|  |
| --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  **VIỆN CNTT & TT**  logo_128  BÁO CÁO BÀI TẬP LỚN  **TÍNH TOÁN PHÂN TÁN**  **Đề tài:**  **Cài đặt và thử nghiệm MPI trên Rocks Clusters với chương trình bẻ khóa MD5 brute force**  Sinh viên thực hiện: Nguyễn Cao Duy Khương  MSSV: 20142374    Giảng viên hướng dẫn: TS. NGUYỄN HỮU ĐỨC  Hà Nội, 4/2018 |

# Giới thiệu đề tài

## MD5

MD5 (Message-Digest algorithm 5) là một thuật toán mã hóa khá nổi tiếng và kinh điển. Nó được dùng để tạo ra một chuỗi 128 bit duy nhất từ một chuỗi dữ liệu nhập bất kỳ. Tuy nhiên ngày nay người ta có thể dễ dàng giải mã nó chỉ bằng thuật toán tham lam (brute force) và vì thế thường được chỉ dùng để kiểm tra tính toàn vẹn dữ liệu của một tập tin thay vì sử dụng cho mục đích bảo mật.

## MPI

Thuật ngữ MPI là viết tắt của Message Passing Interface, chỉ một dạng giao thức kết nối của máy tính. Nó nằm trong chuẩn de facto cho kết nối giữa các nút chạy một chương trình song song trên bộ nhớ chia sẻ được phân phối. Tập MPI thi hành bao gồm một thư viện các thủ tục sao cho có thể gọi được từ các chương trình Fortran, C, C++ hay Ada. Lợi thế của MPI so với các thư viện cũ là nó vừa thuận tiện (vì MPI thực thi cho hầu hết các kiến trúc bộ nhớ phân phối) vừa nhanh (vì mỗi thủ tục được tối ưu hóa cho phần cứng mà nó đang chạy). Thường được so sánh với PVM và có thể kết hợp nó để tạo thành PVMMPI.

## Rocks Clusters

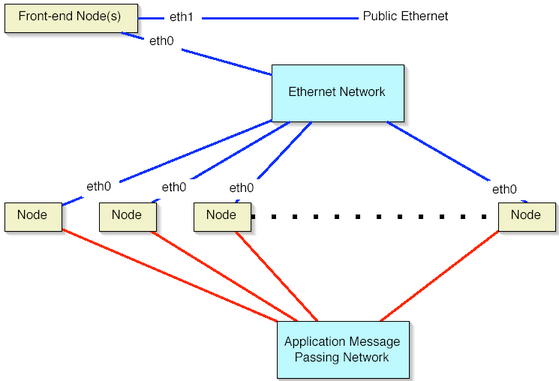
Rocks Clusters là một dự án mã nguồn mở cung cấp cho người dùng với một phân phối Linux dựa trên hệ điều hành CentOS và được thiết kế từ mặt đất lên để cung cấp một cụm đầy đủ tính năng trên một giải pháp cho các cụm CD Red Hat Linux.

Đề tài này tiếp cận phương pháp tính truyền thông điệp của MPI triển khai trên Rocks Clusters nhằm cung cấp một cụm tính toán song song giải quyết bài toán thử nghiệm về bẻ khóa MD5 với chiến lược brute force. Vì thời gian có hạn và kiến thức của bản than còn hạn chế nên đề tài có thể còn nhiều sơ sót, mong được thầy giáo và các bạn đóng góp ý kiến.

# Phương pháp, cài đặt

Phần này phân tích phương pháp và cách cài đặt hệ thống. Đầu tiên chúng ta cần cài đặt cụm tính toán, ở đây theo sự gợi ý của thầy thì em chọn Rocks Cluster.

## Cài đặt cụm Rocks Clusters



Trên đây là sơ đồ cài đặt cụm tổng quát. Em đã cài 3 máy ảo, một frontend và hai compute node. Vì dung lượng máy tính có hạn nên đối với mỗi máy ảo em cấp 36GB ổ cứng và 1GB RAM. Frontend có 2 cổng network kết nối, một ethernet nối ra bên ngoài (eth1) và một kết nối với mạng private (eth0). Các compute node được kết nối vào private network eth0.

## Cài đặt MPI

Phần trước khi cài đặt Rocks, nếu đã add roll HPC (High Performance Computing) từ đầu thì MPICH2 và OPENMPI đã được include sẵn. Hoặc nếu roll chưa được cài từ ban đầu thì ta cũng có thể thêm vào bằng lệnh

# rocks add roll hpc.iso

## Thiếp đặt và lập trình frontend

Trên máy frontend, cần tạo một tài khoản khác root, sử dụng lệnh:

# useradd username

# rocks sync users

Để dễ dàng trong việc chia sẻ tài nguyên giữa máy thật và máy ảo, ta sẽ thiết lập một thư mục chia sẻ chung. Sử dụng lệnh:

# sudo mount -t vboxsf <Thư mục máy thật> <Thư mục máy ảo>

Từ đó có thể dễ dàng chuyển file mã nguồn từ máy thật sang chạy trên máy ảo.

Lúc này trên thư mục chia sẻ chung, ta tạo một file chứa tên các máy compute nodes ví dụ đặt tên là machines

compute-0-0

compute-0-1

Tiếp theo là chạy code bên phần frontend:

$ ssh-agent $SHELL

$ ssh-add

$ mpirun -np 2 -machinefile machines <path file nguồn đã được biên dịch>

## Viết chương trình nguồn

Chương trình viết bằng c++, sử dụng mpi.h (của mpi) và md5.cpp, một file code md5 ngoài để thực hiện thuật toán mã hóa ra dạng mã MD5.

Phương pháp truyền thông điểm điểm đơn giản được áp dụng. Trong đó dựa vào rank của tiến trình ta điều khiển luồng như sau:

* Một tiến trình gốc sẽ nhận thông tin về độ dài đầu vào của mã cần tìm, nó chia cho chính mình và các tiến trình i các miền giá trị để thực hiện vét cạn, tìm ra xâu ban đầu. Hai hàm được áp dụng là MPI\_Send và MPI\_Recv.
  + Đầu tiên tiếng trình gốc gửi dữ liệu cho các tiến trình con sử dụng
  + Tiếp theo nó vét cạn trên miền được giao, nếu tìm được xâu thì trả về
  + Nếu chưa tìm được xâu, tiến trình gốc lắng nghe (MPI\_Recv) từ các tiến trình khác xem chúng tìm ra xâu chưa.
* Các tiến trình con cũng vét cạn trên miền của mình và gửi lại cho tiến trình gốc khi kết thúc hoặc tìm kiếm thành công với MPI\_Send

## Biên dịch và chạy chương trình

Để dịch chương trình .cpp, sử dụng lệnh

# mpic++ main.cpp -o crack

Chương trình biên dịch ra file có tên là crack. Tiếp theo để chạy chương trình trên cụm ta sử dụng lệnh

# mpirun -np 2 -machinefile machines crack

File machines đã được chuẩn bị ở phần trên, ngoài ra thì -np 2 ở đây khai báo số tiến trình sử dụng.

# Thử nghiệm kết quả

Phần này trình bày về kết quả chạy thử nghiệm chương trình với mã md5 có bảng ký tự và độ dài khác nhau.

|  |  |  |  |
| --- | --- | --- | --- |
| Xâu mẫu | Độ dài xâu | Bảng ký tự | Thời gian tìm bản rõ |
| caeaca => 48c6c450f1a4a0cc53d9585dc0fee742 | 6 | abcde | 0.4s |
| caeacab =>  2838cc189672f1fa906e88149fd8ca6e | 7 | abcde | 0.5s |
| adgcbfa => b43abd390066189c08da21ad552dd94c | 7 | abcdefg | 4s |
| bdeadh => fb39801f18908b200bbb413114d1ccda | 7 | abcdefghi | 16s |

# Kết luận

Mặc dù phần thử nghiệm kết quả mới được tiến hành ở mức đơn giản, tuy nhiên nó cho ta thấy tác dụng của cụm tính toán. Sau này cần những bài toán cụ thể hơn, những cụm tính toán phức tạp hơn cùng với phương pháp kiểm thử đánh giá hiệu quả hơn để cho chúng ta có thể nhìn thấy một cách trực quan sức mạnh của một cụm tính toán hiệu năng cao. Đây chỉ là một bài toán mẫu đơn giản, tuy nhiên việc cài đặt một cụm tính toán đơn giản để giải quyết môn bài toán giản đơn tại nhà đã đem lại một cảm giác hết sức thú vị. Em rất mong nhân được sự góp ý của thầy giáo và các bạn trong các phương pháp cài đặt cũng như cách triển khai hệ thống trong tương lai.