



Condor

High Throughput Computing

**Polowanie
na niewykorzystaną
moc obliczeniową**

Marcin Rociak

Co to jest?



- z Specjalizowany system wsadowy:
 - y kolejkowanie zadań,
 - y priorytetowanie zadań,
 - y klasyfikacja zasobów.
- z Efektywnie wykorzystuje moc obliczeniową stacji roboczych.

Typowy scenariusz



- z Użytkownik przedkłada zadanie.
- z Condor szuka odpowiedniej maszyny i uruchamia na niej zadanie.
- z W przypadku awarii zadanie jest przenoszone na inną maszynę i wykonywane od punktu, w którym zostało przerwane.

Co robi Condor



- z Condor łączy zbiory stacji roboczych i dedykowane klastry w jedno 'urządzenie' (*Condor pool*).
- z Dopasowuje dostępne zasoby do zgłaszanych zadań za pomocą *ClassAd* tak, aby wszystkich uszczęśliwić (zarówno właścicieli maszyn jak i uruchamiających zadania).

Co robi Condor (c.d.)



- z Condor robi wszystko aby wykonać zadanie nawet jeśli maszyny
 - y 'padną' (lub zostaną odłączone od sieci),
 - y skończy im się wolna przestrzeń dyskowa,
 - y są często używane przez innych,
 - y są daleko i zarządzane przez kogoś innego.

Dopasowywanie - *ClassAd*

- z *Classified Advertisement*
- z Użytkownicy (zadania) precyzują swoje wymagania:
 - x „Potrzebuję maszyny INTEL/LINUX z 256MB RAM”
- z Właściciele (maszyny) precyzują ograniczenia:
 - x „Uruchamiaj zadania tylko jeśli akurat nie pracuję i nigdy nie uruchamiaj zadań należących do xxxx”.

Usługi




- z Każdy uruchamiany za pomocą Condora program automatycznie korzysta z:
 - y wyszukiwania / dopasowywania zasobów,
 - y wsadowego zarządzania kolejką,
 - y szeregowania zadań.

Usługi (c.d.)



- z Jeśli istnieje możliwość przelinkowania programu z bibliotekami Condora, to dodatkowo dostępne są:
 - y kontrola i restartowanie zadań,
 - y migracja zadań,
 - y zdalne wywołania systemowe.

Kontrole zadań (*checkpointing*)



- z Zapis całego stanu procesu w pliku z obrazem.
 - y Pamięć, CPU, I/O itp.
- z Proces może być uruchomiony dokładnie od tego miejsca (np. na innej maszynie w przypadku awarii).

Zdalne wywołania systemowe



- z Wywołania systemowych funkcji we/wy są wyłapywane i transmitowane do maszyny, z której wysłano zadanie.
- z Umożliwia przeźroczystą migrację zadań.
- z Brak konieczności zmian w kodzie źródłowym.
- z Niezależne od języka programowania.

Ograniczenia



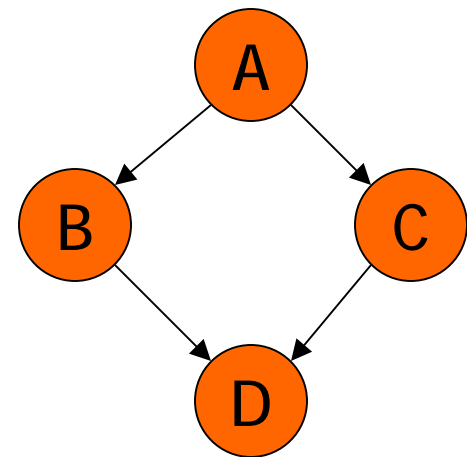
- z Brak konieczności zmian w źródłach, ale jeśli chcemy korzystać z usług kontroli i migracji zadań, to:
 - y niedopuszczalne zadania wieloprocesowe (`fork()`, `exec()`, `system()`),
 - y niedopuszczalna komunikacja za pomocą potoków, semaforów, pamięci dzielonej.
 - y ...

Zadania zależne od siebie

z *DAGMan - Directed Acyclic Graph Manager*

y Umożliwia wyspecyfikowanie zależności pomiędzy zadaniami, tak aby możliwe było automatyczne ich zarządzanie.

x „Nie uruchamiaj zadania B dopóki zadanie A nie zakończyło się powodzeniem, itd.”



Daemons (1/5)



Z condor_master

- y Pracuje na wszystkich maszynach.
- y Uruchamia wszystkie inne daemony, w razie problemów wysyła e-mail do administratora.
- y Zachowuje się jak serwer dla poleceń administrujących Condorem.
 - x (*condor_reconfig, condor_restart, condor_off, condor_on, ...*)

Daemony (2/5)



Z condor_startd

- y Reprezentuje maszynę w systemie.
- y Odpowiedzialny za uruchamianie, wstrzymywanie i zatrzymywanie zadań.
- y Narzuca życzenia właściciela maszyny.

Daemony (3/5)



Z condor_schedd

- y Reprezentuje użytkowników w systemie.
- y Zarządza kolejką zadań.
- y Odpowiedzialny za kontaktowanie się z dostępnymi maszynami i wysyłanie im zadań.
- y Obsługuje polecenia operujące na kolejce zadań
 - x (*condor_submit, condor_rm, condor_q, ...*)

Daemony (4/5)



Z condor_collector

- y Zbiera informacje od wszystkich daemonów w systemie, tworząc bazę danych systemu.
- y Każdy daemon wysyła do niego okresowo uaktualnienie *ClassAd*.
- y Odpowiada na zapytania
 - x innych daemonów,
 - x użytkownika (*condor_status*).

Daemons (5/5)



Z condor_negotiator

- y Dokonuje dopasowywania zasobów do żądań.
- y Pobiera od collectora informacje nt. dostępnych maszyn i beczynnych zadań.
- y Próbuje dopasować zadania do maszyn, które będą je obsługiwać (muszą być wzajemnie spełnione wymagania zarówno zadania jak i maszyny).

Rodzaje maszyn



z Central Manager

- y master
- y startd
- y schedd
- y negotiator
- y collector

z Regular Node

- y master
- y startd
- y schedd

z Execute-Only

- y master
- y startd

z Submit-Only

- y master
- y schedd

Bibliografia



- z Condor Team, University of Wisconsin-Madison;
Condor Version 6.3.1 Manual.
<http://www.cs.wisc.edu/condor/manual/v6.3/index.htm>

- z Peter Couvares and Todd Tannenbaum;
Condor Tutorial, First EuroGlobus Workshop, June 2001.
<http://www.cs.wisc.edu/condor/slides/euroglobus-tutorial/euroglobus-tutorial.html>

- z Derek Wright;
Condor Tutorial for Administrators, INFN-Bologna, 6/28/99.
<http://www.cs.wisc.edu/condor/tutorials/inf-admin/index.htm>