Radhika Pasari 2012B4A7445P
Aakriti Jha 2012B1A7840P

Assumptions:

1. Five News items have beeen comsidered which have topic as integer value(later mapped to correspondong news) and timestamp as unquie integer value.
2. The news items are numbered as 1.txt,2.txt uptill the number of processes needed and are read by corresponding reparters
3. Inter-reporter/ Inter-editor communication happens when the buffer of each reporter of size 5 gets full.
4. In BNPM, partition size has to be a multiple of total number of processes.
5. Greater timestamp indicates more recent news.

Algorithm:

Bnps:

- Initial Editor size (or number of reporters per partition) is taken as 4. So, the first process in partition is the editor process and remaining are reporters
- Reporters open file corresponding to their ids and store the topic and timestamp in a structure “news”
- After all have populated their news array completely(5 news pieces), message passsing happens between processes corresponding to each reporter
- A reporter sends news only to the reporters with higher id number present in its partition and the respective reporters recive this news
- After the news is received,the topic is compared with reporter's own news topic and if it does not match match then 1 is sent,else timestamp comparison is done and if the incoming news is not latest(i.e timestamp is smaller than this reporter's timestamp),then a 0 value is sent.
- The reporter collects the boolean values from all higher id reporters and after taking a union of the set,sums the number of valid news it has and sends its recent, valid news to the editor.
- The editor now call MPI_Reduce to sum all valid news and recieve those many news pieces and then prints it.

Bnmp:

- All steps are same as that in bnps except the last step. Summing also happens in an array where each element corresponds to the number of valid news per partition.
- After the reporters have sent their news to editor,it stores the unique news in an array
- Editor then sends news structure to all higher id editors and similar message passing as that in bnps happens.
- Finally, after receiving all messsages,the editors print their unique news.

Components:

- bnps.c:implementation of bnps
- bnmp.c:implementation of bnmp.c
- 1.txt till 11.txt: news files
- readme
- Graphs

Radhika Pasari 2012B4A7445P
Aakriti Jha 2012B1A7840P

Scalability:

Bnps:

Number of reporters can be increased as much as you needed as long as text files of the form id.txt are provided, where id is reporter id which goes from 1 to number of reporters.

Bnpm:

Here also number of reporters and number of editors can be increased as much as needed and partition size can be manipulated as long as it is a divisor of total number of processes.

Expression for number of MPI Calls:

Bnps:

No. Of reporters = k
No.of calls = k(k-1) + x, where x = number of valid news across all reporters
= O(k^2)

Bnpm:

No. Of reporters = k, No. Of partitions = p
No.of calls = p[k(k-1) + x] + p(p-1), where x = number of valid news across all reporters

Results:

The following four graphs have been plotted-

1. BNPS: Number of nodes vs time (Number of processes constant =12)
2. BNPS: Number of processes vs time (Number of nodes constant =2)
3. BNPM: Number of nodes vs time(Number of processes constant=12, Number of partitions constant =3)
4. BNPM: Number of parttions vs time(Number of processes constant=12, Number of nodes constant =2)