**BETWEENNESS CENTRALITY**

L’algoritmo per il calcolo della betweenness centrality prevede il calcolo di tutti gli short paths tra ogni coppia di nodo, il quale risulta estremamente oneroso.  
La versione Naive di tale algoritmo ha richiesto infatti circa 6 ore e 15 minuti per essere eseguita.  
Di conseguenza è stata implementata una versione parallela attraverso la suddivisione dei nodi in j parti al fine di ottenere tempi di esecuzione migliori. Inoltre, la parallelizzazione ha permesso comunque di ottenere gli stessi risultati della versione Naive, risultando essere un ottimo approccio.Con l’utilizzo di 2 job si è ottenuto però solo un miglioramento di circa 1 ora e 30 minuti, ottenendo un tempo di esecuzione pari a 4 ore e 40 minuti.Con l’utilizzo di 4 job, invece, si è ottenuto un adeguato miglioramento, il quale ha impiegato 2 ore e 20 minuti per essere eseguito.Non sono stati provati più di 4 job a causa dei limiti fisici del computer su cui è stato testato l’algoritmo, ma, in generale, al fine di ottenere tempi di esecuzione più brevi possibili, si consiglia la scelta di un numero di job pari al numero di core logici presenti sull’elaboratore.Tale tempo di esecuzione risulta però ancora eccessivo, di conseguenza si è deciso di effettuare un random sampling per ottenere tempi accettabili. Tale sampling è risultato essere un’ottima soluzione, offrendo un ottimo trade-off tra tempi di esecuzione e accuratezza. Di seguito sono mostrati due grafici che contengono rispettivamente i tempi di esecuzione e il numero di nodi della top500 presenti anche nella top500 della versione Naive per ogni fattore di sampling testato (0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5).

