# CS 6210 Fall 2011 Final Solution with Grading Rubric

Name:_____Kishore_____GT Number: _____

**Monday December 12, 2011 (8:00 AM to 10:50 AM)**
**Expected time to finish the exam: 111 mins**

**Note:**

1. **Write your name and GT number on each page.**
2. The test is **CLOSED BOOK** and **NOTES.**
3. **Please provide the answers in the space provided.** You can use scratch paper (provided by us) to figure things out (if needed) but you get credit **only** for what you put down in the space provided for each answer.
4. For conceptual questions, **concise bullets** (not wordy sentences) are preferred.
5. Where appropriate **use figures** to convey your points (a figure is worth a thousand words!)
6. **Illegible answers are wrong answers. WE MEAN IT!!**
7. Please look through the whole test before starting so that you can manage your time better.

Good luck!

| Question number | Points earned | Running total |
|---|---|---|
| 1 (Max: 1 pts) | | |
| 2 (Max: 9 pts) | | |
| 3 (Max: 20 pts) | | |
| 4 (Max: 20 pts) | | |
| 5 (Max: 20 pts) | | |
| 6 (Max: 20 pts) | | |
| 7 (Max: 10 pts) | | |
| Total(Max: 100 pts) | | |

1. (1 min, 1 point) "A Needle in the Haystack" reminds you of

   a) Google file system
   b) Amazon's key-value store
   c) Coral CDN
   d) Facebook photo store
   e) many other useless phrases in the English language!

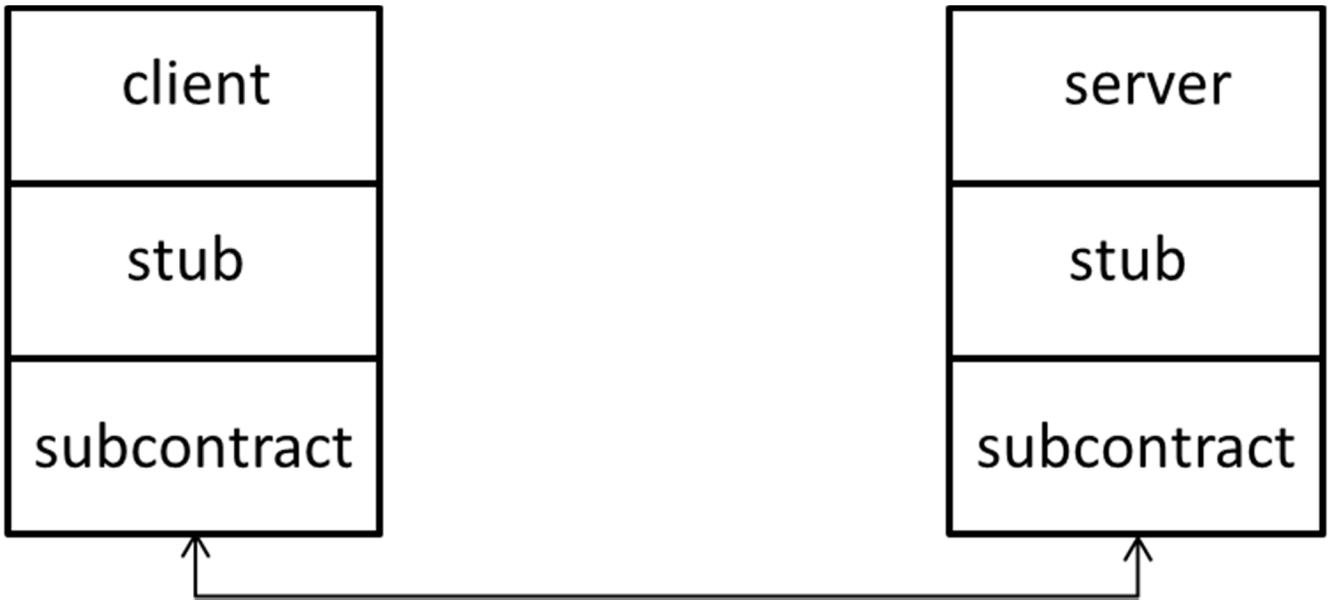
   **+1 everyone gets regardless of answer**

# CS 6210 Fall 2011 Final Solution with Grading Rubric

Name:_____Kishore_____GT Number: _____

**2. Distributed Objects and Middleware (10 min, 9 points)**

(**Spring**) In the Spring Kernel, a client-server interaction is represented as shown in this Figure:




Explain the role played by the subcontract in this interaction (**bullets please not wordy sentences).**

a) Replaceable layer in the RPC stack that sits on top of the network layer of the OS
b) Performs the heavy-lifting that needs to be done in Marshalling and unmarshalling (args/results) on both sides by learning the types of objects involved in the RPC call from the client/server stubs above, and knowing the specifics of the network layer below
c) Optimize the marshaling process commensurate with the location of client and server (e.g., shared memory between them)
d) On the server side, interface allows turning language-specific objects (e.g., C++) into Spring objects to facilitate creation of client objects and communication end-points in the clients.
e) On the server side interface allows shutting down the service
f) Process incoming calls to the server and after constructing the local state from the communication buffer , pass it up to the server stub for server object invocation
g) Subcontract is extensible and can be discovered/installed at runtime (i.e., dynamic), allowing transparent migration of servers
h) Subcontract allows incremental evolution of the server (e.g., singleton, replicated, cached, etc.)

**+1.5 for each of b, c, f, g**
**+0.5 for each of a, d, e, h**
**1 point gratis for some knowledge**

# CS 6210 Fall 2011 Final Solution with Grading Rubric

Name:_____Kishore_____GT Number: _____

**3. GMS, DSM and File Systems (25 min, 20 points)**

(a) (10 min, 5 points) (**GMS**) Recall that in Global Memory System (GMS), upon page replacement, the replaced page is sent to a peer memory on the LAN rather than the local disk. To implement this scheme, the paper describes an algorithm that is divided into epochs.

(i)(3 points) In each epoch, the current initiator of the algorithm collects the age information from all the nodes, and sends each node a weight vector that represents the distribution of the oldest pages across all the nodes. How does each node use the weight vector?

- On a page fault, a node picks the oldest page from the global cache (or the oldest page from its local cache, if its global cache is empty). Let this be X.
- It discards X (writes to disk if dirty) if Age(X) > MinAge
- Else it sends X to a peer node Ni; The choice of picking of Ni is proportional to the weight Wi associated with the node Ni.

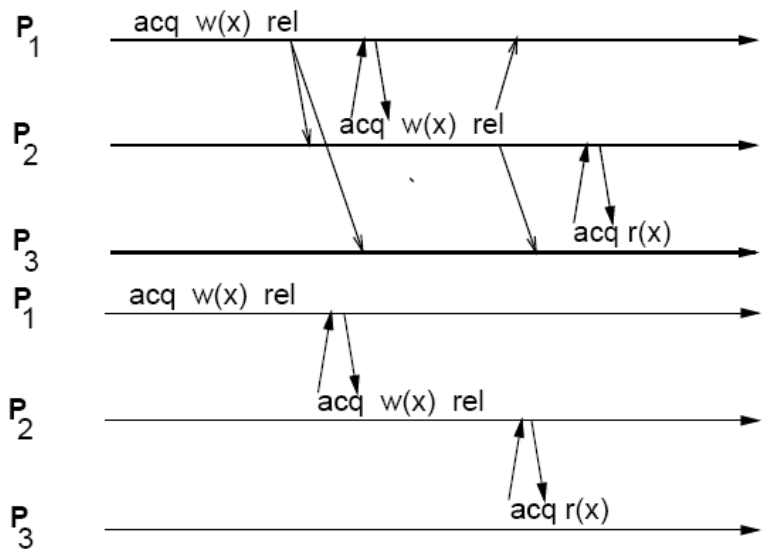
**+3 if answer contains third bullet; else +1 for each of first two bullets**

(ii) (2 points) The current initiator also selects the initiator node for the next epoch as the one that has the largest share of oldest pages in the system. What is the rationale for this choice?

- Let node Ni have the maximum share of the M pages to be replaced in the current epoch; this implies that Ni has been the least active one in the past epoch. In other words, this node has been mainly a memory server for the other nodes in the previous epoch. Therefore, it makes sense to make this the initiator for the next epoch.

**+2 if the highlighted phrase in the answer**

(b) (5 min, 5 points)(**DSM**)What is the difference between eager release consistency and lazy release consistency? Use figures to get your point across.




- Eager: send coherence actions to all modified data **before lock release** +2
- Lazy: Learn and perform the coherence actions **at the point of lock acquisition** +2
- Eager: May incur unnecessary communication overhead if there is not much active sharing of state across synchronization epochs +1
- Lazy: May incur latencies for reconstructing shared state at the point of lock acquisition if there is a lot of active sharing

# CS 6210 Fall 2011 Final Solution with Grading Rubric

Name:_____Kishore_____GT Number: _____

(c) (5 min, 5 points)(**Coda**) Recall that Coda distributed file system supports disconnected operation, i.e., a client workstation can get a file from a server and work locally even in the absence of network connection to the server. In order to support disconnected operation and yet provide Unix semantics, Coda supports a "callback" mechanism. Explain how this mechanism works.

+2
- When client opens a file, the client caches the entire file in its local disk, and register a "callback" with one of the replicated servers.

+2
- If another client writes to this file, then the server **informs** the client (for whom callbacks are registered with the server) that the file has been modified

+1
- Callback is a mechanism and not a policy; i.e., it allows a client to make a determination as to how to use this information about the change to the file

(d) (5 min, 5 points)(**xFS**) We say that metadata management in xFS is "dynamic". Explain what exactly is meant by this statement.

+3
- "File" to "manager" mapping is not static as in NFS. The metadata for a file (e.g., i-node) **may not be at the same node as the location of the file** itself.

+2
- The metadata for a file can be **migrated to a different node in the network** transparently for balancing the load on the servers for metadata management.

# CS 6210 Fall 2011 Final Solution with Grading Rubric

Name:_____Kishore_____GT Number: _____

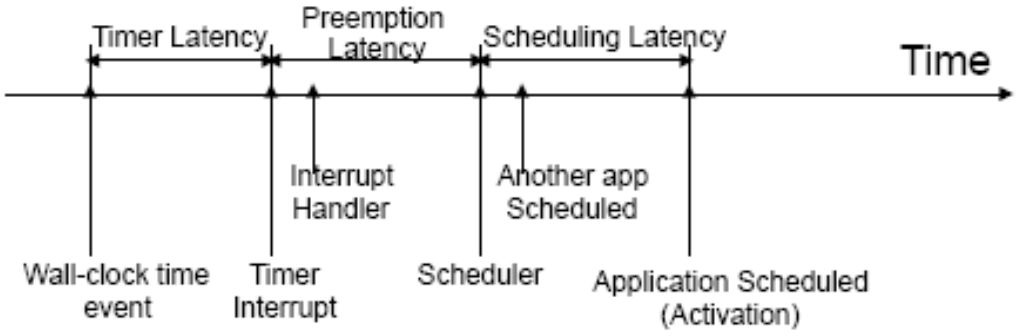
**4. Real-time and Multimedia (20 min, 20 points)**

(a) (10 min, 10 points) (**RT Linux**) Define the following terms(use figures to help get your point across)

(i) (2 points) "timer latency"

(ii) (2 points) "preemption latency"

(iii) (2 points) "scheduling latency"




- Timer latency: distance between event occurrence and timer interrupt due to the granularity of the timing mechanism
- Preemption latency: distance between timer interrupt and opportunity to schedule the event due to activity on the CPU being non pre-emptible (e.g., kernel in the middle of an interrupt handler)
- Scheduling latency: distance between when the event becomes schedulable and when it actually gets scheduled due to higher priority applications already in the CPU scheduling queue

**+2 for each highlighted phrase or equivalent**

(iv)(4 points) How do these terms affect a latency sensitive application?

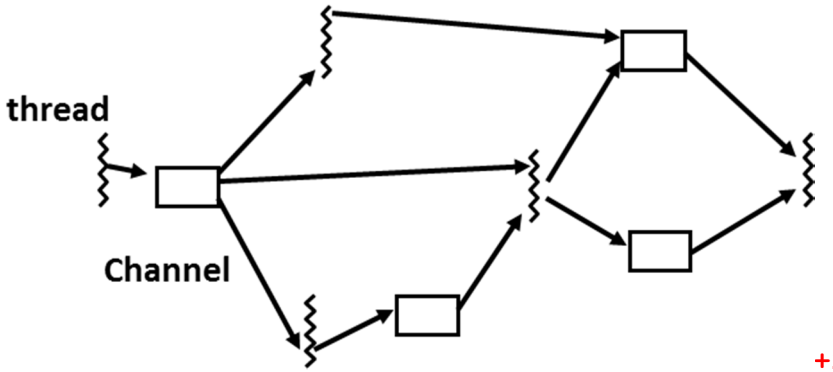
- Latency sensitive applications may have a tight "deadline" between sensing and actuation. These latencies will hurt meeting such deadlines.
- Further, there could be priority inversion if the event happened on behalf of a high priority process but the "server" that handles the event is lower in priority compared to what processes are already in the CPU scheduling queue

**+2 for each highlighted phrase or equivalent**

# CS 6210 Fall 2011 Final Solution with Grading Rubric

Name:_____Kishore_____GT Number: _____

(b) (10 min, 10 points) (**PTS**) How does the PTS computational model (with PTS threads and channels) differ from a Unix distributed program written with sockets and processes? Use figures to get your point across.




+2 for figure

- PTS threads
  a) "time" is a first class entity manipulable from PTS thread to associate timestamp with data items put into channels
  b) Since a PTS thread has a handle on "time" it can use it for correlating input items from multiple streams based on time, and also propagate temporal causality through the computation graph

  +1.5 for a; 0.5 for b
- PTS channels
  a) They support **many-many** connection as opposed to the one-one connection of Unix sockets
  b) A PTS channel as opposed to a Unix socket is not just a transport abstraction; it allows writing stream manipulation code based on time and naturally admits stream synchronization and persistence
  c) The contents of a Unix socket is a stream of bytes; the contents of a PTS channel is also a stream of bytes but **indexed** by **wall-clock time**
  d) A PTS thread can an **"put"** item (a stream of bytes) associating a wall-clock time as a **timestamp**
  e) A PTS thread can **"get"** items from a channel that span a time window ("lower bound" and "upper bound")
  f) Items in a PTS channel are **automatically garbage collected** or **persisted on permanent storage** depending on the properties associated with the channel

  +1.5 for each of a, c, and f
  +0.5 for each of b, d, and e

# CS 6210 Fall 2011 Final Solution with Grading Rubric

Name:_____Kishore_____GT Number: _____

**5. Failures, Consistency, and Recovery (20 min, 20 points)**

(a) (10 min, 10 points) (**LRVM**)

(i) (6 points) Give two examples of OS subsystems and how they may benefit from a persistent virtual memory.

+2
- File system
  a) Both the metadata (e.g., i-nodes in Unix file system) and the user data in the files need to be persistent. For efficiency it is customary to have them cached in virtual memory. Persisting the virtual memory avoids the need for synchronous (i.e., blocking writes) writes to the permanent storage.

+1 how

+2
- Programming language runtime/libraries with persistent objects
  a) Long running applications often benefit from abstractions that allow persisting the state of the computation for checkpoint/restart. Persistent VM will allow the efficient implementation of such abstractions

+1 how

(ii) (4 points) Why is it better to keep the changes to virtual memory in log segments in implementing a persistent virtual memory?

- Memory footprint of applications that would need persistence support is usually large (e.g., file system).
- Writes to VM that may need persistence may be scattered all over the address space.
- This will lead to the familiar "small write" problem on the disk leading to performance inefficiency.
- This is the reason for writing all the changes to the virtual address space as log records of the form <VM offset, new value> contiguously to a Log file will be an efficient solution to avoid the small write problem.

**+1 for each highlighted phrase or equivalent**

# CS 6210 Fall 2011 Final Solution with Grading Rubric

Name:_____Kishore_____GT Number: _____

(b) (5 min, 5 points)(**RioVista**) The Vista RVM library is 700 lines of code as opposed to 10K lines for a comparable functionality implemented in LRVM. Explain why.

- Two problems lead to system failure
  1. Power failure
  2. Software crash

+2

- RioVista does away with the first problem with a battery backed file cache
- Vista is an RVM that sits on top of the reliable Rio file cache

+0.5

- It uses "undo" log mapped to the Rio file cache for restoring old values to VM in case of abort or system crash

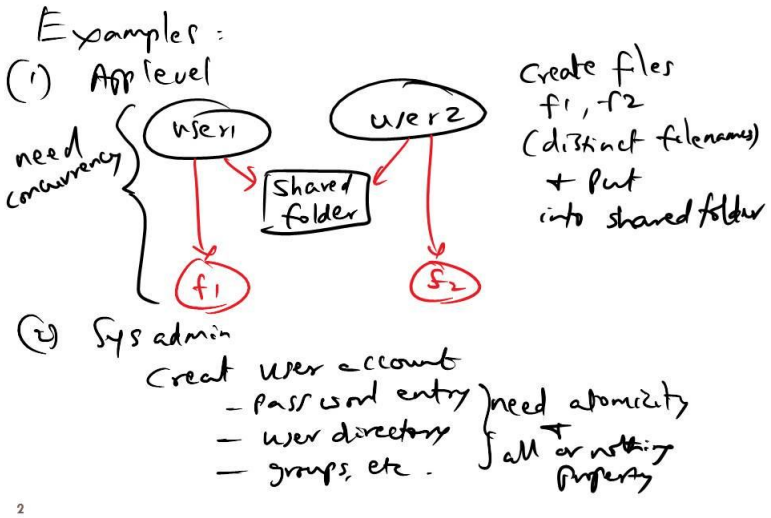
+2

- It does not have redo logs or truncation code

+0.5

- Checkpointing and recovery code simplified due to "undo" log in Rio file cache

(c) (5 min, 5 points) (**TxOS**) Give one user level example and one system level example where transactional semantics will help in increasing concurrency while ensuring atomicity and isolation for the actions.




- User level:
  - User1 and User 2 can create f1, and f2, respectively, concurrently.
  - They need atomicity and isolation to update the shared folder into which they want to place the files.
- System level
  - Need all or nothing property to all the actions shown pertaining to an account creation

**+2 for each example above left**
**+1 for the details in the text box above right**

# CS 6210 Fall 2011 Final Solution with Grading Rubric

Name:_____Kishore_____GT Number: _____

**6. Internet Scale Computing (25 min, 20 points)**

(a) (5 min, 5 points) (**Giant-scale services**) What do "D" and "Q" stand for in the DQ principle? Explain how the DQ principle helps in handling graceful degradation of giant scale services.

- D is the "Harvest" defined as $D = D_v/D_f$, where $D_v$ is the available data, and $D_f$ is the complete data in the giant-scale service
- Q is the "Yield" defined as $Q = Q_c/Q_o$, where $Q_c$ is the queries completed, and $Q_o$ is the queries offered to the giant-scale service in a given time period
- Graceful degradation:
  - DQ is a constant for a system capacity
  - If system capacity goes down (due to failure, routine maintenance, etc.)
    - Keep harvest (D) unchanged and reduce the Yield (Q)
    - Reduce harvest (D) and keep the Yield unchanged (Q
    - Allows system administrator to make an explicit choice as to which to preserve and which to sacrifice

+1

**+1 for each highlighted phrase or equivalent**
**+1 for the two bullets identified above**

(b) (5 min, 5 points) (**Coral CDN**) Explain the metadata server overload that can happen in a traditional DHT with an example.

- Assume 10 nodes each generate content. Assume that the hashed keys for all the 10 generated content are all clustered around the IP-address hash of one particular node say N1. In this case the key-value pairs, <key1, node-id1>, <key2, node-id2>, …, <key10, node-id10> will all be stored in the same node N1. If all these 10 generated content are the most popular content in the CDN, then, even though the content themselves are hosted on distinct nodes, the discovery of the content location requires accessing the metadata stored in N1. This leads to an overload of the metadata server N1.

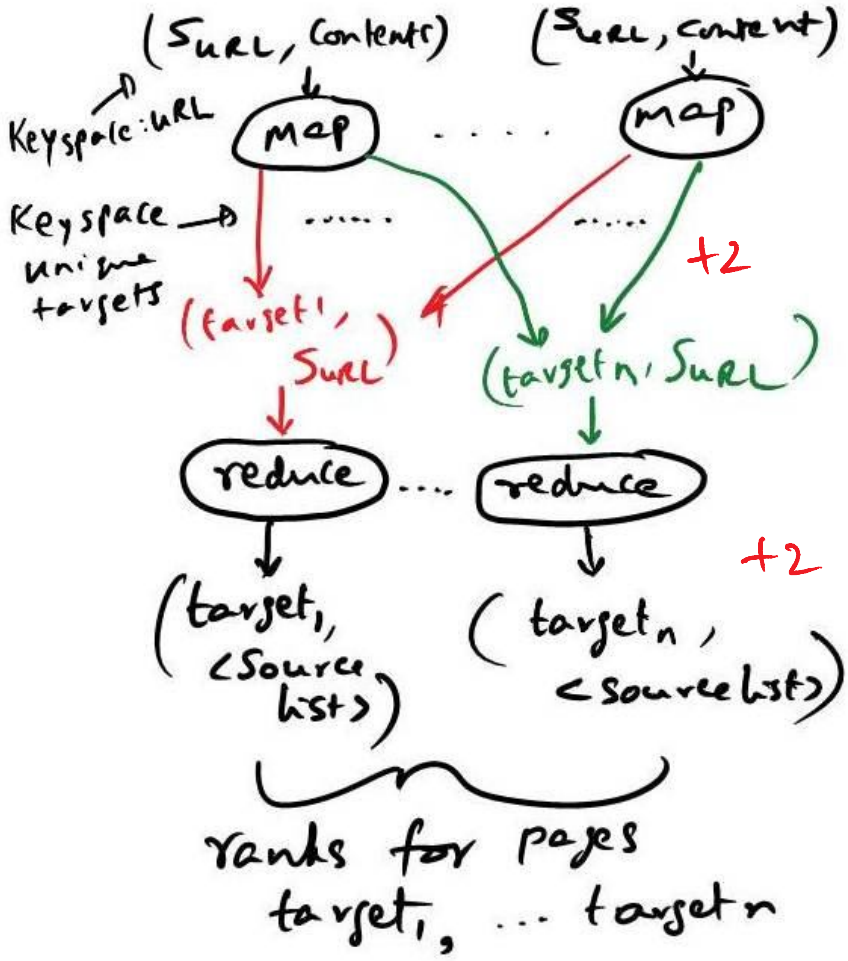
. . . .

**+5 if the answer says that a lot of hashed content keys are clustered around the hash of the IP address**

# CS 6210 Fall 2011 Final Solution with Grading Rubric

Name:_____Kishore_____GT Number: _____

(c) (15 min, 10 points) (**Map-Reduce**) Show pictorially a Map-Reduce structure (you don't have to write the code for map and reduce functions) for ranking the pages on the web. Be explicit in identifying the <key, value> of the map and reduce functions both for input and output.



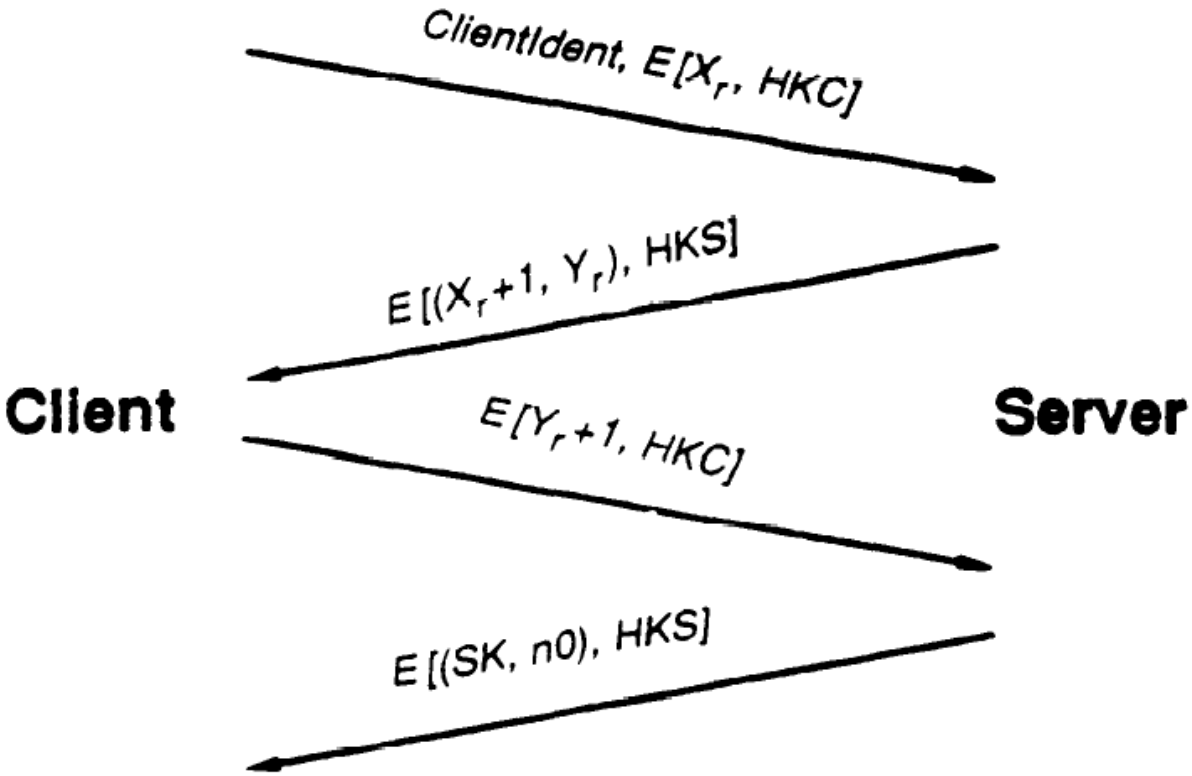

- <Key, value> input to map function:
  - <source URL, content of the source page>
- <Key, value> output of Map and input to Reduce
  - <target URL, URL of source page where found>
- <Key, value> output of Reduce
  - <target URL, rank of the target URL>

+2 +2 +2

# CS 6210 Fall 2011 Final Solution with Grading Rubric

Name:_____Kishore_____GT Number: _____

**7. Security (10 min, 10 points)**
(**AFS**)




With respect to the above client-server interaction in the Andrew File system, answer the following questions:

(a) (2 points) $X_r$ is a random number generated by the client. What purpose does this serve?

On receiving the response from the server E[($X_r$+1, $Y_r$], HKS], the client can decrypt and retrieve $X_r$ from the response. If the extracted $X_r$ is the same as what was sent by the client, it helps establish the genuineness of the server to the client. That is, this safeguards against replay attack on the client.

**+2 for the highlighted phrase or equivalent**

(b) (2 points) $Y_r$ is a random number generated by the server. What purpose does this serve?

On receiving the response from the client E[($Y_r$+1], HKC], the server can decrypt and retrieve $Y_r$ from the response. If the extracted $Y_r$ is the same as what was sent by the server, it helps establish the genuineness of the client to the server. That is, this safeguard against replay attack on the server

**+2 for the highlighted phrase or equivalent**

# CS 6210 Fall 2011 Final Solution with Grading Rubric

## Name:_____Kishore_____GT Number: _____

(c) (2 points) What is "ClientIdent" during an RPC session establishment?

It is the "secrettoken" that would have been obtained during the login of the user on the Virtue workstation.

**+2 for highlighted phrase**

(d) (2 points) Is "ClientIdent" sent in plaintext or encrypted? Why in either case?

It is sent in plaintext. Since AFS uses private key infrastructure, the server needs the id in plaintext to look up the authorization database and retrieve the private key corresponding to this ClientIdent for decrypting the message.

**+1 for each highlighted phrase**

(e) (2 points) What is SK? What is its purpose?

It is a session key generated anew by the server for the new RPC session. This is to prevent the overuse of handshake key (HKC) which has a longer lifetime. For example, HKC during the login process is the password of the user (which has a lifetime determined by the policy in vogue for the system, GT makes us change passwords every 3 months). Similarly, the HKC that Virtue extracts from the "ClearToken" for this user as a result of the login process has a lifetime of the login session (which can be up to 24 hours in the Andrew system).

**+1 for each highlighted phrase**