Studio e implementazione di metodi di previsione dei guasti per politiche di scheduling in ambito Desktop Grid

Relatore
Dott. Massimo Canonico

Candidato
Guido Vicino

Laurea in Informatica dei Sistemi Avanzati e Servizi di Rete
Università del Piemonte Orientale
"Amedeo Avogadro"

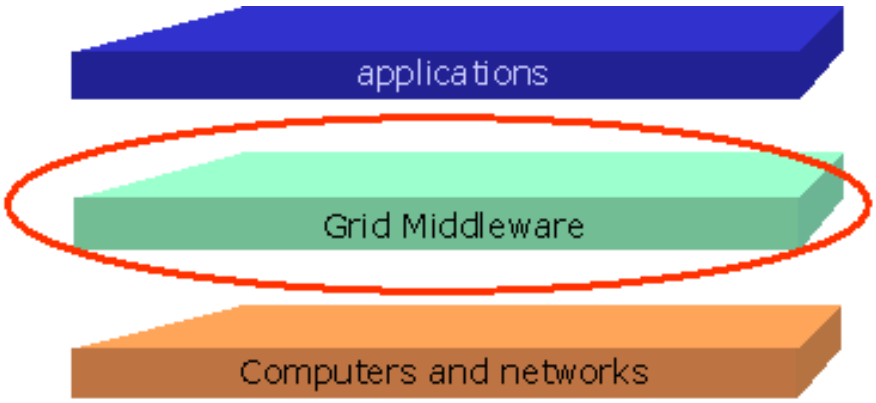
Anno Accademico 2006/2007

Introduzione
Grid Computing

- "Il *Grid Computing* (GC) è l'infrastruttura hardware e software che fornisce un accesso affidabile, consistente e pervasivo a grandi capacità computazionali" (Ian Foster e Carl Kesselman)
- "Il *Grid Computing* concerne la condivisione coordinata di risorse e la risoluzione di problemi in organizzazioni virtuali dinamiche e multi-istituzionali." (Ian Foster e Carl Kesselman)
- Grande eterogeneità di tipologie di *risorse condivise*.
- La *Virtual Organization* (VO) come un insieme di individui e/o istituzioni che sono definite da regole comuni di condivisione delle risorse. Regole dinamiche nel tempo ed interne a ciascuna VO.

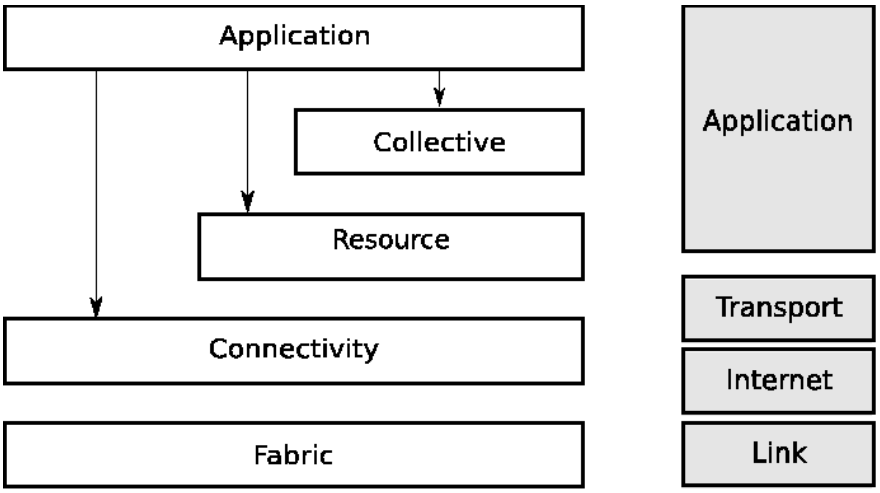
Introduzione
Middleware

- Il *middleware* è uno strato software che si preoccupa di nascondere e gestire autonomamente i dettagli relativi alla molteplicità delle risorse, all'autenticazione, all'accesso ai calcolatori o relativi all'allocazione dei lavori, delle applicazioni ed al trasferimento dei dati.



Introduzione
Storia del Grid Computing

- Il GC nasce nei primi anni '80 e '90 come estensione del "parallel computing" nel tentativo di collegare i vari centri di supercalcolo sparsi negli Stati Uniti.
- Nel 1995 nascono i due progetti più importanti di quelli che allora venivano chiamati *metacomputer*:
  - *Factoring via Network-Enable Recursion* (FAFNER)
  - *Information Wide Area Year* (IWAY)
- Nell'autunno del 1997 all'Argonne National Laboratory nasce il progetto *The Grid*, gestito da Ian Foster e Carl Kesselman.
- Contemporaneamente nascono *Globus* e *Legion* i primi sistemi software per Grid.
- In Europa nascono progetti analoghi quali *GridWay*, *GridLite* ed *Unicore*.
- Nascita dei primi gruppi di coordinamento fino all'attuale *Open Grid Forum* (OGF).
- *Service Oriented Architectures* (SOA) ed intenti futuri.



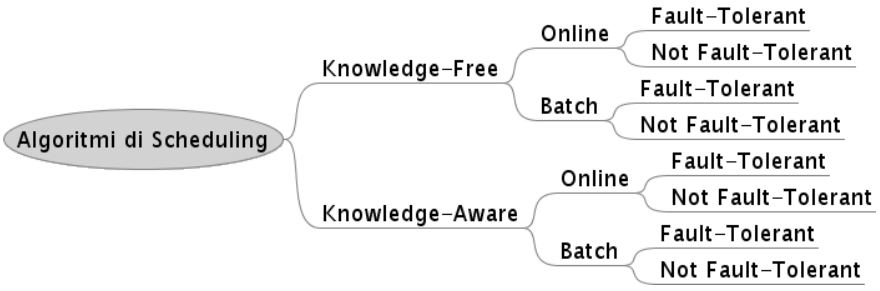
Anatomia di una Grid (I. Foster, C. Kesselman)

Esistono diverse applicazioni per il Grid Computing, ed è possibile distinguere tra due insiemi di queste:

- Distinzione tra
  - *Applicazioni Tradizionali*
  - *Applicazioni Grid-Enabled*.
- Ambiti applicativi:
  - Supercalcolo distribuito
  - Calcolo ad Alto Rendimento
  - Calcolo On-Demand
  - Calcolo Data-Intensive
  - Collaborative Computing

Introduzione
Scheduling

- Lo *Scheduling* mira a trovare una giusta allocazione dei lavori o *job* da eseguire dato un numero variabile di calcolatori disponibili all'interno di una VO.
- Problema dell'eterogeneità delle risorse.
- Diverse tipologie di politiche di scheduling.



Introduzione
Software per il Grid Computing

Middleware classici

- Globus Toolkit Version 4
- Legion

Desktop Grid

- BOINC
- OurGrid
- Entropia
- XtremWeb

Predizione dell'affidabilità

Introduzione

- L'*affidabilità* è il rapporto tra i tempi di *uptime* e quelli di *downtime*.
- Un *predittore* dato un *insieme di misurazioni pregresse* sui tempi di uptime, determina una *previsione* sul prossimo tempo di guasto della macchina.
- Sono stati studiati i vari metodi di previsione in letteratura e si è deciso di implementare i seguenti quattro:
  - Predizione Lineare
  - Predizione tramite distribuzioni Weibull ed Iperesponenziali
  - Network Weather Service
  - Predittori Ibridi
- Ognuno prende in input *n* misurazioni e restituisce 1 predizione.

Predizione dell'affidabilità

Predizione Lineare

- La *predizione lineare* è un operazione matematica dove i valori futuri di un segnale discreto sono stimati come funzione lineare dei campioni precedenti.
- La più comune rappresentazione è data da:

$$\tilde{s}_n = -\sum_{k=1}^{p} a_k s_{n-k}$$

- Questa tecnica ereditata dall'elettronica può essere utilizzata anche per predire il comportamento di guasto di una macchina.

Predizione dell'affidabilità

Predizione tramite distribuzioni Weibull ed Iperesponenziali

- *Distribuzione Weibull*

$$f_w(x) = \alpha\beta^{-\alpha}x^{\alpha-1}e^{-(x/\beta)^\alpha}$$

$$F_w(x) = 1 - e^{-(x/\beta)^\alpha}$$

- *Distribuzione Iperesponenziale*

$$f_H(x) = \sum_{i=1}^{k} p_i\lambda_i e^{-\lambda_i x}$$

$$F_H(x) = 1 - \sum_{i=1}^{k} p_i e^{-\lambda_i x}$$

Predizione dell'affidabilità

Network Weather Service

- Il *Network Weather Service* (NWS) è un sistema estensibile per la generazione dinamica di previsioni sulle prestazioni delle risorse negli ambienti di calcolo distirbuito
- É composto da due elementi principali:
  - *Sensori*
  - *Predittori*
- Dispone di vari metodi di previsione:
  - Media (adattativo)
  - Mediana (adattativo)
  - Modelli autoregressivi
  - Sezione dinamica della previsione

Predizione dell'affidabilità

Predittori Ibridi

- Ho diversi metodi di previsione, ma qual'è più accurato sulla macchina in esame?
- Siano dati *N* metodi di previsione e l'insieme delle *m* misurazioni totali rilevate sulla macchina *dataset*, con la grandezza di quest'ultimo $m \geq 20$, allora si procede in tal maniera:
  1. si divide *dataset* in due insiemi uguali, al primo insieme si darà il nome di *trainingset*, al secondo insieme il nome di *testingset*;
  2. si ottengono *N* valori predetti usando come misurazioni i valori all'interno di *trainingset*;
  3. viene scelta una predizione alla volta e si calcola il residuo rispetto ad ogni misurazione contenuta nell'insieme *testingSet* e si calcola l'*errore residuo medio*;
  4. si seleziona come "vincente" la previsione che ha ottenuto il *minimo errore residuo medio*;
  5. si calcola la previsione finale utilizzando il metodo di previsione vincente sopra tutto il *dataset*.

Implementazione dei metodi di previsione

Implementazione di politiche di scheduling knowledge-aware

Realizzazione di tre politiche di scheduling all'interno di OurGrid:

1. *chooseGuMpredUptime()*: assegna al task la macchina con maggiore affidabilità predetta;
2. *chooseGuMeffCpu()*: assegnare al task la macchina con il miglior:

$$EffCpu = clockrate * cpuloadpredetto$$

3. *myChooseGum()*: assegna al task la macchina con il miglior:

$$E_i = \alpha \frac{EffCpu}{\sum_{j=1}^{n} EffCpu_j} + \beta \frac{Uptime_i}{\sum_{j=1}^{n} Uptime_j} \qquad \text{con } \alpha + \beta = 1$$

Implementazione dei metodi di previsione

Predittori dell'affidabilità

- Realizzazione di una libreria in Java per la predizione dell'affidabilità delle macchine chiamata *libPredictor*.
- Creazione di quattro predittori:
  - NwsPredictor
  - LinearPredictor
  - BrevikPredictor
  - HybridPredictor
- La libreria è progettata per essere:
  - *estensibile* tramite una ingegnerizzazione Object Oriented;
  - *veloce* tramite l'utilizzo di chiamate native utilizzanti le *Java Native Interface* (JNI)

Implementazione dei metodi di previsione

Simulazione

- Per effettuare uno studio sui metodi di previsioni studiati ed implementati vi è il bisogno di strumenti di simulazione opportuni.
- Realizzazione di un *Simulatore ad eventi discreti* e di strumenti per:
  - generare un insieme di misurazioni pregresse tipiche di scenari Desktop Grid;
  - simulare lo scheduling di *Bag of Tasks* (BoT);
  - verificare ed ordinare i dati in fogli di calcolo.
- Il simulatore e gli strumenti di generazione ed analisi dei risultati sono stati scritti in Java e UNIX Shell Script.

Analisi dei risultati

Scenari di sperimentazione

Quattro tipologie di scenari di simulazione:

	Bassa Affidabilità	*Alta Affidabilità*
Bassa Potenza Computazionale	Scenario 2	Scenario 1
Alta Potenza Computazionale	Scenario 4	Scenario 3

Caratteristiche di ciascuno scenario:

Scenario	*CpuPower*	*DispoVary*	*RepairTime*
scenario1	$U(1,2)$	$W(\alpha,\beta)$	300s
scenario2	$U(1,2)$	$U(12000,36000)$	3600s
scenario3	$N(5,1.5)$	$W(\alpha,\beta)$	300s
scenario4	$N(5,1.5)$	$U(3600,10800)$	3600s

Analisi dei risultati

Confronto dell'accuratezza dei metodi di previsione (1)

Questo livello di analisi mira a determinare l'*accuratezza* dei vari metodi di previsione. Descriviamo in breve come si è proceduto:

1. Si sono generati tramite le distribuzioni scelte per ogni scenario diversi tempi di guasto:

# guasti generati	20	100	300	500	700	1000
# minimo (20%)	4	20	60	100	140	200

2. Si è usato il 20% di esse per generare la prima previsione p_m.
3. Si è confrontata p_m con il tempo di guasto f_{m+1} ricavandone un errore assoluto

$$ErrAss_m = p_m - f_{m+1}$$

ed un errore relativo

$$ErrRel_m = ErrAss_m / f_{m+1}$$

4. Si calcolano $ErrAss_m$ e $ErrRel_m$ incrementalmente fino ad ottenere $n - m$ predizioni.

Analisi dei risultati

Confronto dell'accuratezza dei metodi di previsione (2)

Si sono avuti i seguenti risultati:

- Mancanza di un metodo di previsione accurato in ambienti con macchine estremamente affidabili (Scenario 1 e 3).
- Mancanza di un incremento d'accuratezza rilevante con l'aumentare dello storico (anzi in 1 e 3 viceversa).
- L'utilizzo di distribuzioni Weibull o Iperesponenziali non si adattano meglio in scenari ad alta affidabilità.
- Negli scenari dove le macchine sono poco affidabili i migliori risultati sono dati da NWS e dal metodo di Previsione Ibrida con errori relativi medi da $0.23 \leq e_r \leq 0.25$.

Analisi dei risultati

Applicazione dei metodi di previsione nella schedulazione (1)

Si sono effettuate simulazioni utilizzanti la *chooseGuMpredUptime()* per determinare se l'utilizzo di metodi di previsione dell'affidabilità portino un risparmio in *tempo di completamento medio dei task*. Si è proceduto in tal maniera:

1. Si sono generati tramite una $U(18000, 54000)$ i tempi di esecuzione di un singolo task considerata una macchina di potenza computazionale unitaria.
2. Per ogni scenario si sono sottomessi diversi gruppi di task variando il *Resource Rate* (RR) definito come:
$$RR = \frac{NumeroDeiTask}{NumeroMacchineGrid}$$
dove i gruppi sottomessi per ogni scenario sono i seguenti:

RR	0.3	0.5	0.7	1	3	5	7	10	20	50
Task	5	8	11	15	45	75	105	150	300	750

3. Si è confrontata la differenza tra il tempo totale della schedulazione per ogni metodo data la politica *chooseGuMpredUptime()* con quelli di una *Workqueue*.

Analisi dei risultati

Applicazione dei metodi di previsione nella schedulazione (2)

I risultati ottenuti sono stati i seguenti:

- Negli scenari con macchine molto affidabili, la politica si dimostra pari ad una politica *knowledge-free*.
- Nello Scenario 2 caratterizzato da macchine poco affidabili e poco potenti l'utilizzo dei metodi di previsioni offre guadagni da 20.25 secondi fino a 420.96 secondi.
- Nello Scenario 4 caratterizzato da macchine poco affidabili, la politica si dimostra pari ad una politica *knowledge-free*.

Conclusioni

Sommario dei contributi

- Studio comparativo dei vari metodi di previsione
- Analisi dei vantaggi sull'uso di informazioni inerenti l'affidabilità delle macchine in politiche di scheduling knowledge-aware
- Creazione di un metodo di previsione ibrida
- Creazione di una libreria per la predizione dell'affidabilità

Conclusioni

Lavori futuri

- Creazione di metodi di previsione a lungo termine
- Studio ed implementazione di politiche knowledge-aware miste
- Nuovi metodi di previsione ibrida
- Estensione e miglioramento della libreria di previsione

Tabelle dei risultati

Accuratezza delle previsioni per lo Scenario 1

20	Nws	Linear	Brevik	Hybrid
ErrAss	808202499.02	574856150.81	521831017.17	808202499.02
ErrRel	0.99	1.28	**0.98**	0.99
100	Nws	Linear	Brevik	Hybrid
ErrAss	406581601	621703885.56	281702702.7	294782926.31
ErrRel	**0.98**	1.86	0.99	0.99
300	Nws	Linear	Brevik	Hybrid
ErrAss	526394654.5	937252710.37	306085941.52	313152173.68
ErrRel	1.28	2.15	0.99	**0.99**
500	Nws	Linear	Brevik	Hybrid
ErrAss	607490392.43	817293966.52	335201062.2	338361999.63
ErrRel	1.32	1.98	0.99	**0.99**
700	Nws	Linear	Brevik	Hybrid
ErrAss	630661531.62	955426047.16	326669786.35	337237755.04
ErrRel	1.33	2.33	0.99	**0.99**
1000	Nws	Linear	Brevik	Hybrid
ErrAss	534305620.01	876669838.43	327093354.53	328661286.48
ErrRel	1.48	2.43	0.99	**0.99**

Tabelle dei risultati

Accuratezza delle previsioni per lo Scenario 2

20	Nws	Linear	Brevik	Hybrid
Errore assoluto	6728.14	6938.28	10119.49	7775.47
Errore relativo	**0.25**	0.32	0.44	0.28
100	Nws	Linear	Brevik	Hybrid
Errore assoluto	6341.17	7092.2	10594.69	6821.65
Errore relativo	**0.25**	0.29	0.44	0.28
300	Nws	Linear	Brevik	Hybrid
Errore assoluto	6023.48	6651.45	10438.16	6470.42
Errore relativo	**0.24**	0.28	0.44	0.25
500	Nws	Linear	Brevik	Hybrid
Errore assoluto	5789.78	6585.18	10281.06	6276.15
Errore relativo	**0.23**	0.27	0.43	0.26
700	Nws	Linear	Brevik	Hybrid
Errore assoluto	5968.03	6632.72	10286.47	6342.58
Errore relativo	**0.24**	0.28	0.43	0.26
1000	Nws	Linear	Brevik	Hybrid
Errore assoluto	5892.86	6617.63	10243.02	6287.41
Errore relativo	**0.23**	0.27	0.43	0.25

Tabelle dei risultati

Accuratezza delle previsioni per lo Scenario 3

20	Nws	Linear	Brevik	Hybrid
Errore assoluto	458700245.8	620880076.72	451858602.97	358504933.53
Errore relativo	**0.91**	1.32	0.96	0.99
100	Nws	Linear	Brevik	Hybrid
Errore assoluto	499756987.92	1016734981.81	285474367.88	316649595.35
Errore relativo	**0.9**	1.79	0.99	0.99
300	Nws	Linear	Brevik	Hybrid
Errore assoluto	558753408.89	904149554.53	346749119.55	346749119.55
Errore relativo	1.16	2.08	0.99	**0.99**
500	Nws	Linear	Brevik	Hybrid
Errore assoluto	663455234.24	1165517147.36	345333949.14	364751695.99
Errore relativo	1.43	2.45	0.99	**0.99**
700	Nws	Linear	Brevik	Hybrid
Errore assoluto	624534634.13	917809123.51	337200030.28	337765089.17
Errore relativo	1.69	2.46	0.99	**0.99**
1000	Nws	Linear	Brevik	Hybrid
Errore assoluto	547913664.45	754048285.57	245071435.46	249137946.47
Errore relativo	1.51	2.33	0.99	**0.99**

Tabelle dei risultati

Accuratezza delle previsioni per lo Scenario 4

20	Nws	Linear	Brevik	Hybrid
Errore assoluto	1979.97	2018.61	2973.76	2279.48
Errore relativo	**0.26**	0.3	0.44	0.33
100	Nws	Linear	Brevik	Hybrid
Errore assoluto	1835.16	2021.84	3141.14	1999.09
Errore relativo	**0.24**	0.27	0.43	0.28
300	Nws	Linear	Brevik	Hybrid
Errore assoluto	1802.96	2011.88	3005.44	1934.69
Errore relativo	**0.24**	0.27	0.43	0.25
500	Nws	Linear	Brevik	Hybrid
Errore assoluto	1818.88	1908.83	3042.72	1883.53
Errore relativo	**0.24**	0.27	0.43	0.25
700	Nws	Linear	Brevik	Hybrid
Errore assoluto	1826.71	2004.46	3031.17	1934.63
Errore relativo	**0.24**	0.27	0.43	0.26
1000	Nws	Linear	Brevik	Hybrid
Errore assoluto	1750.43	1983.44	3019.45	1880.89
Errore relativo	**0.24**	0.27	0.44	0.26

Tabelle dei risultati

Tempi di completamento relativi ad una Workqueue nello Scenario 1

RR	Task	Nws	Linear	Brevik	Hybrid
0.3	5	-1.63	-1.65	-1.63	**-1.63**
0.5	8	0	-0.77	0	**0**
0.7	11	-1.5	-1.57	-1.5	**-1.5**
1	15	-3.03	-3.41	-3.03	**-3.03**
3	45	3.02	3.02	3.02	**3.02**
5	75	-1.78	**-1.4**	-1.74	-1.74
7	105	-1.74	**-1.65**	-1.76	-1.76
10	150	-3.16	**-3.06**	-3.11	-3.11
20	300	-5.01	**-4.91**	-5.04	-5.04
50	750	2.41	2.41	2.41	**2.41**

Tabelle dei risultati

Tempi di completamento relativi ad una Workqueue nello Scenario 2

RR	Task	Nws	Linear	Brevik	Hybrid
RR	Lavori	Nws Ass.	Linear Ass.	Brevik Ass.	Hybrid Ass.
0.3	5	337.39	337.39	**338.53**	337.39
0.5	8	31.4	33.22	20.25	**35.24**
0.7	11	260.26	256.92	**261.17**	246.24
1	15	37.17	52.98	**55.63**	48.41
3	45	81.26	111.35	109.03	**113.73**
5	75	351.89	**420.96**	415.76	385.99
7	105	97.38	102.48	95.56	**146.84**
10	150	**323.65**	306.51	305.22	307.94
20	300	**145.86**	80.07	48.21	30.67
50	750	255.84	288.77	240.7	**289.49**

Tabelle dei risultati

Tempi di completamento relativi ad una Workqueue nello Scenario 3

RR	Task	Nws	Linear	Brevik	Hybrid
RR	Lavori	Nws Ass.	Linear Ass.	Brevik Ass.	Hybrid Ass.
0.3	5	-1.6	0	0.21	**0.21**
0.5	8	1.18	2.35	2.35	**2.35**
0.7	11	-1.52	-0.88	-0.88	**-0.88**
1	15	-1.58	-0.24	-0.24	**-0.24**
3	45	0.02	**0.11**	0.08	0.08
5	75	**0.73**	0.63	0.62	0.62
7	105	0.62	0.58	0.71	**0.71**
10	150	0.43	**0.48**	0.41	0.41
20	300	0.59	**1.71**	0.57	0.57
50	750	0.97	0.97	0.97	**0.97**

Tabelle dei risultati

Tempi di completamento relativi ad una Workqueue nello Scenario 4

RR	Task	Nws	Linear	Brevik	Hybrid
0.3	5	0	0	0	**0**
0.5	8	-0.47	-0.47	-0.47	**-0.47**
0.7	11	4.02	4.02	4.02	**4.02**
1	15	-1.14	-1.14	-1.14	**-1.14**
3	45	-0.25	-0.83	-4.92	**0.51**
5	75	-0.14	-1.35	-0.71	**0.08**
7	105	**-2.26**	-2.69	-2.85	-2.34
10	150	3.38	4.75	3.38	**4.75**
20	300	3.26	3.26	3.26	**3.26**
50	750	1.27	1.27	-1.17	**1.27**