

TRƯỜNG ĐẠI HỌC
BÁCH KHOA HÀ NỘI



TRUNG TÂM
TÍNH TOÁN HIỆU NĂNG CAO



NGHIÊN CỨU
CÁC HỆ THỐNG TÍNH TOÁN HIỆU NĂNG CAO
VÀ ỨNG DỤNG MÔ PHỎNG VẬT LIỆU VI MÔ

Trong Khuôn Khổ Hợp Tác Khoa Học Công Nghệ
Theo Nghị Định Thư Với Ấn Độ Giai Đoạn 2004 – 2005

Chủ Nhiệm Đề Tài: PGS. TS. Nguyễn Thanh Thủy

Đề tài nhánh

PHẦN MỀM MÁY CHỦ TÍNH TOÁN
(Tài liệu kỹ thuật: Đặc tả hệ thống)

(1/4)

Hà Nội, 3-2006

5957 - 1

2577106

Lời Nói Đầu

Tài liệu này được thực hiện sau một thời gian làm việc về Tính toán hiệu năng cao của nhóm nghiên cứu High Performance Computing tại Trung tâm Tính Toán Hiệu Năng Cao trường Đại Học Bách Khoa Hà Nội.

Trong tài liệu này, chúng tôi xin trình bày, đặc tả các chức năng của **BKluster** – một hệ thống tính toán song song ghép cụm dựa trên kiến trúc hệ Beowulf và mô hình lập trình truyền thông điệp cùng với bộ phần mềm **BKlusware** - một tập các phần mềm hỗ trợ tối đa nhiều người dùng sử dụng hệ thống BKluster ở nhiều mức độ khác nhau. Nội dung của tài liệu được chia thành 5 chương:

Chương 1: Giới thiệu về mục đích và các hướng tiếp cận của nhóm nghiên cứu khi xây dựng hệ thống.

Chương 2: trình bày về các chức năng chính của hệ thống dưới góc nhìn của người sử dụng.

Các chương 3, 4, 5 trình bày chi tiết về chức năng của 3 hệ thống con tạo thành BKluster: Hệ thống phát triển tích hợp, Hệ thống Thực thi tính toán, Hệ thống giám sát và quản trị. Các chức năng được biểu diễn chủ yếu dưới dạng các biểu đồ Use case của ngôn ngữ mô hình hóa UML.

Mục lục

Lời Nói Đầu	1
Mục lục.....	2
Danh mục hình vẽ	4
CHƯƠNG 1 GIỚI THIỆU HỆ THỐNG	5
CHƯƠNG 2 CÁC CHỨC NĂNG CHÍNH CỦA HỆ THỐNG.....	8
2.1. Môi trường phát triển tích hợp.....	9
2.2. Môi trường thực thi tính toán.....	11
2.3. Bộ công cụ quản trị và giám sát hệ thống.....	12
CHƯƠNG 3 HỆ THỐNG PHÁT TRIỂN TÍCH HỢP	14
3.1. Module phát triển chương trình	14
3.1.1. Chức năng tạo một project mới và chọn một project đã có sẵn	15
3.1.2. Chức năng làm việc với dự án.....	16
3.1.3. Chức năng biên dịch chương trình.....	17
3.2. Module hỗ trợ gỡ rối chương trình - BKPD	18
3.2.1. Pha kết nối.....	19
3.2.2. Pha thiết lập	20
3.2.3. Pha phục vụ.....	22
3.2.4. Pha kết thúc	23
CHƯƠNG 4 HỆ THỐNG THỰC THI TÍNH TOÁN.....	25
4.1. Chức năng Submit Job	27
4.2. Chức năng Alter Job	28
4.3. Chức năng Hold Job.....	29
4.4. Chức năng Release Job	30
4.5. Chức năng Del Job.....	30
4.6. Chức năng Show Info Detail.....	31
CHƯƠNG 5 HỆ THỐNG GIÁM SÁT VÀ QUẢN TRỊ HỆ THỐNG	32
5.1. Module Quản lý gói phần mềm	32
5.2. Quản lý người dùng	33
5.3. Quản lý cấu hình hệ thống	35
5.3.1. Quản lý các nút tính toán	35
5.3.2. Quản lý cluster server	36

5.3.3. Quản lý chế độ làm việc	37
5.4. Module giám sát hoạt động.....	38
5.4.1. Chức năng giám sát tài nguyên hệ thống.....	39
5.4.2. Giám sát thông lượng mạng kết nối	40
5.4.3. Chức năng ghi nhật ký hoạt động	40
5.5. Module Đánh giá hiệu năng hệ thống	41
Tài liệu tham khảo	42

Danh mục hình vẽ

Hình 2-1 Kiến trúc phân tầng hệ thống BKluster	8
Hình 2-2 Kiến trúc phân hệ hệ thống BKluster	9
Hình 3-1 Use case Phát Triển Chương Trình	15
Hình 3-2 Use case Tạo Một Project mới và Chọn Project có sẵn	16
Hình 3-3 Hình 4 2 Use case Làm Việc Với Dự Án.....	17
Hình 3-4 Hình 4 2 Use case Biên Dịch Chương Trình.....	18
Hình 3-5 Kịch bản hoạt động của BKPD	19
Hình 3-6 Kịch bản hoạt động của BKPD - Pha kết nối.....	19
Hình 3-7 Kịch bản hoạt động của BKPD - Pha thiết lập.....	21
Hình 3-8 Kịch bản hoạt động của BKPD - Pha phục vụ.....	22
Hình 3-9 Kịch bản hoạt động của BKPD - Pha kết thúc	24
Hình 4-1 Use case thực thi tính toán	27
Hình 4-2 Use case chức năng Submit Job	28
Hình 4-3 Use case chức năng Alter Job.....	29
Hình 4-4 Use case chức năng Hold Job.....	29
Hình 4-5 Use case chức năng Release Job	30
Hình 4-6 Use case chức năng Del Job	31
Hình 4-7 Use case chức năng Show Infor Detail	31
Hình 5-1 Use case Quản lý gói phần mềm	33
Hình 5-2 Use case quản lý người dùng.....	34
Hình 5-3 Use case quản lý các nút tính toán của hệ thống.....	36
Hình 5-4 Use case quản lý cluster server	37
Hình 5-5 Use case giám sát tài nguyên hệ thống.....	40
Hình 5-6 Use case Đánh giá hiệu năng hệ thống.....	41

CHƯƠNG 1

GIỚI THIỆU HỆ THỐNG

Nhu cầu về các hệ thống tính toán lớn đã tạo ra những hướng nghiên cứu rất mạnh trên nhiều lĩnh vực kiến trúc máy tính, phần cứng và phần mềm. Trong thập niên cuối cùng của thế kỷ trước, hướng tiếp cận ghép các hệ thống nhỏ thành một hệ thống lớn hơn được xem là khả thi, đặc biệt phù hợp với điều kiện kinh tế, công nghệ của các nước đang phát triển.

Với mong muốn đáp ứng các nhu cầu tính toán của các phòng thí nghiệm các viện nghiên cứu trong và ngoài trường Bách Khoa, chúng tôi xây dựng một *hệ thống tính toán song song ghép cụm dựa trên kiến trúc Beowulf và mô hình lập trình truyền thông điệp - hệ thống BKluster*

Trên thực tế, với một tập các máy tính được nối mạng và cài đặt một môi trường truyền thông giữa các tuyến trình như LAMP/MPI, ta đã có thể triển khai các chương trình tính toán song song dựa trên mô hình truyền thông điệp. Tuy nhiên, để hệ thống đó thực sự trở thành một dịch vụ tính toán, một hệ thống tính toán cho phép nhiều người dùng cùng khai thác một lúc cũng như phục vụ cho nhiều người lớp người dùng. Và thực tế cho thấy, để khai thác được các hệ thống phân cụm ở mức thô sơ (chỉ được trang bị môi trường truyền thông giữa các tiến trình), người dùng phải có những kiến thức về hệ thống, hệ điều hành ở mức độ nhất định. Nói cách khác, những người có nhu cầu tính toán thực sự lại gần như không có khả năng sử dụng hệ thống. Hơn nữa, hầu hết việc triển khai của hệ thống không thực

sự khai thác hết hiệu năng của hệ thống, gây lãng phí.

Xuất phát từ quan sát đó, chúng tôi muốn xây dựng hệ thống tính toán phân cụm thành một dịch vụ tính toán, thậm chí là một hệ thống tính toán thực sự hỗ trợ tối đa nhiều lớp người dùng khai thác hệ thống một cách hiệu quả. Xét từ phương diện một hệ thống tính toán, trước hết BKluster cho phép người dùng thực hiện một tác vụ tính toán. Hơn thế, BKluster còn phải là một hệ đa nhiệm, đa phiên làm việc, nghĩa là cho phép nhiều người cùng khai thác hệ thống cùng một lúc một cách độc lập. Ngoài ra, chúng tôi cũng đề xuất ra các công cụ hỗ trợ việc phát triển một chương trình tính toán được thực thi trên BKluster. Chúng tôi kỳ vọng các công cụ này không chỉ hữu ích với những người chưa thực sự quen với việc phát triển chương trình song song mà nó trở thành một công cụ chuẩn trong việc xây dựng các chương trình tính toán theo chuẩn MPI. Không chỉ dừng ở việc hỗ trợ người dùng, chúng tôi cũng đưa ra những công cụ giúp những người quản trị hệ thống có khả năng nắm bắt toàn bộ thông tin hệ thống cũng như thiết lập các tham số giúp hệ thống hoạt động một cách hiệu quả nhất. ***Tập hợp các phần mềm giúp BKluster trở thành một hệ thống tính toán gọi là BKlusterware.***

Đứng trên phương diện người dùng, BKluster hỗ trợ sử dụng nhiều mức, người dùng thuần túy, người phát triển chuyên nghiệp và người quản trị hệ thống. Người dùng thuần túy khai thác hệ thống dựa trên các phần mềm tính toán có sẵn hoặc đưa ra yêu cầu tính toán dựa trên một ngôn ngữ đặc tả của hệ thống đề ra. Đối với nhóm người dùng này, khái niệm song song là trong suốt. Hệ thống cũng cho phép người dùng phát triển các phần mềm tự viết các chương trình tính toán nhằm tối ưu hoá dựa trên đặc trưng của từng bài toán cụ thể. Các chương trình được phát triển theo mô hình lập trình truyền thông điệp. Tuy nhiên, ở mức độ này người dùng cũng có thể bỏ qua một số vấn đề phức tạp trong việc tính toán song song như việc phân tải, sắp đặt các chương trình song song trên các nút tính toán. Nói cách khác, sự phức tạp trong tổ chức và các thành phần hệ thống đã được che lấp giúp người dùng nhìn BKluster như một máy tính nhiều bộ vi xử lý. Lớp người

dùng cuối cùng – người quản trị mạng buộc phải nhìn hệ thống một cách chân thực – một tập các máy tính được “liên kết” dựa trên môi trường truyền thông. Các công cụ quản trị và theo được xây dựng không phải để che đậy sự phức tạp của hệ thống. Nó chỉ giúp người dùng giảm bớt khối lượng công việc phải làm của người quản trị cũng như hiện thị một cách trực quan về hệ thống, giúp người quản trị có những quyết định đúng đắn.

Trong các phần tiếp sau, chúng tôi sẽ mô tả về hệ thống theo trình tự sau:

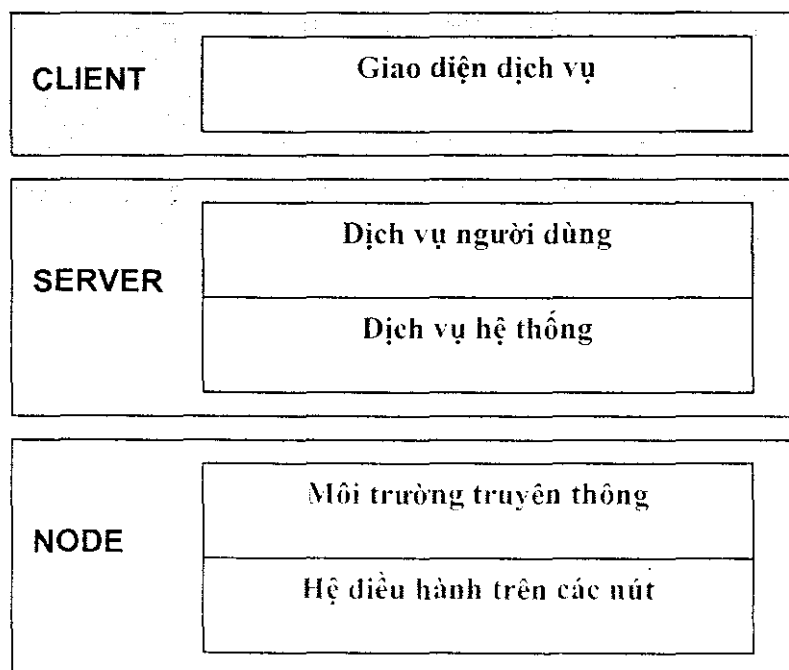
Chương 2 trình bày các chức năng của hệ thống. Các chức năng được mô tả từ phương diện người dùng. Nội dung chủ yếu của chương đề cập tới môi trường phát triển, môi trường tính toán và bộ công cụ quản trị và giám sát hệ thống. Thực chất đây chính là các thành phần cơ bản thuộc tầng trên cùng (tầng client) của 3 hệ thống con trong BKluster: hệ thống phát triển chương trình, hệ thống thực thi và hệ thống quản trị, giám sát hệ thống. Các đặc tả về 3 hệ thống này sẽ được đề cập tới trong các chương tiếp sau.

Chương 3, 4 và 5 đề cập tới các luồng trao đổi thông tin giữa các thành phần trong từng hệ thống.

CHƯƠNG 2

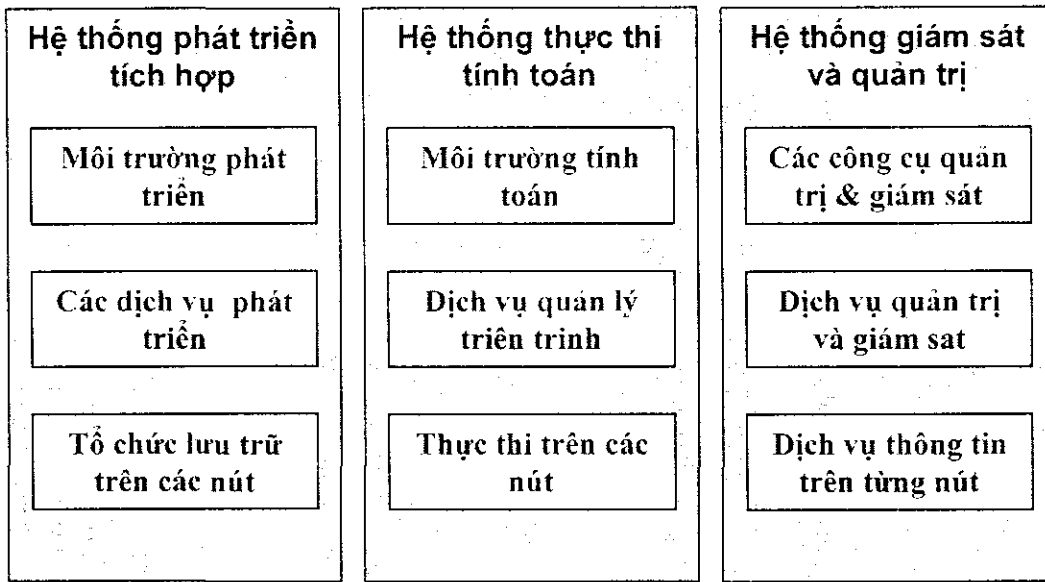
CÁC CHỨC NĂNG CHÍNH CỦA HỆ THỐNG

Chức năng của hệ thống được nhìn nhận qua tầng trên cùng của từng hệ thống con. Hình vẽ sau mô tả sơ qua kiến trúc phân tầng của cả hệ thống BKluster và cũng chính là kiến trúc phân tầng chung cho cả ba hệ thống con: hệ thống phát triển tích hợp, hệ thống thực thi tính toán, hệ thống giám sát và quản trị.



Hình 2-1 Kiến trúc phân tầng hệ thống BKluster

Nếu hình vẽ trên, phân chia hệ thống BKluster theo chiều ngang thì hình vẽ sau mô tả việc phân chia BKluster theo chiều dọc. Nó mô tả, kiến trúc các phân hệ trong BKluster.



Hình 2-2 Kiến trúc phân hệ hệ thống BKluster

Người dùng khai thác hệ thống thông qua 3 giao diện:

- Môi trường phát triển tích hợp
- Môi trường tính toán
- Bộ công cụ quản lý và giám sát

Hệ thống hỗ trợ hai dạng kết nối giữa các dịch vụ hệ thống và dịch vụ người dùng: kết nối cục bộ và kết nối từ xa. Dưới hình thức kết nối cục bộ, tầng dịch vụ người dùng đóng vai trò là shell của hệ thống. Dưới hình thức kết nối từ xa theo mô hình client- server, tầng dịch vụ người dùng thực chất là các client của nhiều dịch vụ tích hợp.

2.1. Môi trường phát triển tích hợp

Môi trường phát triển bao gồm một tập các công cụ hỗ trợ người sử dụng trong tất cả các pha trong quá trình viết mã cho một chương trình bao gồm: soạn thảo và

quản lý mã nguồn, biên dịch tập mã nguồn thành chương trình thực thi và tìm lỗi chương trình.

Môi trường phát triển cũng giúp người dùng có hai hướng tiếp cận để xây dựng chương trình tính toán:

- Xây dựng chương trình tính toán dựa theo mô hình lập trình truyền thống điệp .
- Xây dựng chương trình thông qua đặc tả các thao tác tính toán dựa trên các phép toán đại số bằng ngôn ngữ PCS.

Theo hướng tiếp cận thứ nhất người dùng phải có kiến thức nhất định về lập trình song song đặc biệt là mô hình lập trình truyền thống điệp. Với những người lập trình chuyên nghiệp như vậy, các công cụ của hệ thống trợ giúp họ trong việc soạn thảo mã lệnh sửa, quản lý các file mã nguồn trong một dự án và biên dịch chương trình bằng một lệnh duy nhất. Hướng tiếp cận thứ 2 dành cho những người không có nhiều kiến thức về lập trình song song nói riêng và tin học nói chung. Họ sẽ mô tả bài toán của mình thông qua các ma trận và các phép toán trên đó. Thứ tự của các phép toán cũng được mô tả thông qua một số lệnh lập và điều kiện đơn giản. Việc biên dịch hệ thống gắn liền với việc “song song hóa bài toán”.

Hiện tại, module gỡ rối chương trình mới chỉ hỗ trợ những người lập trình chuyên nghiệp. Nói cách khác, module này mới chỉ áp dụng cho các chương trình được xây dựng theo hướng tiếp cận đầu tiên. Với module này, người dùng có thể:

- Xác định trạng thái hiện tại của mỗi tiến trình
- Xem nội dung các thông điệp đã trao đổi
- Điều khiển luồng chương trình

Việc xác định trạng thái hiện tại của từng tiến trình cục bộ được thực hiện bằng cách xem giá trị của các biến trong các chương trình đó tại thời điểm mong muốn. Thời điểm được xác định sẽ được đặt các break point thông qua giao diện của cửa sổ soạn thảo mã nguồn. Người dùng cũng có thể xem xét nội dung thông điệp thông qua các biến vùng đệm được tạo ra trước và sau mỗi khi gửi và nhận thông

diệp trên các tiến trình. Cuối cùng, người dùng có theo dõi luồng chương trình bằng cách sử dụng các break point hợp lý và cơ chế chạy, dừng chương trình tại các break point đã định.

Trong môi trường tích hợp, chúng tôi cũng đề xuất môi trường kiểm thử cho phép kiểm thử các chương trình trong quá trình xây dựng. Ở mức độ đơn giản nhất, môi trường này cho phép thực thi trên hệ thống và cho phép thu thập kết quả trả lại cho người dùng. Về mặt chức năng, môi trường kiểm thử chương trình gần giống môi trường thực thi tính toán. Tuy nhiên, việc thu thập kết quả được thực hiện ở mức thấp và thô hơn. Nghĩa là kết quả chạy trên từng nút được gửi về mà không có sự tổng hợp.

2.2. Môi trường thực thi tính toán

Môi trường thực thi gồm các công cụ và ứng dụng giúp người dùng khai thác khả năng tính toán của hệ thống. Người dùng khai thác hệ thống dưới hai hình thức:

- Đề trình yêu cầu tính toán bằng công cụ đề trình công việc
- Sử dụng các ứng dụng đã cài đặt sẵn

Với một số ứng dụng có sẵn, người dùng có thể tận dụng khả năng tính toán để giải quyết một số bài toán với kích thước lớn trong một số lĩnh vực như tối ưu phi tuyến, phân tích dữ liệu và mô hình hóa. Đây là những ứng dụng được xây dựng chuyên biệt cho hệ thống và phục vụ cho các mục đích tính toán chung trong một phạm vi rộng. Các lĩnh vực được chọn mang tính phổ biến nhưng không đảm bảo phù hợp cho tất cả các bài toán của người dùng. Điều này đòi hỏi phải có những bước tiền xử lý đầu vào cũng như xử lý kết quả cho phù hợp với các ràng buộc mà ứng dụng yêu cầu.

Ngoài việc sử dụng các ứng dụng cụ thể, người dùng có thể đề đạt một yêu cầu tính toán của riêng mình thông qua công cụ đề trình công việc. Một yêu cầu tính toán được mô tả bởi 2 phần chính:

- Tập các chương trình thực thi

- File đặc tả các tham số cấu hình tính toán

Tập các chương trình mô tả quá trình tính toán, trong đó, mỗi tiến trình cụ thể được tạo ra là một thể hiện của một chương trình trong tập này. Các chương trình được biên dịch ở dạng nhị phân có khả năng thực thi trên hệ thống. Trong trường hợp, người dùng có chương trình mã nguồn, họ có thể sử dụng môi trường phát triển để tạo ra các chương trình thực thi. Hướng tiếp cận được khuyến cáo vì nó đảm bảo chương trình có thể chạy tốt trên hệ thống.

File đặc tả tham số cấu hình tính toán chứa các thông tin để thực thi chương trình trong đó quan trọng là số thể hiện cụ thể của mỗi chương trình và các thể hiện này được thực thi trên bao nhiêu nút tính toán. Bộ các tham số được đề xuất dựa trên các tham số yêu cầu cho một công việc của PBS (một số tham số của PBS đã được cụ thể hóa theo cấu hình của hệ thống).

Các yêu cầu tính toán được đề xuất thông qua một giao diện, trong đó tập các chương trình thực thi có thể được lựa chọn trên thư mục làm việc hiện hành của người dùng trên BKluster hoặc được upload từ hệ thống cục bộ. Các tham số sẽ được điền vào bởi người dùng hoặc do hệ thống tự xác định nếu người dùng chọn cấu hình mặc định

2.3. Bộ công cụ quản trị và giám sát hệ thống

Bộ công cụ quản trị giám sát hệ thống giúp người quản trị nắm bắt toàn bộ các thông tin bên trong hệ thống. Đây là công cụ giành cho người quản trị, nó đưa lại một cái nhìn về hệ thống khác hẳn hai môi trường tính toán và phát triển chương trình. Hai môi trường trên cố gắng che lấp sự phức tạp phía dưới của hệ thống, tính phân tán và các yếu tố cấu thành hệ thống phân cụm là trong suốt với người dùng. Người dùng nhìn nhận hệ thống như một “máy tính” thông thường. Ngược lại, để có thể khai thác hiệu quả bộ công cụ hiểu rõ các thông tin được cung cấp, người quản trị phải có những hiểu biết về hệ thống phân cụm. Bộ công cụ được đề xuất để đơn giản hóa các thao tác của người quản trị và hiện thị thông tin một cách

trực quan.

Các thông tin về hệ thống được chia làm hai lớp

- Các thông tin tĩnh
- Các thông tin động

Các thông tin tĩnh chủ yếu mô tả cấu hình hệ thống bao gồm số lượng nút, cấu hình từng nút, các tham số hệ thống và người dùng hệ thống. Các thông tin này có tính ổn định cao. Cùng với khả năng thu thập các thông tin mang tính quản lý, thông tin về hiệu năng hệ thống cũng được cung cấp thông qua một bộ đánh giá hiệu năng của hệ thống.

Các thông tin động phục vụ việc giám sát hệ thống, các thông tin này biến thiên liên tục trong khoảng thời gian nhỏ. Các thông tin này bao gồm, số công việc đang được thực thi, số tiến trình trên mỗi nút và tài nguyên hiện tại được sử dụng. Tài nguyên ở đây đề cập chủ yếu tới phần trăm sử dụng CPU, bộ nhớ trong và thông lượng mạng.

Ngoài việc cung cấp một cách tương đối đầy đủ và toàn diện các thông tin liên quan tới hệ thống, bộ công cụ cũng cho phép người quản trị thao tác trên các thông tin này, đặc biệt là các thông tin quản lý. Các thao tác chủ yếu là thêm, bớt, tìm kiếm một nút tính toán, xác lập các tham số hệ thống.

CHƯƠNG 3

HỆ THỐNG PHÁT TRIỂN TÍCH HỢP

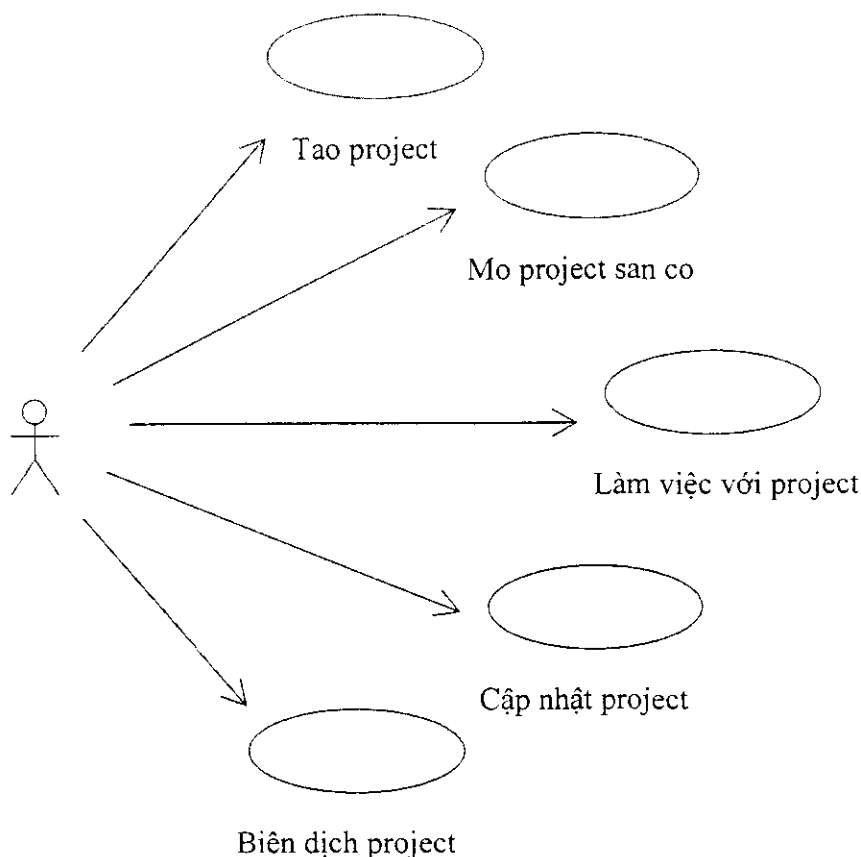
Hệ thống con phục vụ phát triển chương trình chạy trên hệ thống đảm nhận ba nhiệm vụ lớn: quản lý và hỗ trợ soạn thảo mã nguồn, biên dịch chương trình và hỗ trợ tìm lỗi. Ở tầng người dùng, tất cả các dịch vụ trên được cung cấp thông qua một giao diện duy nhất là một môi trường phát triển tích hợp.

Trên thực tế, môi trường phát triển tích hợp ở tầng client được cấu thành từ hai thành phần có các chức năng tách biệt và có thể hoạt động hoàn toàn độc lập. Do vậy, nếu xem xét hệ thống theo khía cạnh các thành phần cấu thành, hệ này có thể phân chia thành hai thành phần chính tương ứng với hai module chính và các môi trường đóng vai trò là phương tiện tương tác của người dùng với hệ thống. Hai module chính của hai thành phần gồm: module quản lý và biên dịch chương trình chương trình và module tìm lỗi.

3.1. Module phát triển chương trình

Phát triển các chương trình tính toán là một trong những yêu cầu không thể thiếu giúp hệ thống có tính mở. Bằng việc tự phát triển các chương trình tính toán phục vụ các mục đích riêng của mình, người dùng sẽ khai thác hệ thống một cách hiệu quả hơn, đồng thời việc đó cũng làm tăng thêm tính năng cho hệ thống.

Module phát triển chương trình hỗ trợ đầy đủ các giai đoạn để xây dựng một chương trình tính toán với các chức năng chính được mô tả trong hình vẽ sau:

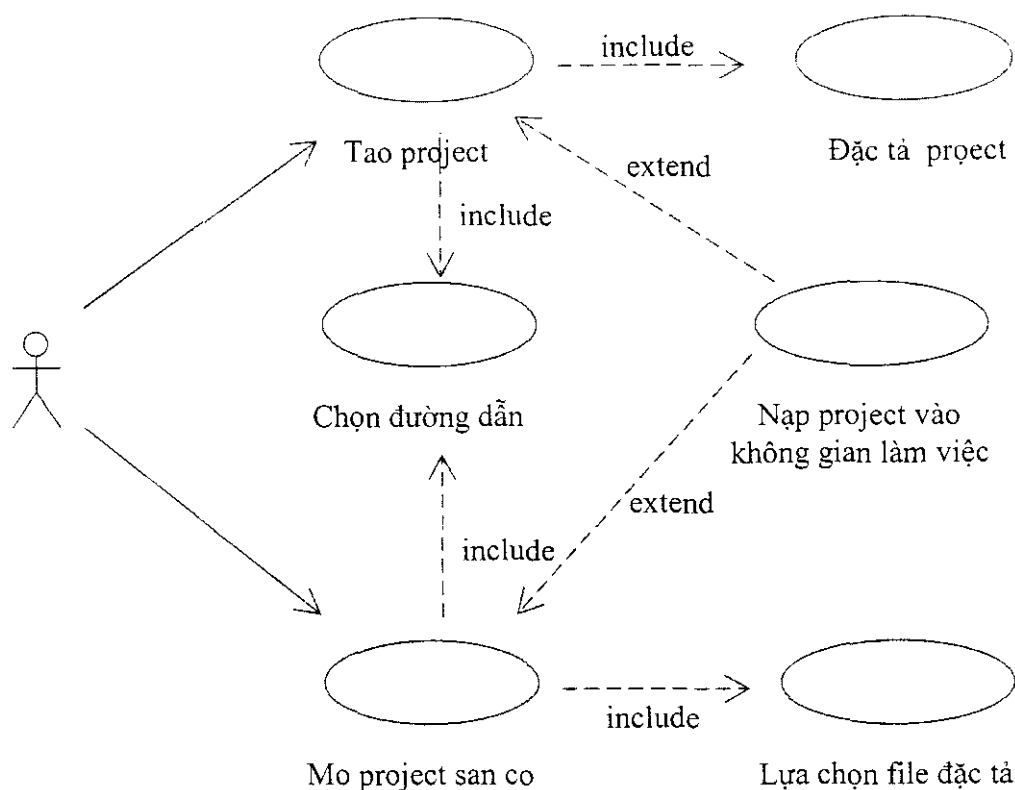


Hình 3-1 Use case Phát Triển Chương Trình

Lưu ý rằng, trong 5 chức năng trên thì chỉ duy nhất chức năng “Làm việc với project” là không có sự tương tác với cluster server, chức năng này tương tác với các file trên hệ thống file cục bộ. Ngược lại, 4 chức năng còn lại đều tương tác với cluster server, việc xử lý thông tin được thực hiện chủ yếu trên server, client chỉ đóng vai trò tựa như một ftp client. Tuy nhiên, việc tương tác với server hay không và thông tin được xử lý ở đâu là hoàn toàn trong suốt với người dùng.

3.1.1. Chức năng tạo một project mới và chọn một project đã có sẵn

Hai chức năng này có nhiều điểm khá tương đồng. Đứng từ phía người dùng, kết thúc việc kích hoạt chức năng này, họ kỳ vọng có một project được mở trên màn hình làm việc. Nói cách khác, project được nạp vào không gian làm việc và sẵn sàng cho các thao tác tiếp sau:

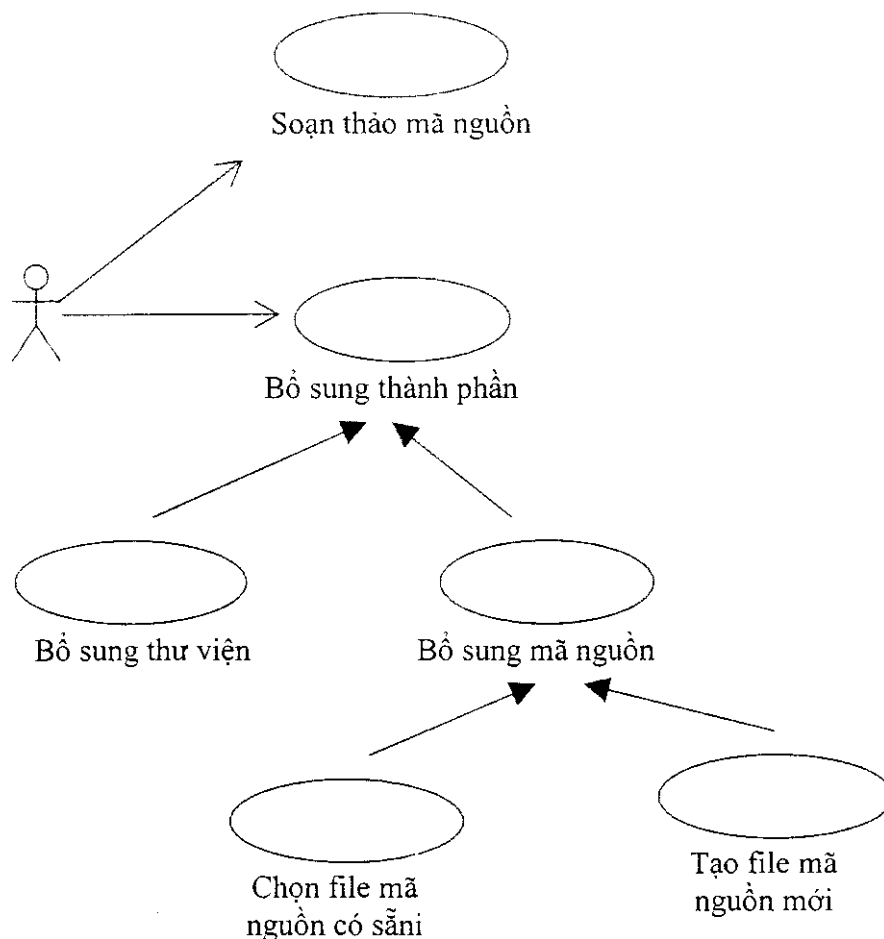


Hình 3-2 Use case Tạo Một Project mới và Chọn Project có sẵn

3.1.2. Chức năng làm việc với dự án

Đây là nhóm chức năng lớn giúp người dùng thật sự tạo ra các thành phần cụ thể cho chương trình. Khi làm việc với một dự án người dùng sẽ có hai thao tác chủ yếu:

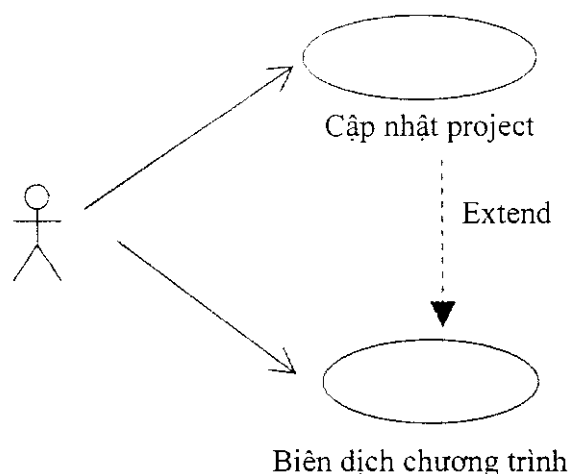
- Bổ sung thêm các thành phần vào dự án, các thành phần này có thể là một file mã nguồn, file header hay thư viện
- Soạn thảo mã nguồn chương trình.



Hình 3-3 Use case Làm Việc Với Dự Án

3.1.3. Chức năng biên dịch chương trình

Đứng từ góc nhìn người dùng, thao tác này cực kỳ đơn giản. Người dùng chỉ cần đưa lệnh biên dịch với project đang làm việc. Khi đó, thông qua môi trường làm việc, yêu cầu đó sẽ được gửi lại cho server và server thực hiện việc biên dịch đối với file project tương ứng. Chức năng này có duy nhất một ràng buộc là chương trình đã được cập nhật lên server. Trong trường hợp ngược lại, người dùng sẽ được yêu cầu cập nhật các sửa đổi trong project



Hình 3-4 Use case Biên Dịch Chương Trình

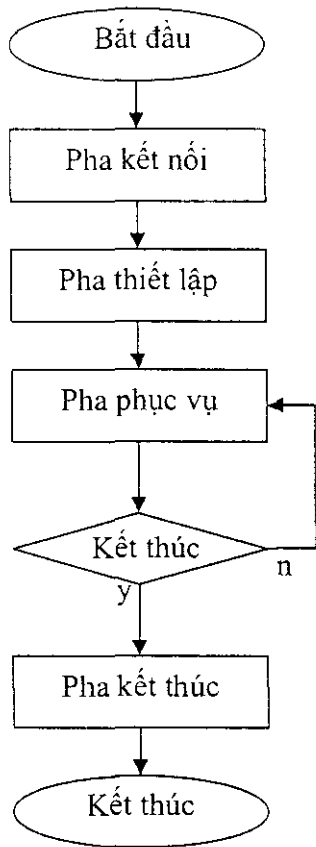
3.2. Module hỗ trợ gỡ rối chương trình - BKPD

Module tìm lỗi cho phép người dùng phát hiện các lỗi logic một cách hiệu quả. Đứng từ phương diện người dùng, module là một chương trình hỗ trợ tìm lỗi của chương trình viết theo chuẩn MPI. Hơn thế, chương trình cho phép người dùng thực hiện các thao tác từ xa. Nói cách khác, người dùng làm việc trên máy đơn mà vẫn có thể kiểm thử trên chương trình trên một hệ thống phân cụm – hệ thống song song thực sự.

Xét về mặt chức năng, hệ thống cho phép người dùng thực hiện các thao tác

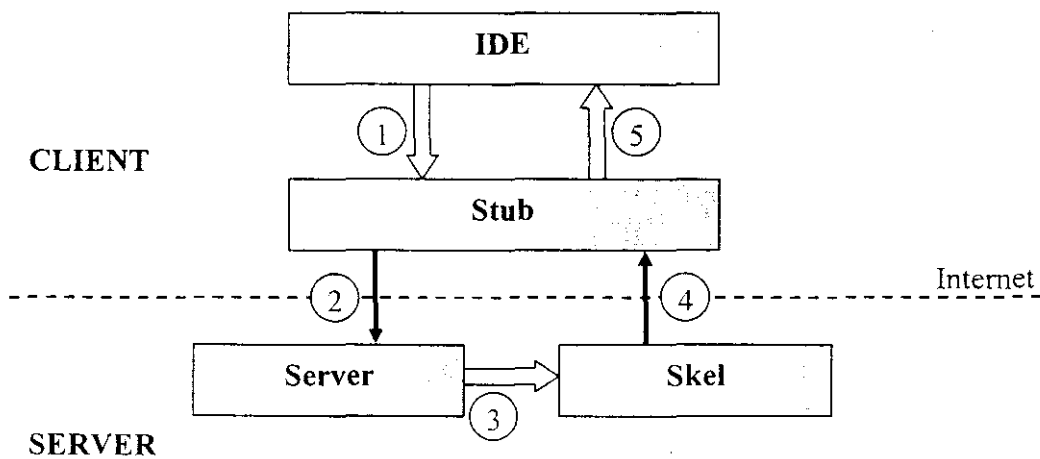
- Xem giá trị các biến trên từng tiến trình đơn
- Điều khiển luồng chương trình của một hoặc nhiều tiến trình

Kịch bản hoạt động của BKPD bao gồm 4 pha chính : pha kết nối, pha thiết lập, pha phục vụ và pha kết thúc.



Hình 3-5 Kịch bản hoạt động của BKPD

3.2.1. Pha kết nối



Hình 3-6 Kịch bản hoạt động của BKPD - Pha kết nối

- 1 Người dùng sử dụng IDE gọi các hàm của Client communicator kết nối đến server.
- 2 Client communicator kết nối đến server.
- 3 Server xác nhận người dùng, sau đó sinh ra tiến trình Skel phục vụ người dùng.
- 4 Skel gửi thông tin kết nối thành công về cho Client communicator. Thông tin xác nhận chính là số nút tính toán có tại trung tâm.
- 5 Client communicator thông báo về cho người dùng thông qua IDE.
- 6 Lúc này hệ thống sẵn sàng phục vụ người dùng.

Các ngoại lệ có thể xảy ra :

- Client communicator không kết nối được đến server (vấn đề đường truyền mạng)
- Server xác nhận người sử dụng không có quyền.

3.2.2. Pha thiết lập

Trạng thái :

Client và Skel đã kết nối, sẵn sàng chờ nhận yêu cầu của người dùng : tên chương trình cần chạy, số nút chạy.

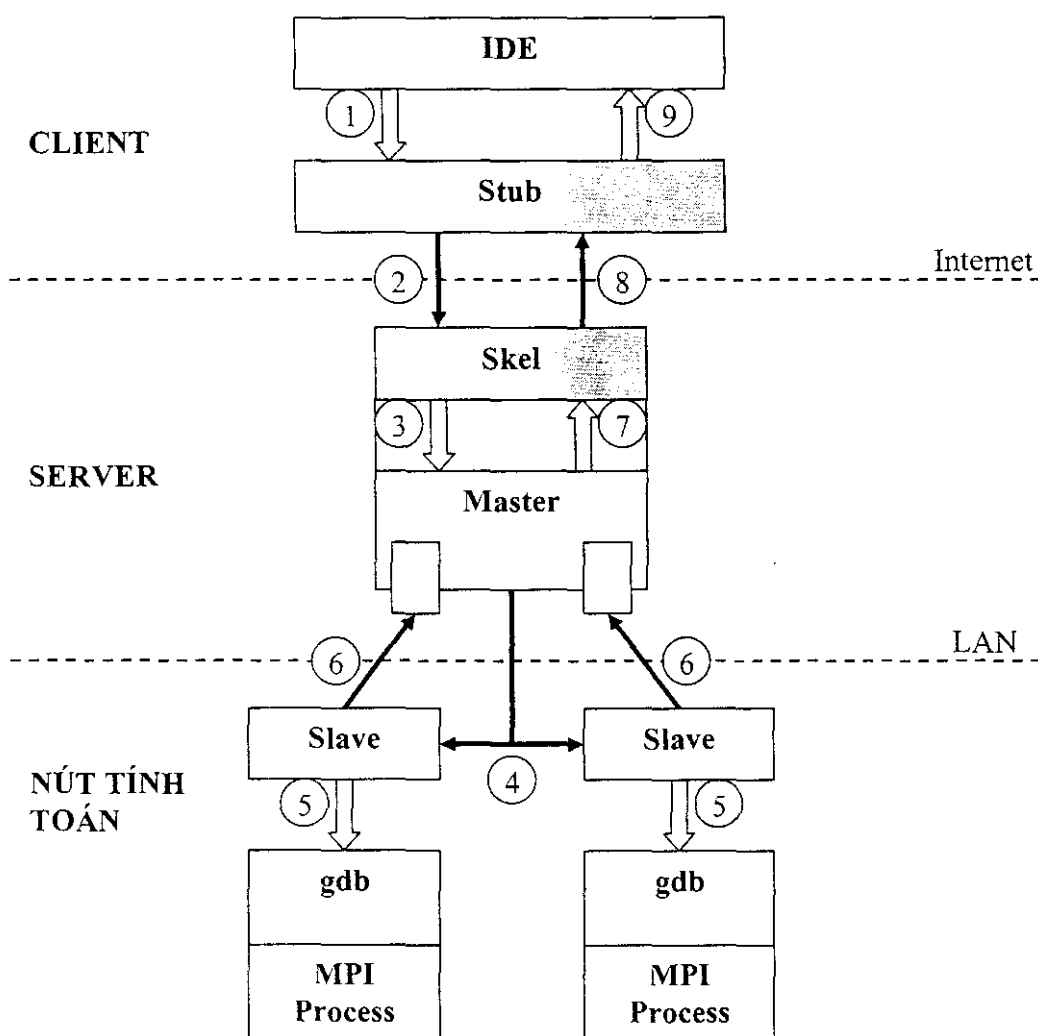
Các bước :

- 1 Người sử dụng thông báo chương trình muốn gỡ lỗi và số nút chạy cho client communicator thông qua IDE.
- 2 Client communicator truyền thông tin đến Skel.
- 3 Skel khởi tạo Master, sau đó gọi hàm API của Master yêu cầu khởi tạo môi trường gỡ lỗi.
- 4 Master triệu gọi các Slave trên các nút tính toán.
- 5 Các Slave gọi gdb, khởi sinh các tiến trình MPI song song.
- 6 Slave thiết lập kết nối đến Master và gửi thông tin thông báo khởi tạo thành công về cho Master.
- 7 Master nhận thông báo từ slave, sau đó trả về kết quả cho Skel.

- 8 Skel truyền kết quả về cho Client communicator.
- 9 Client communicator trả về cho IDE để hiển thị thông báo khởi tạo môi trường thành công.

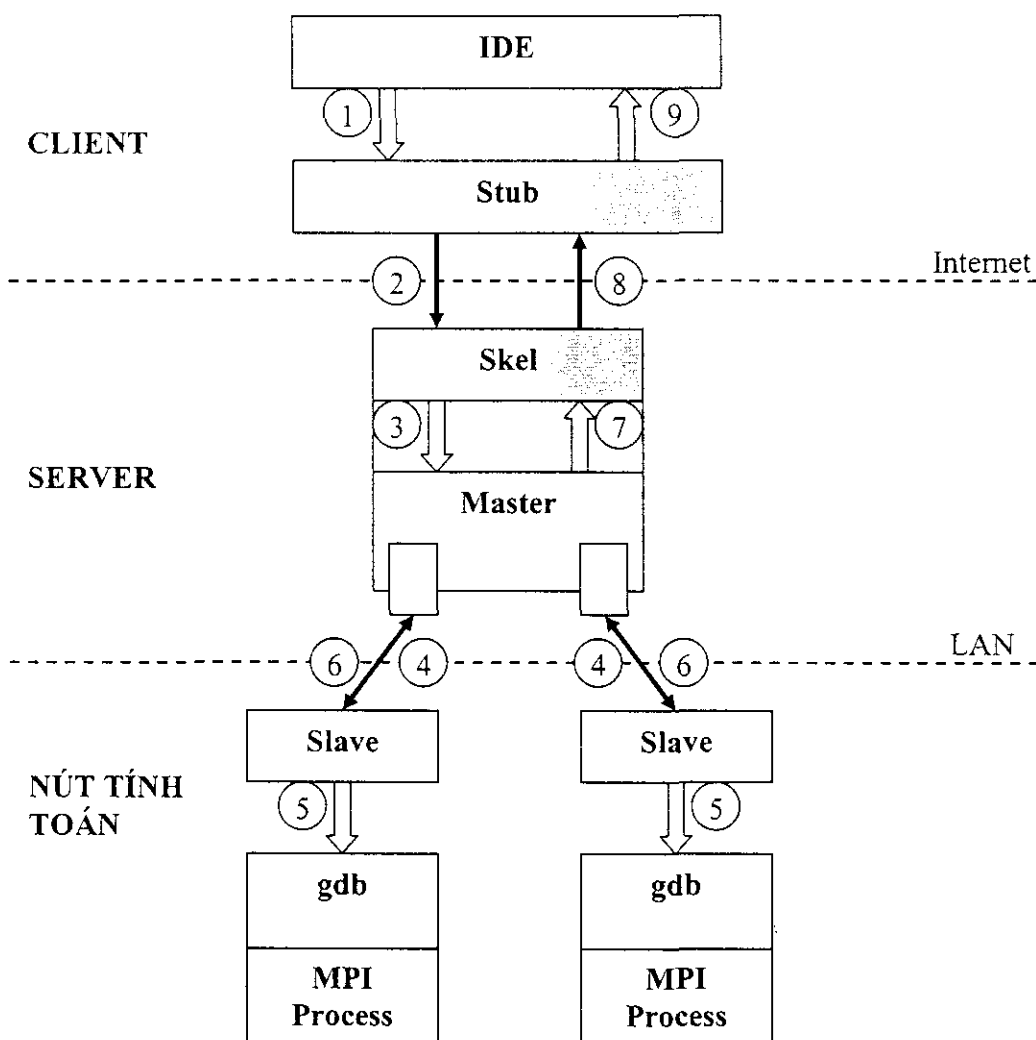
Các ngoại lệ có thể xảy ra :

- Người dùng yêu cầu chạy chương trình không tồn tại.
- Các Slave khởi tạo không thành công (có vài Slave không chạy).



Hình 3-7 Kịch bản hoạt động của BKPD - Pha thiết lập

3.2.3. Pha phục vụ



Hình 3-8 Kịch bản hoạt động của BKPD - Pha phục vụ

Trạng thái ban đầu:

Môi trường debug đã được khởi tạo thành công. Hệ chờ người sử dụng nhập vào các lệnh debug.

Các bước :

- 1 IDE gọi hàm thực hiện lệnh của người dùng do Client communicator cung cấp.
- 2 Client communicator truyền lệnh cho Skel.

- 3 Skel gọi hàm thực hiện lệnh do Master cung cấp.
- 4 Master xử lý lệnh, sau đó truyền xuống cho slave thông qua kết nối được thiết lập trong pha thiết lập.
- 5 Slave điều khiển gdb thực hiện lệnh.
- 6 Slave gửi kết quả về cho master.
- 7 Master tổng hợp kết quả trả về cho Skel.
- 8 Skel truyền kết quả về cho Client communicator.
- 9 Client communicator trả về kết quả cho IDE.

Các ngoại lệ có thể xảy ra

- Câu lệnh sai cú pháp.
- Có một Slave bị block.

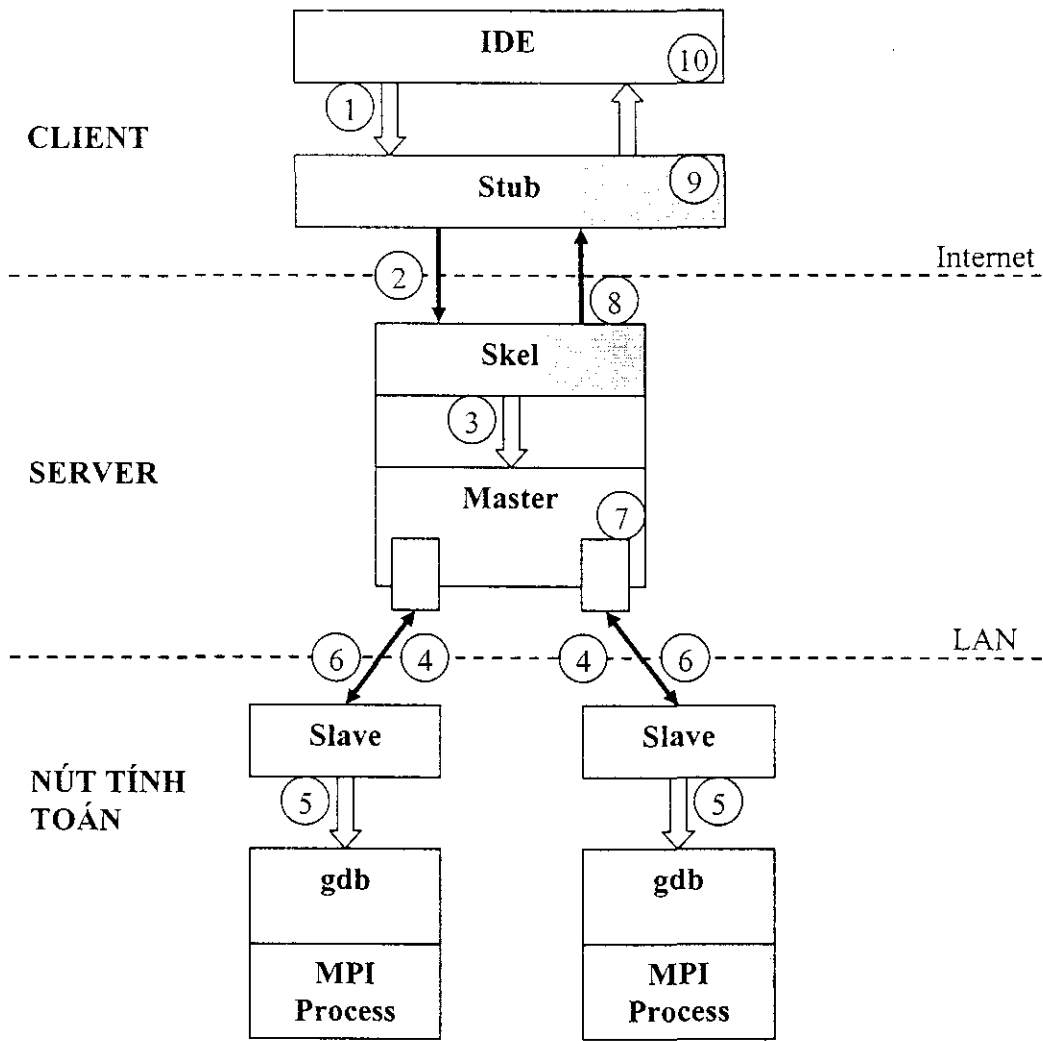
3.2.4. Pha kết thúc

Trạng thái ban đầu :

Hệ đang hoạt động bình thường.

Các bước :

- 1 IDE gọi hàm kết thúc của Client communicator.
- 2 Client communicator gửi thông báo kết thúc cho Skel.
- 3 Skel gọi hàm kết thúc của Master.
- 4 Master gửi thông báo kết thúc cho các Slave.
- 5 Các Slave điều khiển các gdb kết thúc phiên làm việc.
- 6 Slave gửi thông báo về cho Master và tự kết thúc.
- 7 Master tự kết thúc.
- 8 Skel gửi thông báo về cho Client communicator rồi tự kết thúc.
- 9 Client communicator tự kết thúc.
- 10 IDE kết thúc phiên phục vụ.



Hình 3-9 Kịch bản hoạt động của BKPD - Pha kết thúc

CHƯƠNG 4

HỆ THỐNG THỰC THI TÍNH TOÁN

Các ứng dụng sẽ được mô tả một cách độc lập. Trong phần này, đứng từ quan điểm xây dựng các bộ công cụ giúp người dùng khai thác hệ thống, chúng tôi trình bày ở đây công cụ để trình công việc.

Theo quan điểm nhìn nhận của người dùng, một công việc là một file kịch bản được chuyển tới hệ thống tính toán theo lệnh mà hệ thống cung cấp. Đây là một file dưới dạng một dạng ngôn ngữ đơn giản. Các công việc được duy trì trên Server cho tới khi nó được thực thi hoặc bị yêu cầu chấm dứt. Sau khi nhận file script từ phía người sử dụng, nếu như yêu cầu tạo công việc(Job) tới các đối tượng hàng đợi (Queue) tương ứng theo yêu cầu của người sử dụng là thành công thì hệ thống sẽ tạo ra một công việc(Job) tương ứng.

Hệ thống quản lý công việc của BKluster được phát triển dựa trên PBS. Do đó, bộ công cụ để trình công việc đóng vai trò hỗ trợ cho người trong việc giao tiếp với hệ thống quản lý công việc mà cụ thể hơn là với PBS. Trên thực tế bằng các lệnh làm việc trực tiếp của PBS, người dùng cũng có thể để trình một công việc. Tuy nhiên, nó đòi hỏi phải thực hiện nhiều thao tác cũng như phải xác lập nhiều tham số hệ thống. Bộ công cụ được đề xuất nhằm giảm lược tối đa số giao tác thủ công và thiết lập sẵn tham số theo đặc thù hệ thống.

Đứng về mặt chức năng, bộ công cụ có đầy đủ các chức năng tối thiểu như các

lệnh của PBS bao gồm: *qsub*, *qalter*, *qhold*, *qdel*, *qrls*, *qstat* ...

Lệnh đầu tiên chúng ta cần quan tâm chính là lệnh **qsub**, đây là lệnh cho phép người dùng đệ trình một công việc với hệ thống. Công việc được đệ trình phải được viết dưới dạng một file kịch bản từ trước. Tuy nhiên một khó khăn đối với người sử dụng lệnh này chính là việc có quá nhiều tham số cho lệnh và các tham số này đôi khi là rất xa lạ ngay cả với những người sử dụng chuyên nghiệp chứ đừng nói gì đến những người dùng không chuyên nghiệp.

Lệnh tiếp theo mà chúng ta quan tâm là lệnh **qalter**, đây là lệnh cho phép người dùng sử dụng để điều chỉnh các tham số về công việc mà mình đã đệ trình trước đây bằng lệnh *qsub*. Đây là một lệnh cũng rất quan trọng, lệnh này cho phép người dùng có một khả năng mềm dẻo hơn trong quá trình thực thi các công việc của mình.

Cũng như các thao tác khác chúng ta đôi khi cũng muốn xóa bỏ đi những công việc mà mình đã đệ trình trước đây (có thể vì một lý do nào đó) thì lệnh **qdel** sẽ đáp ứng nhu cầu này. Lệnh này cho phép người dùng xóa bỏ một công việc đã được đệ trình từ trước nhưng vẫn chưa được thực hiện.

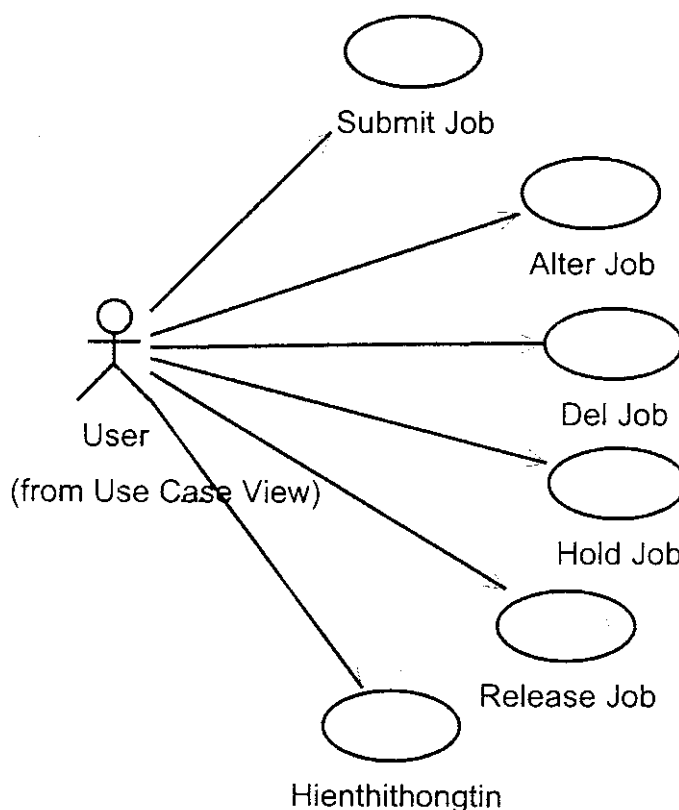
Lệnh tiếp theo mà chúng ta quan tâm chính là lệnh **qhold**, đây là lệnh cho phép người dùng đưa một công việc nào đó vào trạng thái nằm chờ vô thời hạn cho đến khi công việc này được giải phóng bằng lệnh **qrls**. Lệnh này cho phép chúng ta có thể bỏ chỉ chạy các job của mình một cách linh hoạt tùy thuộc vào các điều kiện ngoại cảnh tác động như có khi một công việc có thể được thực hiện ngay nhưng kết quả chúng ta lại chưa cần ngay thì chúng ta có thể sắp xếp cho công việc đó dừng lại để cho một công việc khác có kết quả quan trọng hơn (tại thời điểm này) được thực hiện trước.

Ngược với lệnh **qhold** là lệnh **qrls**, lệnh này dùng để giải phóng các công việc khỏi trạng thái **qhold** và đưa công việc trở lại trạng thái sẵn sàng thực hiện.

Ngoài ra đôi khi người dùng muốn xem thông tin về các công việc của mình mà

vẫn nằm trong hàng đợi chờ thực hiện, khi đó thì lệnh `qstat` sẽ đáp ứng nhu cầu này. Lệnh này cho phép chúng ta có khả năng xem được các thông tin của các công việc đang nằm trong hàng đợi của hệ thống..

Như vậy, ngoài chức năng chính là đề trình công việc, bộ công cụ cũng cho phép người dùng quản lý các công việc đã đề trình.

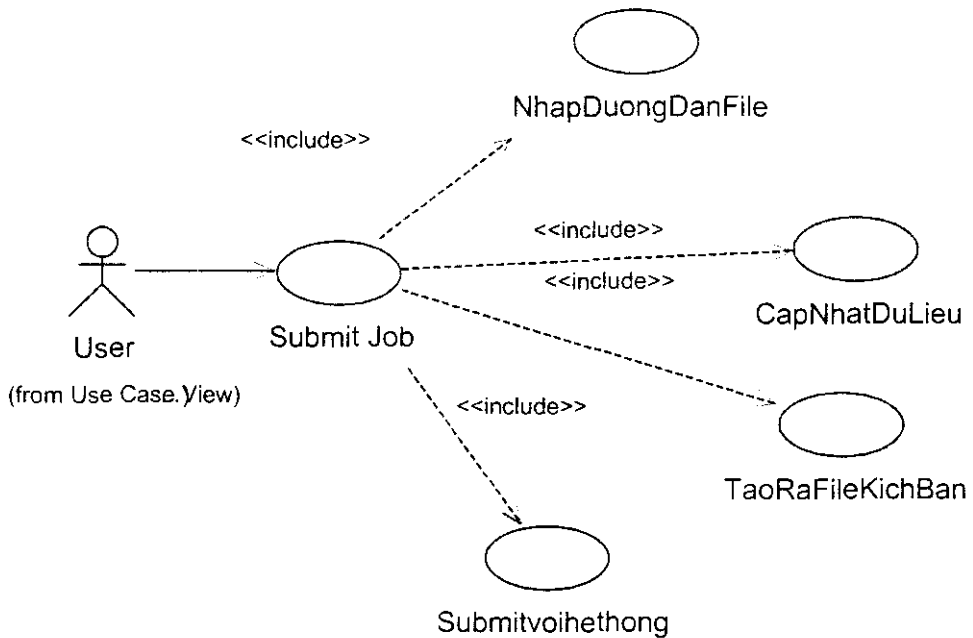


Hình 4-1 Use case thực thi tính toán

4.1. Chức năng Submit Job

Người sử dụng sử dụng chức năng này để đề trình một công việc với hệ thống..Sau khi người kích hoạt chức năng này hệ thống sẽ yêu cầu người dùng đưa đường dẫn đến file kịch bản và hệ thống cũng yêu cầu người dùng nhập vào tên của file mã nguồn đã được dịch rồi. Ngoài hai thông tin bắt buộc trên thì hệ thống cho phép

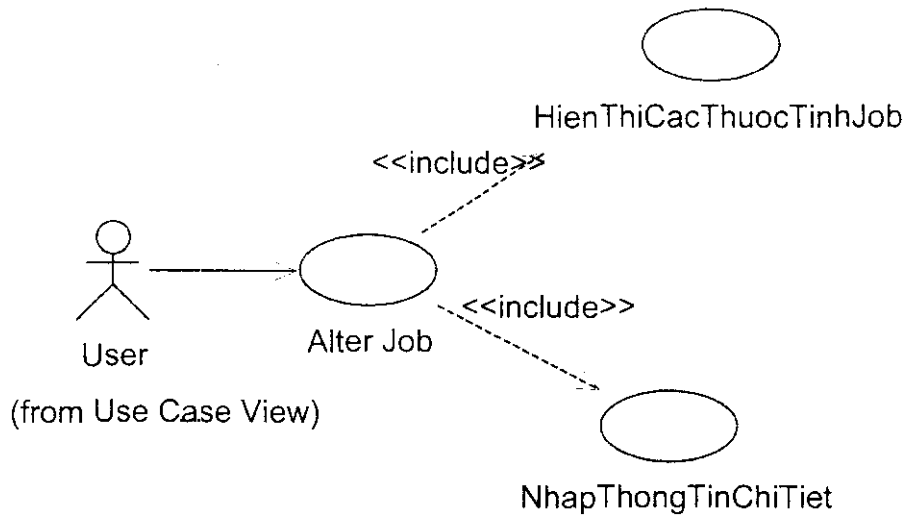
người dùng tự nhập các tham số liên quan đến công việc của mình hoặc lựa chọn các tham số mặc định sau đó hệ thống sẽ tự sinh ra file kịch bản và đệ trình công việc với hệ thống.



Hình 4-2 Use case chức năng Submit Job

4.2. Chức năng Alter Job

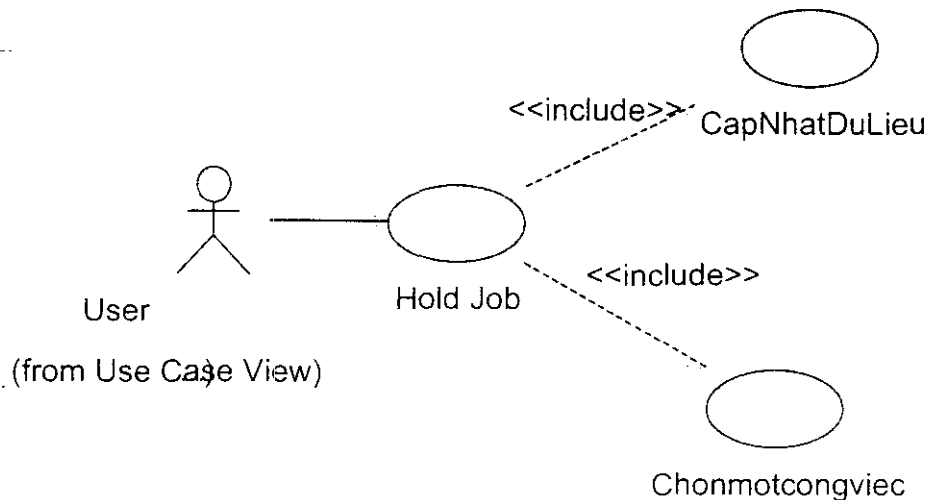
Người dùng dùng chức năng này để điều chỉnh các thuộc tính về một công việc thuộc quyền sở hữu của mình. Sau khi người dùng lựa chọn chức năng này hệ thống sẽ cho hiển thị các thông tin về thuộc tính của công việc mà người sử dụng đã lựa chọn để người dùng có thể dễ dàng hơn trong khi điều chỉnh tham số mới cho công việc này. Sau đó hệ thống sẽ yêu cầu nhập vào các giá trị mới của các thuộc tính của công việc mà chúng ta muốn thay đổi



Hình 4-3 Use case chức năng Alter Job

4.3. Chức năng Hold Job

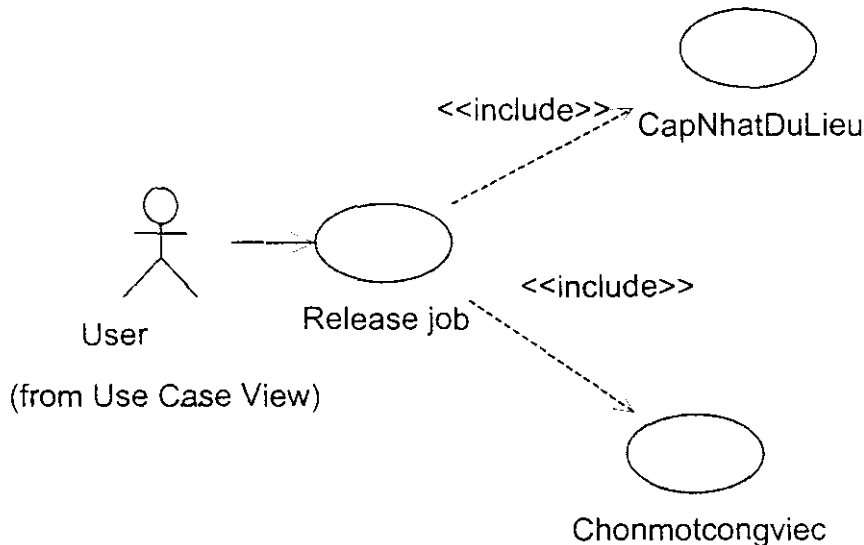
Người sử dụng dùng chức năng này để đưa một job vào trạng thái chờ đợi cho đến khi trạng thái này được giải phóng thì job mới sẵn sàng để đưa vào trạng thái chạy. Hệ thống sẽ kiểm tra xem job tương ứng có ở trong trạng thái running và có thuộc quyền sở hữu của người dùng đó hay không, nếu thỏa mãn hai điều kiện trên thì job tương ứng sẽ được đưa vào trạng thái chờ đợi. và hệ thống sẽ cập nhật lại vào cơ sở dữ liệu trạng thái mới của công việc.



Hình 4-4 Use case chức năng Hold Job

4.4. Chức năng Release Job

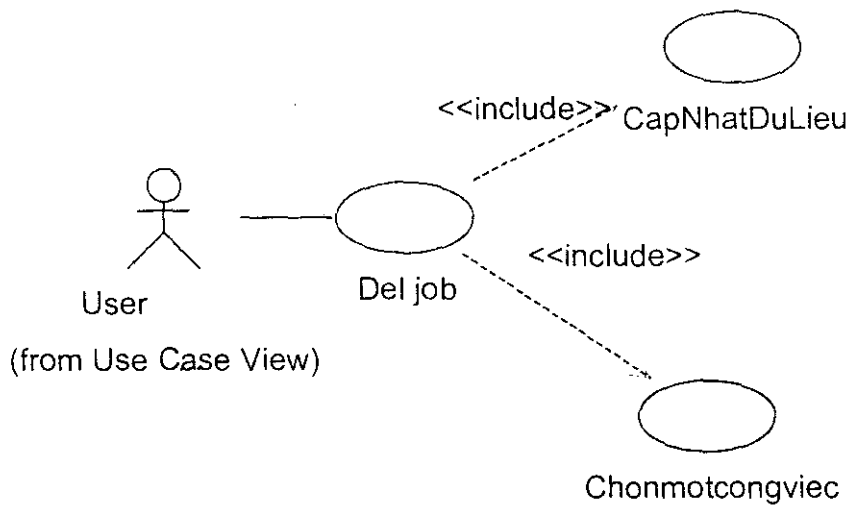
Chức năng này được sử dụng để giải phóng các công ở trạng thái hold sang trạng thái sẵn sàng để được thực thi. Người sử dụng chọn một công đang ở trong trạng thái hold và nhấn vào chức năng **Release job** hệ thống sẽ giải phóng công tương ứng khỏi trạng thái hiện thời đồng thời cập nhật lại cơ sở dữ liệu.



Hình 4-5 Use case chức năng Release Job

4.5. Chức năng Del Job

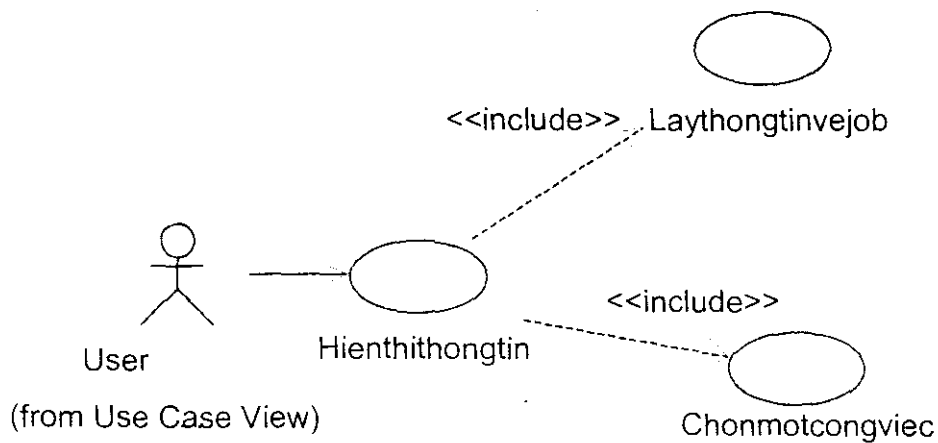
Chức năng này dùng để xóa một công việc ra khỏi hàng đợi. Hệ thống sẽ kiểm tra trạng thái của công việc nếu không có lỗi gì thì sẽ tiến hành xóa bỏ công việc đó. Sau khi thực hiện hệ thống thông báo lại cho người dùng là đã xóa công việc tương ứng khỏi danh sách công việc trong hệ thống đồng thời cập nhật lại trong cơ sở dữ liệu. Trong trường hợp, Job được lựa chọn không thuộc quyền sở hữu của người dùng này, khi đó hệ thống sẽ thông báo lại và yêu cầu người dùng nhập lại thông tin về job hoặc hủy bỏ yêu cầu của người dùng.



Hình 4-6 Use case chức năng Del Job

4.6. Chức năng Show Info Detail

Chức năng này cho phép người dùng xem chi tiết các thông tin về công việc mà mình đã lựa chọn, chức năng này sẽ giúp cho người dùng quản lý tốt hơn các công việc của mình, có khả năng nắm được trạng thái và các tài nguyên mà công việc yêu cầu. Chức năng này được kích hoạt khi người sử dụng chọn một công việc trong danh sách các công việc của mình và lựa chọn chức năng hiển thị thông tin về công việc. Hệ thống sẽ lấy các thông tin liên qua đến công việc mà người sử dụng yêu cầu và hiển thị thông tin cho người sử dụng.



Hình 4-7 Use case chức năng Show Infor Detail

CHƯƠNG 5

HỆ THỐNG GIÁM SÁT VÀ QUẢN TRỊ HỆ THỐNG

Hệ thống quản trị và giám sát hệ thống hỗ trợ người quản trị trong việc nắm bắt đầy đủ và xử lý các thông tin bên trong hệ thống. Hệ thống này bao gồm 5 module lớn:

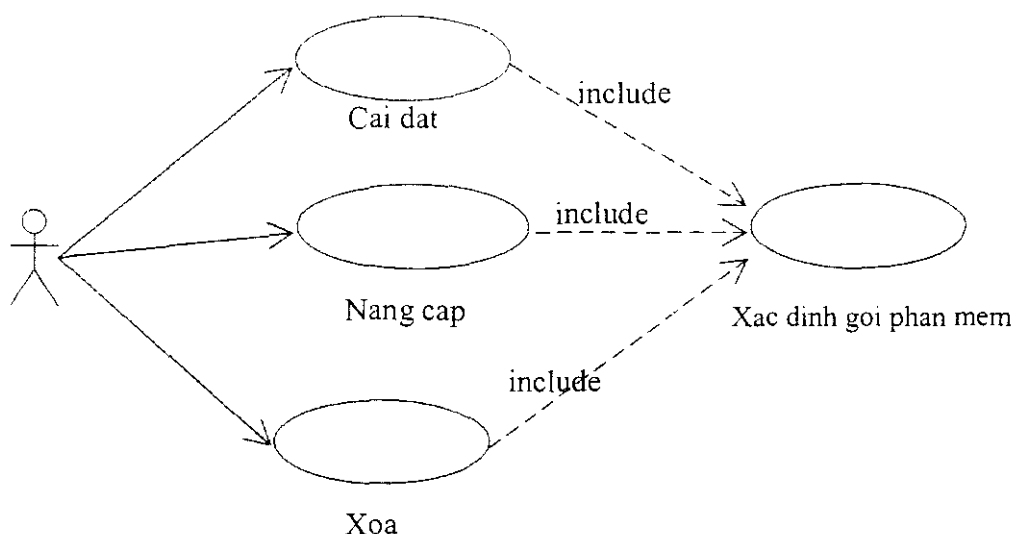
- Module quản lý gói phần mềm
- Module quản lý cấu hình hệ thống
- Module quản lý người dùng
- Module giám sát hệ thống
- Module đánh giá hiệu năng hệ thống

Do việc quản trị liên quan trực tiếp tới các thành phần bên trong, người quản trị không những phải có kiến thức cơ bản về tính toán phân cụm mà cũng cần hiểu rõ các thành phần trong. Bộ công cụ giúp người dùng lấy thông tin dễ dàng hơn và hiện thị thông tin một cách trực quan giúp người quản trị có được mối liên hệ giữa các thông tin thu thập được.

5.1. Module Quản lý gói phần mềm

Module Quản lý gói phần mềm cho phép cài đặt, nâng cấp, xóa và đồng thời cung cấp thông tin thống kê các gói phần mềm đã được cài đặt trong toàn bộ hệ thống.

Module được triển khai theo mô hình Client/Server, quá trình truyền thông được thực hiện thông qua socket, cổng 4242.



Hình 5-1 Use case Quản lý gói phần mềm

5.2. Quản lý người dùng

Mỗi người dùng khai thác các dịch vụ tính toán phải đăng ký một tài khoản. Người quản trị hệ thống sẽ đảm nhận việc chấp nhận đăng ký, phân quyền và quản lý tài khoản của các người dùng khác. Mỗi tài khoản người dùng sẽ có một thư mục trên server để lưu các file chạy, script, log file và kết quả việc chạy chương trình.

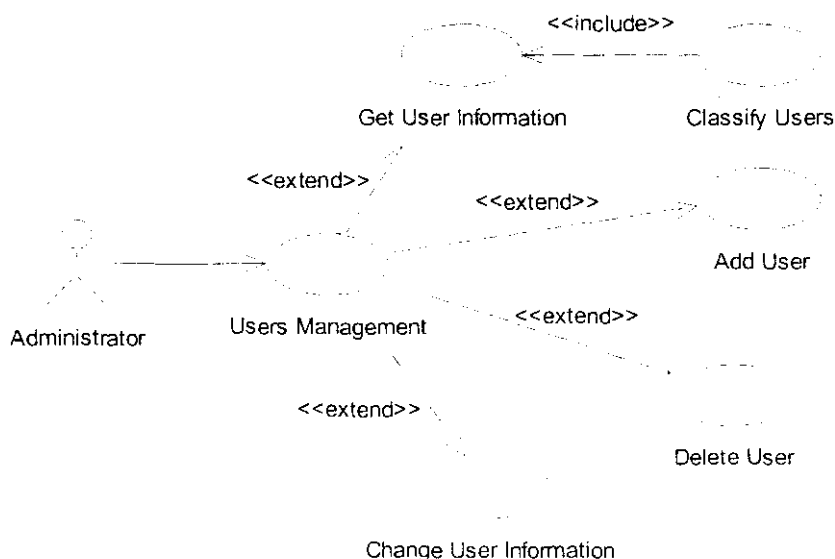
Hệ thống có một số lớp người dùng được liệt kê trong bảng sau:

Người quản trị	<ul style="list-style-type: none"> • Toàn quyền với hệ thống và các người dùng khác • Cấu hình, thêm bớt, xóa, sửa tất cả các đối tượng thuộc hệ thống phân cụm bao gồm: hàng đợi công việc, server tính toán, nút tính toán, các công việc.
Thành viên	<ul style="list-style-type: none"> • Cần đăng ký trước khi trở thành thành viên của hệ thống • Có quyền xem cấu hình hệ thống

	<ul style="list-style-type: none"> • Đề trình, thực thi công việc • Toàn quyền với thư mục được cung cấp trên server quản trị
Khách	<ul style="list-style-type: none"> • Chỉ có quyền xem cấu hình hệ thống, không có quyền nào khác.

Các chức năng chính của module quản lý người dùng bao gồm :

- Thêm người dùng vào hệ thống: thêm người dùng vào hệ thống bao gồm việc thao tác với thư mục của người dùng, cập nhật cơ sở dữ liệu...
- Lấy thông tin về một người dùng.
- Cập nhật thông tin của người dùng: chẳng hạn như thêm quyền cho người dùng đó. Trong BKCluster người dùng được chia làm ba cấp: người quản trị là người có toàn quyền đối với hệ thống, thành viên của hệ thống là người có thể thực hiện các công việc song song, người dùng thông thường là người chỉ có một số quyền hữu hạn như xem thông tin về hệ thống.
- Xóa người dùng



Hình 5-2 Use case quản lý người dùng

5.3. Quản lý cấu hình hệ thống

Chức năng cấu hình cung cấp phương tiện cho người quản trị thao tác với các nút tính toán, server tính toán, và các hàng đợi công việc, cũng như thiết lập các thuộc tính khác như cấp quyền quyền, cho phép/cấm các nút khác tham gia vào hệ thống...

Cấu hình của hệ thống BKluster được xem xét trên các khía cạnh

- Số nút và cấu hình các nút
- Các tham số hệ thống, chủ yếu là các tham số của PBS Server liên quan tới việc cân bằng tải, lập lịch và chế độ làm việc

Module quản lý cấu hình bao được cả 2 lớp thông tin này, nó cung cấp các chức năng cho phép người quản lý thao tác trên các thông tin này một cách hiệu quả. các chức năng đó được chia vào 3 nhóm lớn

- Nhóm chức năng quản lý nút
- Nhóm chức năng quản lý server
- Nhóm chức năng quản lý các chế độ làm việc mà cụ thể là quản lý hàng đợi

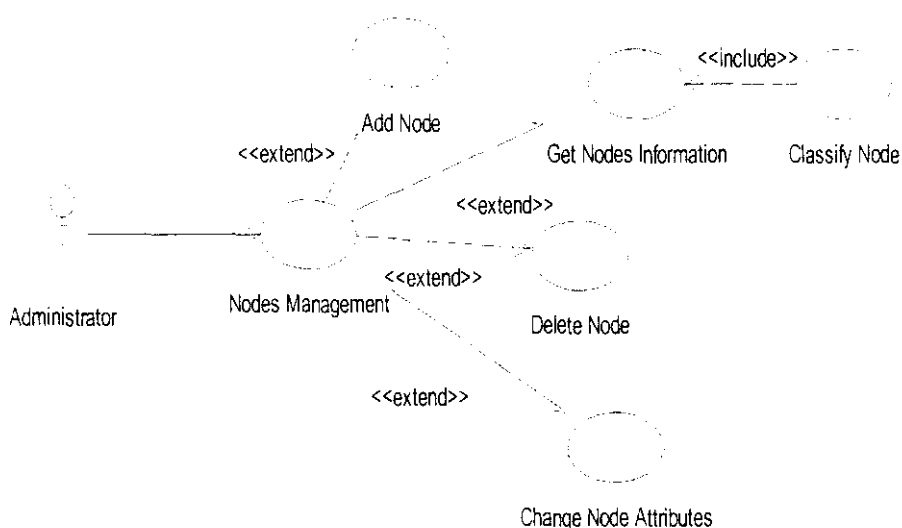
5.3.1. Quản lý các nút tính toán

Nút tính toán trong một hệ cluster đóng vai trò của một phần tử xử lý. Thông thường nó là một máy tính với một hệ thống xử lý riêng, một bộ nhớ độc lập, một hoặc nhiều bộ vi xử lý và một địa chỉ mạng xác định. Người ta thường dùng thuật ngữ *Excution Host* để ám chỉ đến các nút tính toán thay cho thuật ngữ chung trong tính toán song song là process element.

Trong một hệ cluster, đặc biệt trong một hệ cluster sử dụng PBS với chức năng quản lý công việc, các thuộc tính cần quan tâm của một nút tính toán bao gồm: Node_state, Node_type, Name, Name, Propertice, Jobs, Ncpus, Physmem, Loadave.

Các chức năng chính do thành phần Quản lý nút tính toán cung cấp bao gồm:

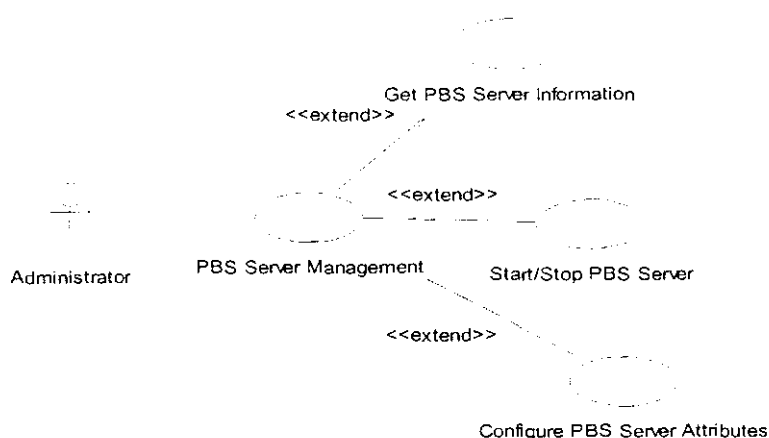
- Xem trạng thái của các nút tính toán(pbs_mom), các trạng thái có thể là tắt, không tham gia tính toán, hoặc đang phục vụ tính toán... .
- Thêm một nút tính toán vào hệ thống
- Bớt một nút ra khỏi hệ thống tính toán
- Thay đổi cấu hình của nút tính toán tham gia vào hệ thống



Hình 5-3 Use case quản lý các nút tính toán của hệ thống

5.3.2. Quản lý cluster server

- Lấy thông tin về trạng thái của server: chúng ta có thể lấy được thông tin về bash server như có bao nhiêu hàng đợi, cho phép cấm người dùng, nút tính toán tham gia vào hệ thống hay không...
- Tắt PBS server: trong một số trường hợp cần thiết cần phải tắt PBS server cho mục đích bảo trì hay an ninh của hệ thống...
- Khởi động lại server tính toán.



Hình 5-4 Use case quản lý cluster server

5.3.3. Quản lý chế độ làm việc

Hàng đợi là một đối tượng được quản lý bởi Server. Một hàng đợi có thể bao gồm nhiều công việc(Job), tập các thuộc tính và một tập hợp các đối tượng dữ liệu bên trong. Các công việc được định vị trong các hàng đợi và trở thành các phần tử của hàng đợi. Các máy khách có thể lấy về các thông tin của hàng đợi và các công việc bên trong hàng đợi thông qua các yêu cầu tới Server.

Có hai dạng hàng đợi cơ bản: *hàng đợi định tuyến* và *hàng đợi thực thi*. Và các thuộc tính bên trong của hàng đợi quyết định kiểu loại của mỗi hàng đợi.

Khi một công việc nằm bên trong hàng đợi định tuyến, nó có xu hướng được chuyển tới một vị trí khác và khái niệm vị trí ở đây có thể bao gồm các hàng đợi khác cùng nằm trên Server hoặc các hàng đợi của các Server khác.

Một Công việc được coi là đã được lấy ra khỏi hàng đợi định tuyến khi:

- Công việc đã thực sự được định vị thành công tới một hàng đợi khác.
- Công việc đã bị xoá khỏi hàng đợi.

- Công việc đã bị người sử dụng chuyển tới một hàng đợi khác.
- Công việc đã bị bỏ qua bởi Server

Khi một công việc nằm trong hàng đợi thực thi, nó đang nằm trong tình trạng chuẩn bị được thực hiện. Một công việc sẽ vẫn là phần tử của hàng đợi thực thi kể cả khi nó đã được lựa chọn để đưa vào thực hiện. Và một công việc được coi là đã được lấy ra khỏi hàng đợi thực hiện khi:

- Công việc đã được thực hiện và đã kết thúc.
- Công việc đã được chuyển tới một hàng đợi khác.
- Công việc đã bị người sử dụng bỏ qua không thực hiện.

Có chế và chiến lược hoạt động của một hàng đợi phụ thuộc vào giá trị của một tập các thuộc tính, ví dụ như `Acl_Group_Enable`, `Acl_groups`, `Acl_host_enable...` Các thuộc tính này có thể được xác lập bởi người quản trị thông qua công cụ quản trị. Các chức năng chính của thành phần này bao gồm:

- Lấy thông tin về các hàng đợi công việc.
- Tạo hàng đợi: tạo các hàng đợi trên server với các thuộc tính do người quản trị cung cấp khi tạo hàng đợi
- Thay đổi các thuộc tính cho các hàng đợi: có thể thêm các thuộc tính, chỉnh sửa tài nguyên... cho hàng đợi.
- Thay đổi trạng thái của hàng đợi (chạy, dừng...)
- Xóa hàng đợi.

5.4. Module giám sát hoạt động

Module giám sát hệ thống cho phép người quản trị nắm bắt chủ yếu các thông tin động, có tần suất thay đổi cao. Thông thường các thông tin chỉ có giá trị tại một thời điểm nhất định. Ngoài việc, cung cấp các thông tin tức thời của hệ thống tới người dùng, module cũng được tích hợp một số công cụ mang tính thông kê cho phép tổng hợp các thông tin về hiện trạng hoạt động hệ thống.

Đối với một hệ thống tính toán, các thông tin đáng lưu ý là các thông tin có liên quan trực tiếp tới khả năng tính toán hiện tại của hệ thống. Trong trường hợp của một hệ thống tính toán phân cụm mà cụ thể là BKluster, chúng tôi lưu tâm tới 2 thông tin sau:

- Thông tin về sử dụng tài nguyên tính toán trên từng nút tính toán tại mỗi thời điểm
- Thông tin về khả năng trao đổi thông tin giữa các nút tính toán tại mỗi thời điểm

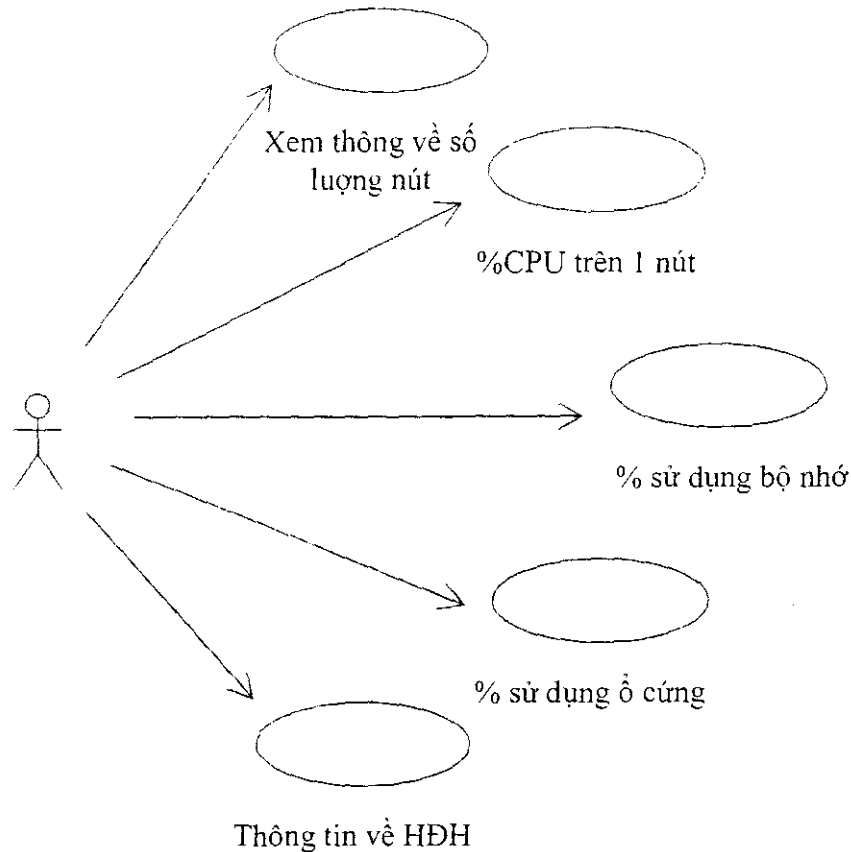
Đứng từ quan điểm đó, module giám sát hệ thống sẽ có chức năng sau:

- Giám sát việc sử dụng tài nguyên trên các nút tính toán
- Giám sát thông lượng mạng kết nối các nút tính toán
- Thống kê hoạt động hệ thống

5.4.1. Chức năng giám sát tài nguyên hệ thống

Thông tin liên quan tới việc sử dụng tài nguyên hệ thống mang tính động trong thời gian hệ thống hoạt động ví dụ, dung lượng bộ nhớ đang sử dụng, phần trăm sử dụng CPU. Thông tin này cần được phân biệt với thông tin về tài nguyên hệ thống. Thông tin tài nguyên hệ thống ví dụ như tốc độ xử lý của CPU, dung lượng bộ nhớ lại được xem là các thông tin tĩnh.

Hệ thống giám sát tài nguyên quan tâm chủ yếu tới các thông tin động. Chức năng này cho phép cập nhật các thông tin này một cách liên tục và hiện thị một cách trực quan bằng các đồ thị. Thông qua các đồ thị, người quản trị có thêm nhiều thông tin khác, đặc biệt là mối liên hệ theo thời gian.



Hình 5-5 Use case giám sát tài nguyên hệ thống

5.4.2. Giám sát thông lượng mạng kết nối

Chức năng này cho phép người quản trị theo dõi hoạt động trao đổi thông tin giữa các nút tại mỗi thời điểm và hiện thị thông tin này theo thời gian. Khả năng trao đổi thông tin giữa hai nút được xác định dựa trên số gói tin ở tầng mạng (packet) vào ra trên 2 interface tương ứng

5.4.3. Chức năng ghi nhật ký hoạt động

Các thông tin tập trung vào ba loại : *thứ nhất*, thông tin về các hoạt động của người dùng (tổng số công việc, thời gian thực hiện, thời gian yêu cầu...); *thứ hai*, thông tin ghi lại các hoạt động của server; *thứ ba*, thông tin về hoạt động của các hàng đợi (độ hiệu quả, số lượng công việc).

- Lưu thông tin về hoạt động của người dùng, xử lý với mục đích thống kê

- Lưu thông tin về hoạt động của server
- Lưu thông tin về hoạt động của các hàng đợi.

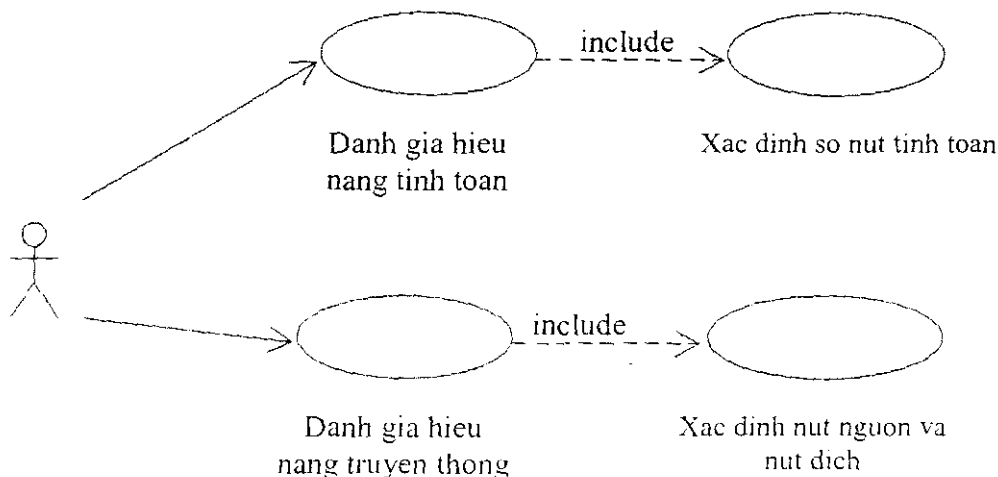
5.5. Module Đánh giá hiệu năng hệ thống

Đánh giá hiệu năng hệ thống là một công cụ hữu ích giúp người quản trị cluster có được những điều chỉnh cần thiết nhằm đạt hiệu năng tối ưu. Đối với một hệ thống tính toán nói chung, hiệu năng hệ thống được lượng hoá theo đơn vị số phép tính trên giây. Ngoài tiêu chí chung, để thấy rõ đặc tính của hệ thống phân cụm, người ta có thể đối sánh kết quả chung với các hiệu năng của từng thành phần tính toán và thông lượng trao đổi thông tin giữa các thành phần.

Module đánh giá hiệu năng của hệ thống BKluster cho phép người quản trị thực hiện 2 phép đo bao gồm:

- Xác định năng lực tính toán của một số nút bất kỳ (≥ 1) tính theo đơn vị GFLOP dựa trên một tập các bài toán chuẩn:
- Xác định năng lực truyền thông của môi trường truyền thông giữa các nút

Module đánh giá hiệu năng hệ thống được xây dựng dựa trên việc tích hợp các gói phần mềm đánh giá hiệu năng mã nguồn mở thông dụng, hoạt động với giao diện thân thiện, dễ sử dụng.



Hình 5-6 Use case Đánh giá hiệu năng hệ thống

Tài liệu tham khảo

- [1]. **Thomas Sterling**, “*An Introduction to PC Clusters for High Performance Computing*”, *California Institute of Technology and NASA Jet Propulsion Laboratory, USA*
- [2]. **Albeaus Bayucan Robert L. Henderson Casimir LesiakBhroam Mann Tom ProettDave Tweten**, “Portable Batch System External Reference Specification”. *MRJ Technology Solutions*
- [3]. **Albeaus Bayucan Robert L. Henderson Casimir LesiakBhroam Mann Tom ProettDave Tweten**, “Portable Batch System Internal Design Specification”. *MRJ Technology Solutions*
- [4]. **Albeaus Bayucan Robert L. Henderson Casimir LesiakBhroam Mann Tom ProettDave Tweten**, “Portable Batch System Administrator Guide”. *MRJ Technology Solutions*
- [5]. **Albeaus Bayucan Robert L. Henderson Casimir LesiakBhroam Mann Tom ProettDave Tweten**, “Portable Batch System user guide”. *MRJ Technology Solutions*
- [6]. **PACS Training Group**, “Introduction to MPI”, *University of Illinois*.
- [7]. **LAM/MPI Forum**, “www.lam-mpi.org”.
- [8]. **The LAM/MPI Team**, “LAM/MPI Documents”, *Open Source Lab*
- [9]. **Nicolas J. Nevin**, “The XMPI API and trace file format”, *January 29, 1997*.
- [10]. “UC Berkeley Grid” <http://www.millennium.berkeley.edu/ganglia/>