動的な資源スケジューリングが可能な仮想クラスタ上でのMPI実行環境

数理・計算科学専攻　松岡研究室
立薗　真樹
指導教員　松岡　聡

目次

仮想クラスタ技術

- 複数サイトにまたがる資源の統合的利用環境
  - ユーザーからは単一のクラスタ環境として利用可
  - ユーザーごとに独自の環境を用意
- 仮想クラスタの構成
  - 仮想ノード
    - 仮想計算機上のゲストOS
    - サイトによらず同一環境を構築可能
  - 仮想ネットワーク
    - 仮想ノードを結ぶネットワーク
    - 各仮想クラスタごとに固有のネットワークを提供
    - VPN技術によりNATやファイアウォール越えが可能

仮想クラスタでのMPI実行

- MPIアプリケーションの大規模環境での実行要求
  - 科学技術計算等は膨大な計算力を要求

仮想クラスタを用いた複数サイト環境でのMPI実行

- 容易に大規模資源を利用可能
- クラスタ環境と同様にMPIを利用可能
  - NATやファイアウォールによるノード間到達不可の解消
  - ソフトウェア環境を均質

→しかし複数サイトを基盤とするため問題点が

複数サイト利用による問題

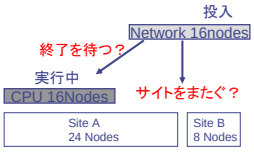
- サイト間ネットワークの存在
  - サイト内と比較して低性能
  - 通信を多用するアプリケーションではボトルネック
- ハードウェア構成の不均質性
  - 各ノードの構成
    - CPU,メモリ,ディスク…

→並列アプリ ケーションにとっての
ボトルネッ クが多数存在

複数ジョブ環境による問題

- 静的なスケジューリングによるスループット低下
  - 資源利用状況が刻一刻と変化
  - 最適な実行資源配置は常に変化
- アプリケーション特性の把握にはプレ実行が必要
  - 総実行時間の増加

→柔軟な資源配置が必要



目次

1. 背景
2. 目的と成果
3. 関連研究
4. 提案
5. 実装
6. 評価
7. まとめ

目的と成果

- 目的
  - 仮想クラスタ技術を用いた複数サイト上でのMPI実行環境の提供
- 成果
  - 実行されるアプリケーションの特性に基づく仮想クラスタの再配置を提案し、プロトタイプを評価
  - MPIアプリケーションのサイト間通信量が複数サイトでの実行時の性能低下に影響することを確認

目次

1. 背景
2. 目的と成果
3. 関連研究
4. 提案
5. 実装
6. 評価
7. まとめ

関連研究 – 仮想クラスタの資源管理

- VioCluster [Xu *et al.* '06]
  - 仮想計算機Xenと仮想ネットワークVIOLINを用いた仮想クラスタ環境
  - 計算資源を監視し適切に仮想ノードを再配置
    - 複数のアプリケーション実行の最適化
  - 構築、再配置時にネットワーク状況は考慮しない

×MPIのようなネットワークに依存するアプリケーションの最適化が困難

関連研究 – 広域対応のMPI実装

- MPI/GXP [Saito et al. '06]
  - 分散環境でサイト間の遅延とアプリケーションでの通信傾向から最適なランクを割り当て
  - サイト間通信を減らすことで高性能を実現
- GridMPI [Matsuda et al. '05]
  - 標準化されたサイト間通信によりローカルのMPI実装の相違を隠蔽して実行可能
  - 複数サイトでの高速な集団通信アルゴリズムを採用

×いずれも動的なノード構成変更には非対応

目次

1. 背景
2. 目的と成果
3. 関連研究
4. 提案
5. 実装
6. 評価
7. まとめ

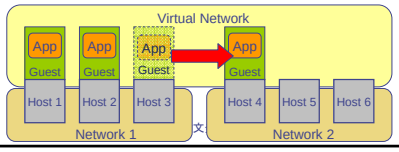
提案

- 仮想クラスタの配置を動的に変更
  - 実行MPIアプリケーションに最適な資源配置へ
  - →システムのスループット向上を目的
- MPIアプリケーションの実行時モニタリング
  - アプリケーションの特性を把握
  - →プレ実行が不要
- サイト間通信量に着目した配置ポリシー
  - サイト間ネットワークの影響を最小化
  - →複数サイトにまたがる資源の有効利用

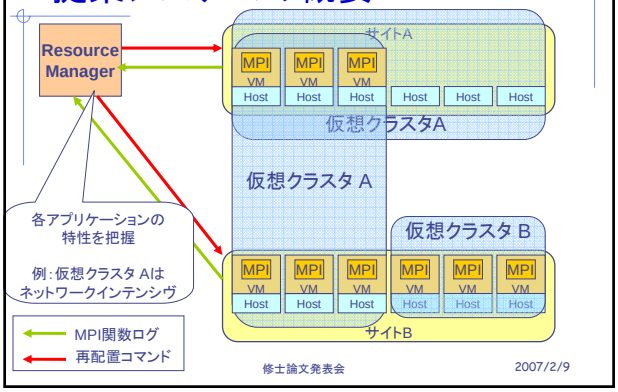
仮想クラスタの再配置

- 仮想クラスタ環境では仮想クラスタの再配置が可能
  - 仮想計算機のゲストOSマイグレーション機能を利用
    - 仮想化により環境に非依存で実行可能
  - サイト間でのネットワークの相違を仮想ネットワークにより吸収

→他サイトへゲストOSのマイグレーションが可能

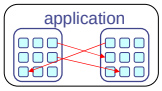


提案システムの概要



仮想クラスタ配置ポリシー

- サイト間通信量に着目したポリシー
  - サイト間通信量：



取得通信ログをもとに
プロセスを2等分し
分割を超える通信をサイト間通信と近似

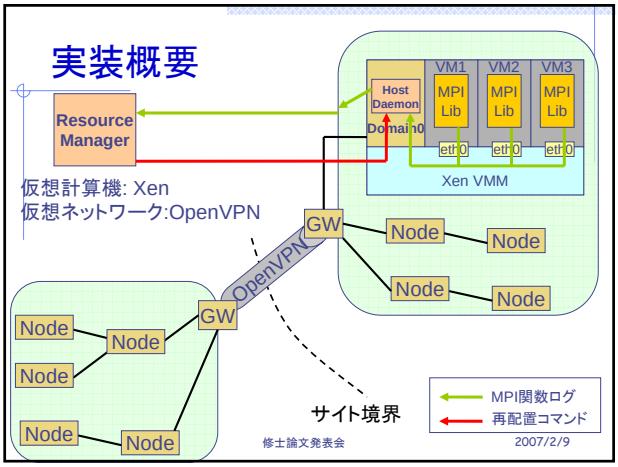
- サイト間通信量を比較
  - 通信量の多いものから単一サイトへ配置
  - 単一サイトへの配置が不可能な場合は複数サイトへ

→通信量の多いアプリケーションを単一サイトへ配置することで
サイト間ネットワークの影響を縮小

目次

1. 背景
2. 目的と成果
3. 関連研究
4. 提案
5. 実装
6. 評価
7. まとめ

実装概要



Resource Manager

- 情報の収集
  - MPI通信関数ログの解析
    - 各MPIプロセスから通信関数呼び出しログを解析
    - サイト間通信量を算出しアプリケーション特性とする
  - 計算資源の利用状況
    - 各ノードのHost daemonにて監視
    - Xenのモニタリング機能を利用
- 仮想クラスタの配置を変更
  - MPIのアプリケーション特性と資源利用状況を考慮
  - 各アプリケーションの最適な実行環境へ再配置

MPI Library

- MPICH-1.2.7p1をベースに実装
- ログの取得
  - ラッパー関数によりMPI関数呼び出しを取得
  - 一定数蓄積後、Resource Managerに送信
- 集団通信の最適化
  - 代表ノード通信によりサイト間通信量を低減 [Kielmann et al. '99]
  - 仮想クラスタ再配置時には新配置にて集団通信トポロジを再構築

目次

評価項目

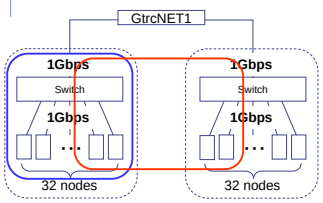
- 評価項目
  - 仮想クラスタ環境でのMPIアプリケーション性能
    - 仮想クラスタ自体の性能
    - サイト間ネットワーク環境の性能への影響
  - 複数種のアプリケーションのサイト間通信と性能低下の関係
    - サイト間通信量と性能低下の関係
  - 動的再配置によるスループットの向上の評価
    - 数種類のジョブ投入に対し提案システムの有効性を評価

評価環境

- 松岡研究室　PrestoIIIクラスタを利用
  - CPU：Opteron 242*2
  - RAM：2GB
  - Network：1000BASE-T
  - Kernel 2.6.12.6
- ベンチマーク
  - MPICH-1.2.7p1
  - Nas Parallel Benchmarks 3.2 クラスC （以降NPBと表記）

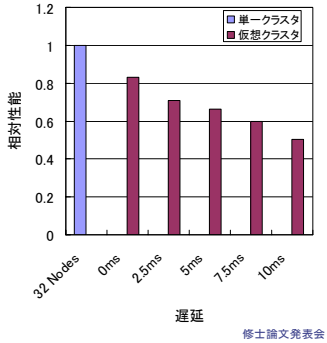
仮想クラスタ環境でのMPI性能

- 2サイト・エミュレート環境上にプロトタイプ環境を構築
- 32ノード単一サイト実行 (青)
  vs 16+16 2サイト実行 (赤)
- 遅延を変化させてLU, CGベンチマークを実行



- 2サイト環境をエミュレート
  - 32ノード＊2サイト
- サイト間にネットワークエミュレータGtrcNET1を挿入
  - 自由に遅延等を設定可能

仮想クラスタ環境でのMPI性能



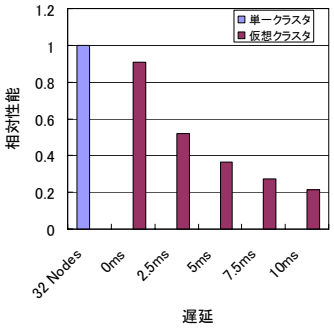
- 仮想クラスタのプロトタイプ環境でのMPIアプリケーション性能
  ※単一クラスタでの実行時に対する相対性能
- 仮想クラスタ環境でサイト間遅延を変化させて性能の変化を計測

LUベンチマーク 32ノード実行
16+16ノードでの配置

サイト間通信量：5.88MB /sec

サイト間遅延5msで65%の性能

仮想クラスタ環境でのMPI性能



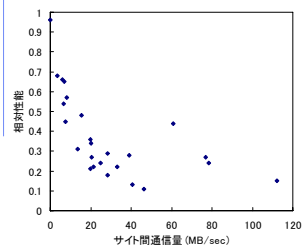
- 仮想クラスタのプロトタイプ環境でのMPIアプリケーション性能
  ※単一クラスタでの実行時に対する相対性能
- 仮想クラスタ環境でサイト間遅延を変化させて性能の変化を計測

CGベンチマーク 32ノード実行
16+16ノードでの配置

サイト間通信量：19.9MB /sec

サイト間遅延5msで約35%の性能

サイト間通信量と性能低下



単一クラスタ実行からの
性能低下とサイト間通信量

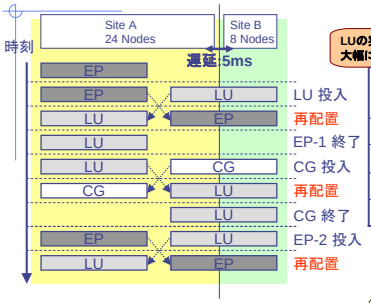
サイト間遅延5ms
以下のアプリを実行

LU, BT, CG, MG, EP
CLASS=B,C
16,32,64ノード

単一クラスタ実行からの
性能低下をプロット

→サイト間通信量と複数サイトでの性能には相関関係あり

提案システムの有効性



LUの実行時間が大幅に削減

実行時間の比較 (秒)

	提案システム	通常環境
LU	420.45	537.38
EP-1	105.28	98.44
EP2	102.67	98.59
CG	118.57	101.97

他の3ジョブは実行時間が微増

平均実行時間が短縮

- 4つのジョブを投入し提案システムと通常環境で実行時間を比較

目次

1. 背景
2. 目的と成果
3. 関連研究
4. 提案
5. 実装
6. 評価
7. まとめ

まとめと今後の課題

- まとめ
  - 仮想クラスタ環境でのMPI実行を提案
    - 仮想クラスタ環境でのMPI性能を評価
    - 単一サイト実行からの性能低下はサイト間通信量に影響されることを確認
  - 仮想クラスタ環境での効率的なMPI実行環境を提案
    - MPIアプリの特性によって実行仮想クラスタを再配置
    - システムの資源使用効率を向上
- 今後の課題
  - より大規模な環境での評価実験
    - 現実のジョブ投入シナリオを基に実際的な評価
  - 一般的なMPI実装との連携のためのAPIの策定
    - Resource Managerとの間で実行ログ、トポロジ情報を

論文発表

- SWoPP '05「仮想計算機を用いて負荷分散を行うMPI実行環境」
  立薗真樹　中田秀基　松岡聡
- SACSIS '06「仮想計算機を用いたグリッド上でのMPI実行環境」
  立薗真樹　中田秀基　松岡聡
- XHPC'06 「Making Wide-Area, Multi-Site MPI Feasible Using Xen VM」 Masaki Tatezono, Naoya Maruyama, Satoshi Matsuoka

アプリケーション特性の把握と分類

- 各ノードのHost daemon、Guest daemonにより情報収集
  - MPIのラッパー関数を作成することで精緻な監視を実現
- ネットワーク・インテンシヴ
  - ノード間通信
    - アプリケーション内の他の並列プロセスとの通信
  - 集団通信
    - Broadcastなどの特殊な通信関数
  - データベース等へのアクセス
    - 大量のデータアクセスが必要な場合
- CPU・インテンシヴ
  - 高CPU使用率

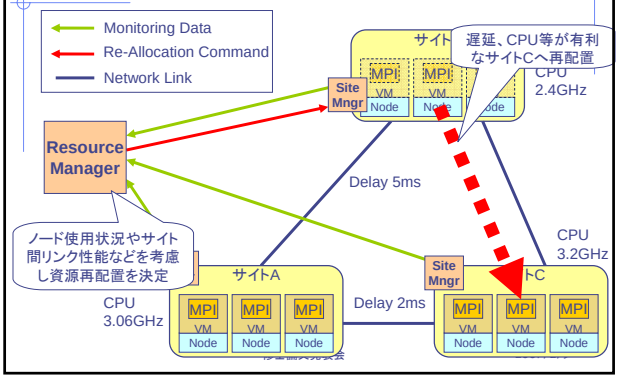
仮想クラスタ配置ポリシー

- アプリケーション特性を考慮した仮想クラスタの配置
  - アプリケーション実行時のサイト間通信量を考慮
    →サイト間通信量の大小により次の二つに分類
    ネットワーク・インテンシヴ
    CPU・インテンシヴ
- 配置ポリシー
  - ネットワーク・インテンシヴ
    - 可能な限り単一サイトへ集約
    - 低性能なサイト間リンクによるボトルネックを回避
  - CPU・インテンシヴ
    - CPU性能の高いノード
      - サイトにはこだわらない
  - 同一特性のアプリケーションでは、サイト間通信量の大小による相対的な特性により配置を決定

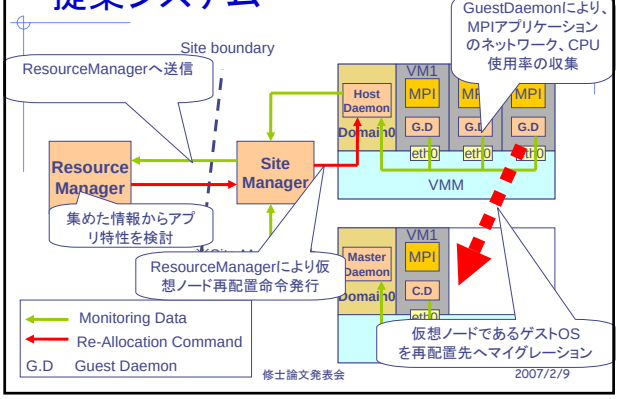
仮想クラスタ上でのMPI実行の問題点

- ハードウェアの不均質性
  - ネットワーク性能、CPU性能…
- サイト間ネットワークの存在
  - 各サイト間は高遅延低バンド幅なWANにより接続
  - 輻輳などによるボトルネックの発生
- 計算資源へのスケジューリング
  - 大規模システムでは使用状況が刻一刻と変化
  - アプリケーションに最適な配置の継続が困難

提案システム



提案システム

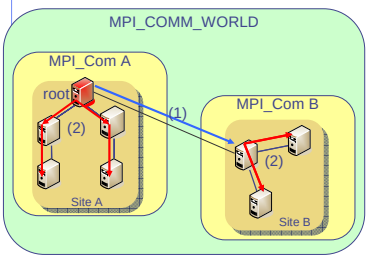


各ベンチマークの許容遅延とサイト間通信量

LUでの2サイト実行での有効性が示された遅延の範囲と
各ベンチマークでのサイト間通信量

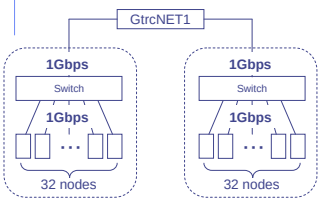
ベンチマーク	許容遅延	サイト間通信量
LU – 16 nodes	0 – 7.5 ms	3.36 MB/sec
LU – 32 nodes	0 – 7.5 ms	5.88 MB/sec
LU – 64 nodes	0 – 2 ms	6.94 MB/sec
CG – 16 nodes	なし	40.7 MB/sec
CG – 32 nodes	なし	19.9 MB/sec
CG – 64 nodes	なし	28.5 MB/sec

サイト間通信量が2サイト実行の性能に影響



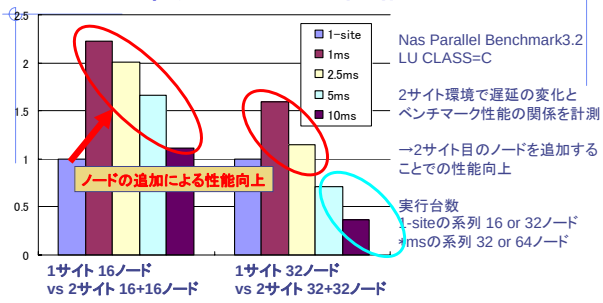
2サイト環境でのMPI性能

- Nas Parallel Benchmarksによる2サイトエミュレーション環境でのMPI性能



- 松岡研究室 PrestoIIIクラスタ
  - CPU：Opteron242 * 2
  - Memory：2GB
  - Network：1000BASE-T
  - Kernel 2.6.12
- サイト間にハードウェアネットワークエミュレータGtrcNET1を使用
  - 自由に遅延等を設定可能
- サイト間通信はOpenVPN2.0.7を使用

2サイト環境でのMPI性能 – NPB LU



Nas Parallel Benchmark3.2
LU CLASS=C

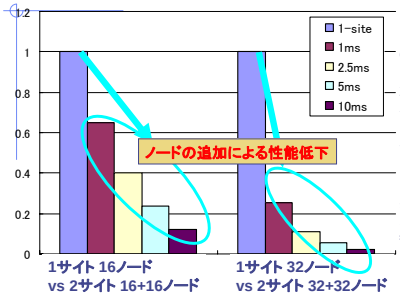
2サイト環境で遅延の変化と
ベンチマーク性能の関係を計測

→2サイト目のノードを追加する
ことでの性能向上

実行台数
1-siteの系列 16 or 32ノード
*msの系列 32 or 64ノード

32ノードでは遅延10msまで性能向上
64ノードでは遅延2.5msまで性能向上、5-10msでは性能低下

2サイト環境でのMPI性能 – NPB CG



Nas Parallel Benchmark3.2
CG CLASS=C

2サイト環境で遅延の変化と
ベンチマーク性能の関係を計測

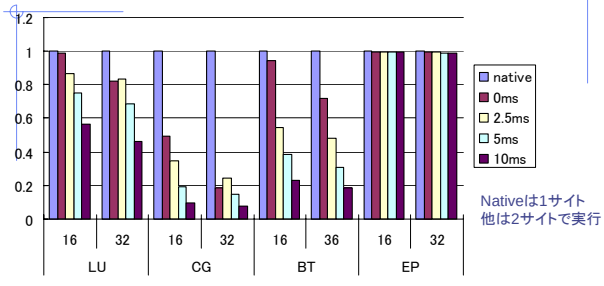
→2サイト目のノードを追加する
ことでの性能向上

実行台数
1-siteの系列 16 or 32ノード
*msの系列 32 or 64ノード

2サイト環境は遅延1msでも性能低下

**→アプリケーションの特性によって
2サイト環境での実行性能に影響**

単一サイトと2サイトでの性能差



Nativeは1サイト
他は2サイトで実行

**遅延挿入による性能低下度合い CG>BT>LU>EP
→サイト間ネットワークの使用特性による性能差**

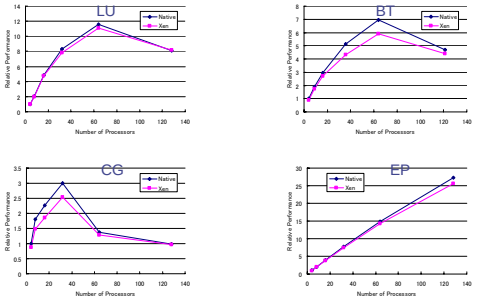
現状のまとめ

- アプリケーション特性により複数サイト実行への影響が変化
  - 2サイト目以降を使用する効果の有無が変わる
  - アプリケーション特性を把握することで最適な実行構成を決定可能
- 仮想クラスタが仮想ノード再配置可能なことに注目
  - 静的に実行資源を決定する必要が無い
    - プレ実行など、ユーザーにコストがかかる性能予測が不要
  - 実行中のアプリケーションを動的に適切な実行構成へ再配置可能
    - 実行中のアプリケーションを監視することで特性がわかる

現状

- 複数サイトでのMPI実行の可能性を検証するベンチマークを実施
  - 異なる問題サイズ、ノード数
  - 複数のVPN実装
  - ハードウェア、ソフトウェアによる遅延エミュレータの使用
  - Xenを用いた仮想環境での性能
- MPIアプリケーションの動作、特性の把握
  - 主にNas Parallel Benchmarks
- 各ノードで動作するClient daemon、Resource Managerを実装中
- 論文発表
  - SWoPP '05「仮想計算機を用いて負荷分散を行うMPI実行環境」 立薗真樹　中田秀基　松岡聡
  - SACSIS '06「仮想計算機を用いたグリッド上でのMPI実行環境」 立薗真樹　中田秀基　松岡聡
  - XHPC'06 「Making Wide-Area, Multi-Site MPI FeasibleUsing Xen VM」 Masaki Tatezono, Naoya Maruyama, Satoshi Matsuoka

Xen&NPB 単一サイトでの性能



Nas Parallel Benchmarkの特性

- 各ベンチマークのサイト間ネットワークの使用頻度とメッセージサイズ(CLASS=C PROCS=32)
  - LU
    - 137回/sec
    - 平均メッセージサイズ　約200Byte
  - CG
    - 78回/sec
    - 平均メッセージサイズ　約18Kbyte
  - BT
    - 29回/sec
    - 平均メッセージサイズ18Kbyte

頻度は他の二つより多いがメッセージサイズが小さい

頻度はLUより少ないがメッセージサイズが大きい

Nas Parallel Benchmarkの特性

	実行時間	Send	Bcast	Allreduce
LU	142.95	1303278	226	
CG	84.29	372599		
BT	185.13	239381	114	
EP				

- 低い遅延であれば2サイトも可能
  - 遅延を考慮したスケジューリングが必要
- しかし、実行前からその特性を知るためにはSample runをしなければならない
- 低遅延な最適環境であれば
- 仮想クラスタなら再配置容易

仮想クラスタ上での MPIアプリケーション実行時の問題

- 不均質なハードウェア
- アプリケーションの特性把握にはプレ実行が必要
  - 通信頻度、CPU利用率
- 複数ユーザーによる共有環境
  - 静的なスケジューリングでは最適な実行が困難
- 最適な資源選択方法が不明
  - サイト間ネットワーク

集団通信の最適化

- グリッド環境のネットワークを考慮したMPI実装

動的なトポロジ変更に

- APIのようなものをつくる？

Nas Parallel Benchmarkの特性

- LU
  - 実行時間 142.95sec
  - Send回数 1303278
    - 195393
  - 平均メッセージサイズ 763Byte
- CG
  - 実行時間 84.29sec
  - Send回数 372599
    - 26564
    - 18Kbyte
  - 平均メッセージサイズ 9000Byte
- BT
  - 実行時間 185.13sec
  - Send回数 239381
    - 内サイト間53570
    - 平均メッセージサイズ18Kbyte
  - 平均メッセージサイズ 23000Byte