BOINC クラスタを利用した KNOPPIX 構築環境の開発

阿部大将† 北川健司† 渡辺義人†
濱野裕樹† 千葉大作† 須崎有康††

1. はじめに

"KNOPPIX"[1] は OS イメージを格納する CD, DVD ディスク内に isolinux, grub 等を利用したブート領域と起動ディスク (minirt.gz), システム領域 ("Compressed loop(CLOOP) デバイス") を持っている.

CLOOP を作成する時間は KNOPPIX の構築時間全体の中で最も大きな割合となり, 全体の九割以上の時間がかかる. 1 これは,CLOOP が連続するサイズ不一致の圧縮ブロックと, 圧縮ブロックのディスク上のオフセットを列挙した配列からなるブロックデバイスであることが主な原因である. ギガバイトオーダの巨大なシステムイメージを圧縮し,CD, DVD ディスクに納まるサイズに変換する作業は多大な CPU リソースを必要とする. さらに,CLOOP の仕様から従来手法では圧縮ブロックがシーケンシャルにすべて揃うまで待つことが要求され, 処理の待ち時間が長大化していた. そこで,CLOOP 作成時間を短縮する方法について検討を行った. 従来手法ではシーケンシャルに実行されていた圧縮プロセスを並列化することで, 圧縮時間の高速化を図った.

並列化されたプロセスを実行する環境として, リソースの有効活用を狙いオフィス環境に PC グリッドを構築し, 遊休 PC の活用を行った. 本稿では,KNOPPIX 構築の高速化についての検討について述べる.

2. 検討

2.1 既存手法の所要時間

既存の KNOPPIX 構築に必要な時間の割合を示すため,CLOOP の作成時間の内訳を計測した. 圧縮対象となったイメージは産総研から配布されている KNOPPIX5.1.1CD-ROM の CLOOP である. CLOOP を展開したイメージを,CLOOP 作成プログラム "create_compressed_fs(以下 ccfs)" を用いて圧縮し, 工程別に所要時間を計測した (表 1).

表 1 CLOOP 作成時間の内訳

処理	所要時間 (sec)	割合 (%)
全体	795.55	100
読み込み	15.10	1.90
圧縮 (zlib)	754.41	94.83
書き出し	10.69	1.34

Debian GNU/Linux 4.0r1, linux kernel:2.6.18
CPU:Athlon64X2 3800+, Mem:3GB

必要となる時間のほとんどが zlib によるデータ圧縮の時間となっている. データ圧縮処理性能は CPU の性能に大きく依存するため,CLOOP の作成についても CPU の性能が大きく影響する.

2.2 並列化

KNOPPIX 構築の高速化の検討として,ccfs をマルチスレッド化した "yet another create_compressed_fs(以下 yaccfs)" を作成した. ccfs と同様に yaccfs でも計測を行った. 計測は表 1 と異なり,4 コアを搭載した Core 2 Quad マシンを利用して行った. 表 2 に,ccfs, yaccfs それぞれの処理時間結果を示す.

yaccfs は ccfs と比較して 37.97%の時間で実行可能であることが分かった. 読み込み, 書き込み時間については yaccfs の方が大幅に時間がかかっているが, これはパイプライン化により並列化された圧縮スレッドの実行待ちが発生したためである.

yaccfs での CPU 稼働時間は実際の実行時間の 3.3 倍となっており, 使用コアすべてを有効に利用しきれていない. これはパイプラインのチューニングの余地があることを示している.

表 2 ccfs の実行時間内訳

	ccfs	yaccfs
処理コア数	1	4
処理	所要時間 (sec)	所要時間 (sec)
全体	477.26	181.25
読み込み	2.66	49.49
圧縮 (CPU 使用時間)	443.09	580.98
圧縮 (実働時間)	443.51	175.12
書き込み	1.80	9.84

KNOPPIX5.1.1CD, linux kernel:2.6.19
CPU:Core2Quad 6600+, Memory: 8GB

2.3 並列処理実行環境の選定

これまでの項で,CLOOP 作成プロセスが極端に CPU 依存であること, 複数の CPU リソースを利用した処理の分散化が有効であることを示した.

yaccfs 等のローカルスレッドを利用した処理の並列化は高い性能を得ることが出来るが,2 コア以上のマルチコアを搭載した高性能マシンが必要になってしまう. また, 特定の計算機にリソースが集中する場合,CLOOP 作成者間で計算待ちが発生してしまい処理の高速化が有効にならない場合もある.

我々のオフィスでは開発用途等のため PC が従業員毎に設置されているが, 昨今の PC はオフィスワーク程度の負荷ではリソースを使いきることが出来ず, 資源を余らせている状態であった.

そこで, このリソースを利用して CPU が必要となる圧縮処理を分散することとした. 遊休 PC リソースの利用法としては, ネットワークを利用しタスクを分散させる PC グリッドの利用が適切であると考えた.

オフィスに設置されているマシンでは MS Windows が最も多く利用されていることから, フレームワークにマルチプラットフォームで実行できる BOINC[2] を採用することとした.

3. 実装

CLOOP の作成で BOINC に任せる部分は, データブロックを圧縮する部分とし, イメージの分割, ブロックデータのまとめは BOINC 外に別プログラムとして実装する. KNOPPIX の CLOOP ブロックのサイズは 64~256KB が目安とされているが, このサイズをそのまま BOINC の一度の仕事量とすると, 通信回数が増大し, 通信に要するオーバヘッドが増大するので,2MB 分の CLOOP ブロックをまとめて一度の通信で送信できるようにした. その他,BOINC の設定等の調整を行った.

4. 評価

BOINC にイメージを処理させた時のサーバ, クライアントの状態を計測した. 計測は, 本評価前の予備評価と性能評価の二通りを実施した. 評価は予備, 性能評価共に同一のタスクを実施した.

具体的な内容としては,KNOPPIX5.1.1CD 日本語版の CLOOP イメージを展開したものを再度圧縮しなおす処理を行った. イメージのサイズは約 1.8GB, ジョブは 2MB 分割で 903 個のワークユニットを実行した. ネットワークは 100BASE-T で構成されている既存のネットワークを利用した. サーバは Debain GNU/Linux 4.0r1,CPU: PentiumD 3.20GHz,Linux Kernel: 2.6.18,Memory:2GB, BOINC 6.1.0(revision:13895), クライアントは数種類のスペックの WindowsXP 端末 12 台,Debian 端末 1 台を利用した.CPU 性能はそれぞれ Pentium4 2.0GHz 以上,Memory は 512MB 以上であり, マルチコア CPU を搭載したマシンは存在しない.

† 株式会社 アルファシステムズ
Alpha Systems, inc
†† 独立行政法人 産業技術総合研究所
Advanced Industrial science and technology

4.1 予備評価

BOINC では, クライアントのジョブの取得はクライアントの状態, ジョブの有無によって自動的に行われる. このため, 計算に参加するクライアントの数は状況によりランダムで, 一定しないことが予測される. そこで, 予備評価として, クライアント側は特別な操作無くこのジョブを消化するのにどの程度の時間が必要なのかを計測した. 図 1 は, 計算の所要時間を棒グラフとして, 計算に参加したクライアントの台数を線グラフとしてプロットしたものである.

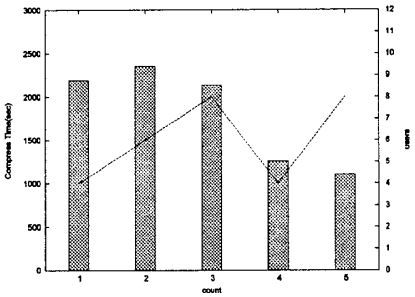

図 1 予備評価:操作無しでの計測

参加したクライアントの台数と性能向上が必ずしも一致しないのは, クライアントが CLOOP 圧縮以外のタスクにリソースを取られている, ネットワークの遅延が発生している等の要因が考えられる.

表 3 に, 実験回毎のクライアント毎の参加回数を示す. 12 台すべてのクライアントが参加した場合の状況は取得できなかった.

表 3 クライアントの参加回数表

回数＼ID	クライアント ID											
	1	2	3	4	5	6	7	8	9	10	11	12
1	-	-	!	!	!	-	-	-	-	-	-	!
2	-	-	!	-	-	!	-	!	-	!	!	!
3	-	!	!	!	!	-	!	!	-	!	-	!
4	!	-	-	!	-	-	-	!	-	-	-	!
5	!	-	!	!	!	-	-	!	-	!	!	!

! … 参加 - … 不参加

4.2 性能評価

予備評価から,BOINC クライアントの計算への参加が一定しないことが分かった. そこで性能評価についてはクライアントを直接操作し, プロジェクトにジョブの取得を要求することとした. 計測は,4,8,12 台それぞれの台数毎に, 圧縮時間を計測した. 計測は数回行いその平均値を取得し, 結果を図 2 にまとめた. 実行させたジョブの内容は予備評価と同様である.

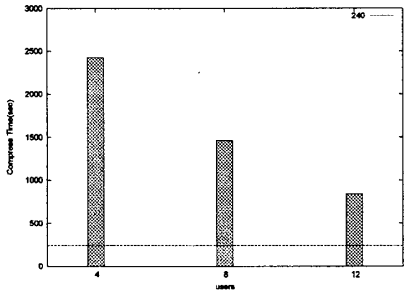

図 2 クライアント数毎の圧縮時間

計測結果から, クライアントの台数が増えるにつれ, 処理時間の短縮を確認する事が出来た. BOINC のスケーラビリティを確認することはできたが, 投入したクライアントの台数が少なく, 性能の限界を確認するには至っていない.

CLOOP 作成時のデータの通信量 (上り 1.8GB, 下り 700MB) と通信環境から, 少なくとも 240 秒程度は通信コストとして利用されていると推察される. ネットワークの通信コストはタスクの処理時間とは別途に必要になるコストであり, 特に数百秒でタスクが終了する CLOOP の作成にとっては非常に大きなオーバヘッドであると言える. このままの推移で推移で通信時間が改善されると,16 台の場合 240 秒を切ることから, 性能限界は 12〜16 台の間にあると考えられる.

図 2 の直線は 240 秒の部分を示し, それを越える部分は BOINC のオーバーヘッド+実際の計測時間+通信のオーバヘッドの値となっている. グラフから, クライアントの増加に従って性能が向上していることが分かる. 計測した範囲でははは台数が性能に結びついている状態であるといえる.

5. 考　　察

今回準備した環境での計測では,100BASE-T 環境に置いてはマルチコア CPU を利用した場合の方が性能が良い結果となった.

性能評価において限界性能を計測することは出来なかったが, 通信速度によって推測される理論値での通信速度のみ限界性能は 250 秒となる. クアッドコアを利用した場合, 全体の限界性能は 181 秒であり,100BASE-T でのデータ通信時間を下回ることは無いため, 現状の我々の環境ではクアッドコア環境を越える性能の PC グリッドを構築することはできない.

Ethernet の標準化タイムライン[3] から,2007 年現在 10GBASE-T の標準化が完了しており,2009 年 11 月には 100GBASE が標準化完了の予定である. このことから,LAN 上の転送速度問題については今後改善されることが期待できる. また, 現状, 通信速度による性能限界を推測することは出来ているが,BOINC フレームワークを利用することによるオーバヘッドについて調査出来ていない状況である. これらの詳細な状況計測するためにはクライアントを今回の性能計測以上に準備し, 本当の性能限界を計測する必要がある.

通信速度の向上に期待する間に,BOINC フレームワークを利用した際の限界性能と性能悪化要因を洗い出し, チューニングを実施していくことが必要となる.

6. おわりに

グリッドフレームワーク BOINC を利用した KNOPPIX 作成時間短縮のための PC グリッドの開発について検討した結果を報告した.

ローカルスレッドでの実効値との性能比較の結果から性能的に PC グリッドの方が劣ることが判ったが, どの程度の性能差があるのかを調査し, 将来的な期待値を出すことで今後の成果につなげることが出来ると考えている.

今後の展開としては,BOINC クライアントに手を入れて, ジョブの投入と共に処理を実行できるようにすることなど, グリッドの性能調整を行っていく. また,BOINC 環境を KNOPPIX に収録し BOINC ネットワークを容易に行えるようにすることで,KNOPPIX 構築以外の作業を行ったり, 他 BOINC プロジェクトに参加したりすることが可能な CD-ROM の作成も行えるようにする予定である.

参 考 文 献

1) KNOPPIX http://www.knoppix.org/
2) D. Anderson. Boinc: A system for public-resource computing and storage. In Fifth IEEE/ACM International Workshop on Grid Computing (GRID'04), p.p. 4-10, Pittsburgh, PA, Nov 2004. IEEE Computer Society (CS) Press.
3) イーサネットの最新動向 -100Gb イーサネット 他-, 西村信治, SACSIS2007 http://www.hpcc.jp/sacsis/2007/SACSIS2007-tutorial-nishimura.pdf