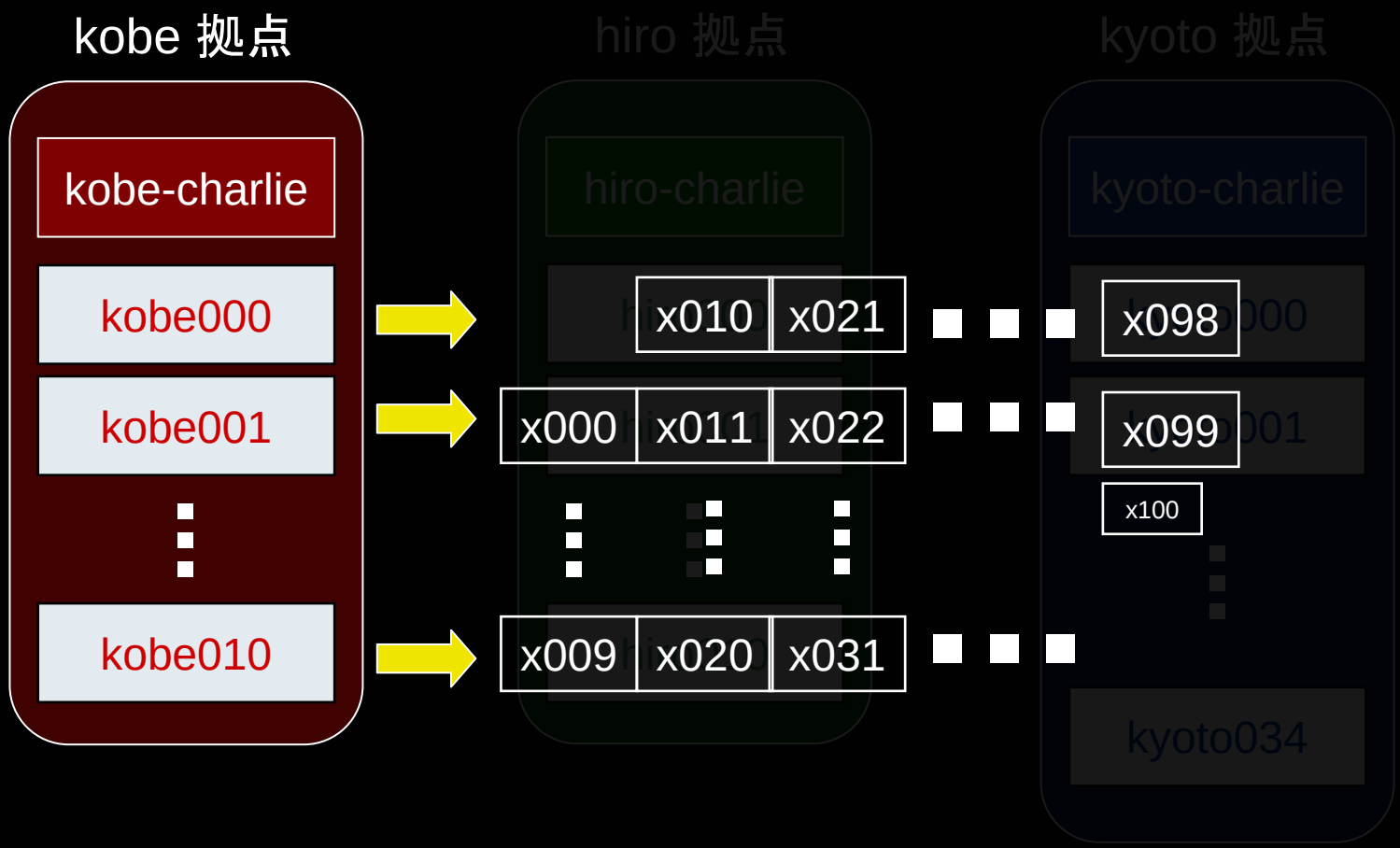


データの配置 (1/3)

- ウェブテキスト1億ページ
  - 巨大 (転送時間が馬鹿にならない)
    - 標準フォーマット 3.1TB (gzip 圧縮)
  - 複数の目的で利用
  - あらかじめ各ノードにばら撒いておく
    - 1万ページ (~300MB) × 100 × 100 に分割
    - 各ブロックを各拠点のローカルディスクに配置
      - 転送、展開に2、3日


ウェブテキスト
x00000
x00001
x10046
未知語検出
ページ毎の未知語用例
x00000
x00001
x10046
用例の並べ替え
キー毎の未知語用例
000000
000001
162060
語彙獲得 (列挙・選択)
獲得語彙
000000
000001
162060
後処理

データの配置 (2/3)


kobe 拠点
hiro 拠点
kyoto 拠点
kobe-charlie
kobe000
kobe001
kobe010
hiro-charlie
x010
x021
x000
x011
x022
x009
x020
x031
kyoto-charlie
x098
x099
x100
kyoto034

ウェブ

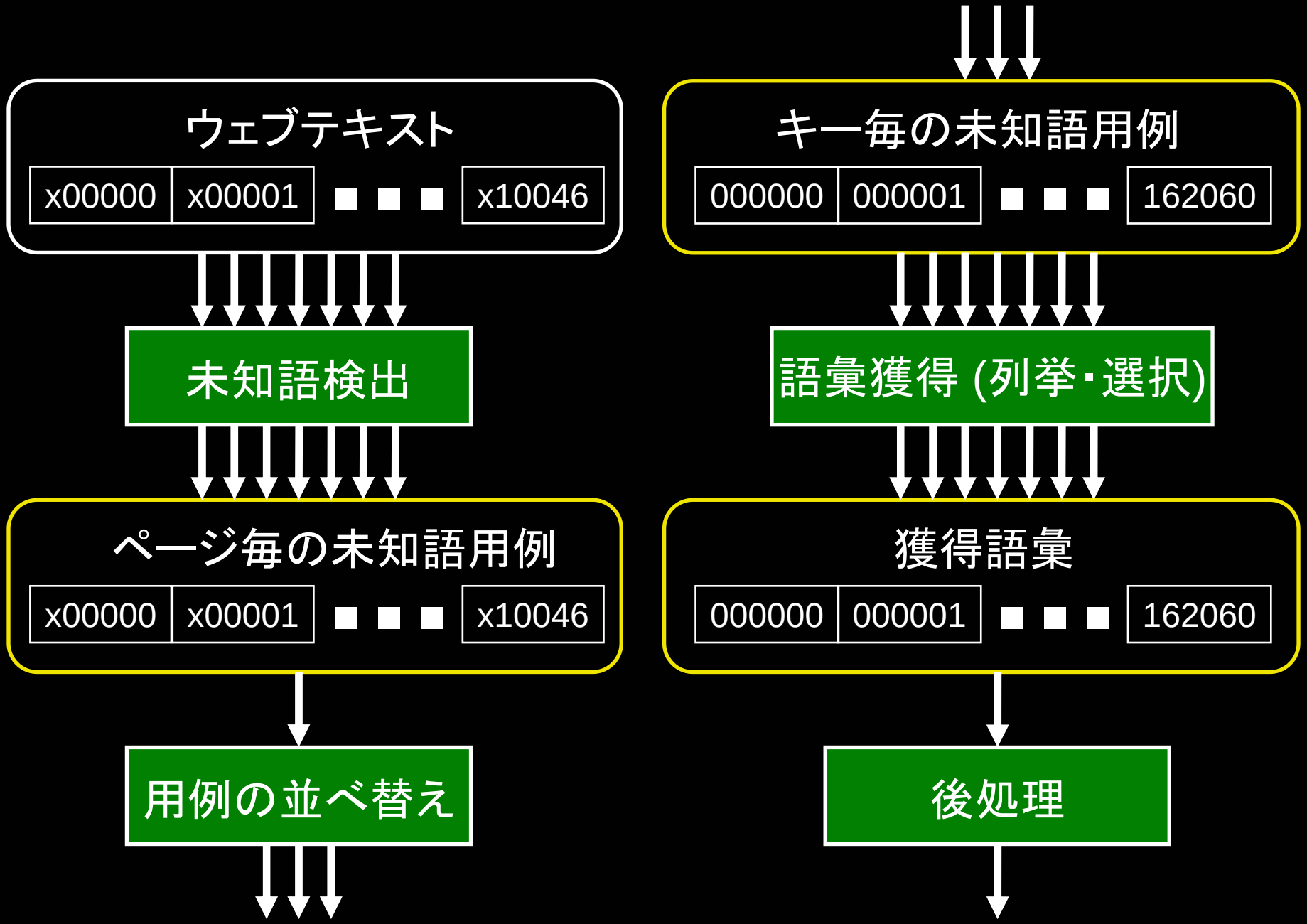
解析結果を付与

日本語文を抽出


```
<RawString>▼座右の銘</RawString>
- <Annotation Scheme="Knp">
  - <![CDATA[
    * -1D <BGH:座右の銘/ざゆうのめい><文頭><文末><体言><用言:判><体言止><レベル:C><区切:5-
    5><ID:（文末）><RID:112><提題受:30><主節>
    + 1D <文節内><係:文節内><文頭><体言><名詞項候補><先行詞候補>
    ▼ ▼ ▼ 特殊 1 記号 5 * 0 * 0 NIL <文頭><記英数カ><英記号><記号><自立><意味有><タグ単位始><文節
    + -1D <BGH:座右の銘/ざゆうのめい><文末><体言><用言:判><体言止><レベル:C><区切:5-5><ID:（文末）>
    座右の銘 ざゆうのめい 座右の銘 名詞 6 普通名詞 1 * 0 * 0 "代表表記:座右の銘/ざゆうのめい" <代表表記
    :ざゆ
    ]]>
  </Annotation>
</S>
<S Offset="2315" Length="348" is_Japanese="1" JapaneseScore="0.86111" is_Japanese_Sentence="1"
Id="5" Log="部分削除:20:(信無くば立たず)">
<RawString>小泉総理の好きな格言のひとつに「無信不立」があります。</RawString>
- <Annotation Scheme="Knp">
  - <![CDATA[
    * 1D <BGH:総理/そうり><文頭><連体修飾><サ変><人名><助詞><体言><係:ノ格><区切:0-
    4><RID:1067><候補:1:2:3>
    + 1D <文節内><係:文節内><文頭><人名><体言><名詞項候補><先行詞候補>
    小泉 こいずみ 小泉 名詞 6 人名 5 * 0 * 0 NIL <文頭><漢字><かな漢字><名詞相当語><自立><意味有><タ
    + 2D <BGH:総理/そうり><連体修飾><人名><助詞><体言><係:ノ格><区切:0-4><RID:1067><サ変><名詞項候
    総理 そうり 総理 名詞 6 サ変名詞 2 * 0 * 0 "人名末尾 代表表記:総理/そうり" <人名末尾><代表表記:総理
    の の の 助詞 9 格助詞 1 * 0 * 0 NIL <かな漢字><ひらがな><連体修飾><付属>
    * 2D <BGH:好きだ/すきだ><連体修飾><用言:形><タグ単位受無視><係:連格><レベル:B-><区切:0-5><ID:
    + 3D <BGH:好きだ/すきだ><連体修飾><用言:形><タグ単位受無視><係:連格><レベル:B-><区切:0-5><ID:
    好きな すきな 好きだ 形容詞 3 * 0 ナ形容詞 21 ダ列基本連体形 4 "代表表記:好きだ/すきだ" <代表表記:好
    * 3D <BGH:格言/かくげん><連体修飾><助詞><体言><係:ノ格><区切:0-4><RID:1067><候補:3>
    + 4D <BGH:格言/かくげん><連体修飾><助詞><体言><係:ノ格><区切:0-4><RID:1067><名詞項候補><先行詞
    格言 かくげん 格言 名詞 6 普通名詞 1 * 0 * 0 "代表表記:格言/かくげん" <代表表記:格言/かくげん><漢字
    の の の 助詞 9 接続助詞 3 * 0 * 0 NIL <かな漢字><ひらがな><連体修飾><付属>
    * 6D <カウンタ:つ><数量><ニ><助詞><体言><係:二格><区切:0-0><RID:1173><格要素><連用要素><候補:
    + 7D <カウンタ:つ><数量><ニ><助詞><体言><係:二格><区切:0-0><RID:1173><格要素><連用要素><名詞項
    ひと ひと ひと 名詞 6 数詞 7 * 0 * 0 NIL <かな漢字><数字><ひらがな><名詞相当語><自立><意味有><タ
    つ つ つ 接尾辞 14 名詞性名詞助数辞 3 * 0 * 0 NIL <カウンタ><かな漢字><ひらがな><付属><非独立タグ
    に に に 助詞 9 格助詞 1 * 0 * 0 NIL <品曖><ALT-に-に-に-9-3-0-0-NIL><品曖-格助詞><品曖-その
    * 5D <BGH:信/しん><括弧始><引用内文頭><体言><係:隣><裸名詞><区切:0-0><RID:1451><候補:5>
    + 6D <BGH:信/しん><括弧始><引用内文頭><体言><係:隣><裸名詞><区切:0-0><RID:1451><名詞項候補><
    「 「 「 特殊 1 括弧始 3 * 0 * 0 NIL <記英数カ><英記号><記号><括弧始><括弧><接頭><非独立タグ接頭
    無 む 無 接頭辞 13 ナ形容詞接頭辞 4 * 0 * 0 "代表表記:無/む" <代表表記:無/む><漢字><かな漢字><接
    信 しん 信 名詞 6 普通名詞 1 * 0 * 0 "漢字読み:音 代表表記:信/しん" <漢字読み:音><代表表記:信/しん
    * 6D <否定表現><ガ><括弧終><助詞><引用内文末><体言><係:ガ格><区切:0-0><RID:1108><格要素><連用要
```

データの配置 (3/3)

- 処理の中間データ
  - 比較的小さい
    - e.g. 検出された未知語の用例: 15GB (gzip圧縮)
  - 1拠点の共有ディスクに格納
  - 実行時に計算ノードに転送


ウェブテキスト
x00000
x00001
x10046
未知語検出
ページ毎の未知語用例
x00000
x00001
x10046
用例の並べ替え
キー毎の未知語用例
000000
000001
162060
語彙獲得 (列挙・選択)
獲得語彙
000000
000001
162060
後処理

並列処理の実行

- gxpc ep を利用
  - Usage: gxpc ep *tasks_file*
  - *tasks_file* に記述されたタスクをスケジューラが適当なノードに割り当て
- *tasks_file* のフォーマット

```
タスク名1　[オプション;]　コマンド1
タスク名2　[オプション;]　コマンド2
⋮
```

未知語検出の並列化 (1/3)

- タスク
  1. 各ページを読み込む
  2. 未知語を検出
  3. 検出した未知語を出力
- 1タスクあたり1万ページを処理
  - 約1万のタスク (x00000 ～ x10046)
  - 1タスクあたり約130分
  - 全体で2, 3日

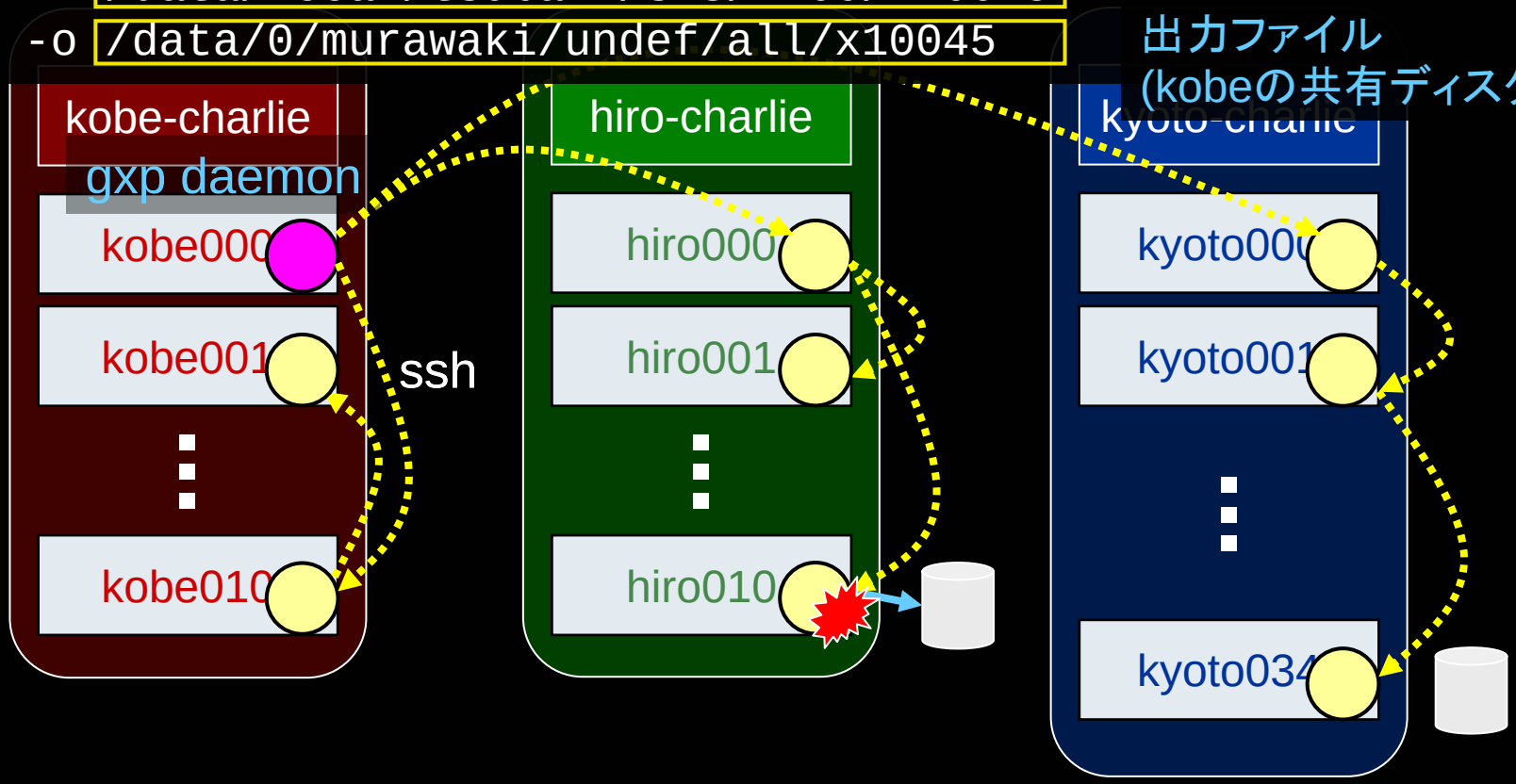
未知語検出の並列化 (2/3)

タスク名　実行ノードを制限 (入力ファイルを持つノード)

```
x10045 :on (kobe009|hiro010|kyoto034); ¥
  sh /home/murawaki/research/lebyr_main.tmp.sh ¥
   -i /data/local/tsubaki/sfs/x100/x10045 ¥
   -o /data/0/murawaki/undef/all/x10045
```

入力ディレクトリ
(ローカルディスク)

出力ファイル
(kobeの共有ディスク)



未知語検出の並列化 (3/3)

- シェルスクリプトの内容
  1. メインプログラム (Perl) を実行
     - 入力はローカルディスク
     - 出力先はローカルの一時ファイル
  2. 一時ファイルを `gzip` で圧縮
  3. 出力ファイルを kobe の共有ディスクに転送
     - kobe 拠点のノードでは `mv`
     - それ以外では `scp`

語彙獲得の並列化 (1/2)

- キー毎にまとめられた用例から並列に語彙獲得
  - 約16万タスク (000000 ～ 162060)
- 入力データにばらつき
  - サイズ: 1KB ～ 200MB
  - 処理時間: 20秒 ～ 数時間
- 大きなタスクから順番に実行
  - 終了をならす

語彙獲得の並列化 (2/2)

- シェルスクリプトの内容
  1. 未知語の用例データを kobe の共有ディスクから転送
  2. メインプログラム (Perl) を実行
     - 出力先はローカルの一時ファイル
  3. 一時ファイルを `gzip` で圧縮
  4. 出力ファイルを kobe の共有ディスクに転送

おわりに

- 自然言語処理における利用事例の報告
  - 大規模データの処理
  - 汎用ツール (gxp) の利用
- 分散ファイルシステムの利用?
  - スケジューラとの連携?