--Question Starting--  
59. In a Hadoop-based system designed to process and store petabytes of data across thousands of servers, a data engineer is optimizing a MapReduce job that processes time-stamped web log data. The job's performance is suboptimal due to excessive data shuffling between the map and reduce phases. To address this, the engineer proposes modifications to the job configuration. Which of the following modifications is likely to minimize the data shuffling without compromising the integrity of the data processing?  
(1) Increasing the number of reducers  
(2) Decreasing the number of mappers  
(3) Implementing a combiner function in the map phase  
(4) Expanding the HDFS block size  
Answer Key: 3  
Solution:  
• (Incorrect): Increasing the number of reducers might actually increase data shuffling as more data needs to be moved across nodes.  
• (Incorrect): Decreasing the number of mappers could limit parallel processing capabilities, potentially increasing processing time rather than optimizing shuffling.  
• (Correct): Implementing a combiner function can reduce the volume of data transferred between the map and reduce phases by aggregating intermediate data locally on each mapper, thus minimizing network congestion and improving overall job performance.  
• (Incorrect): Expanding the HDFS block size affects data storage and retrieval but does not directly influence the shuffling process between map and reduce phases.  
Hence, Option (3) is the right answer.  
  
--Question Starting--  
62. An advanced language translator is designed to optimize the execution of a high-level language into machine code. The translator uses a sophisticated method to handle the binding times of various constructs within the language. Considering the translator's capability to optimize static and dynamic bindings, which scenario best demonstrates a dynamic binding advantage in the context of polymorphic behavior?  
(1) Linking a function call to its definition at compile time  
(2) Resolving variable types based on their runtime values  
(3) Assigning memory addresses to variables during program execution  
(4) Compiling a loop structure with a fixed number of iterations  
Answer Key: 1  
Solution:  
• (Correct): Dynamic binding, especially in the context of polymorphism, allows a function call to be linked to multiple potential actions (definitions) at runtime, enabling more flexible and adaptable software behavior.  
• (Incorrect): Resolving variable types at runtime is related to type safety and type inference, not directly to polymorphic behavior.  
• (Incorrect): Assigning memory addresses during runtime is more about dynamic memory management rather than binding.  
• (Incorrect): Compiling a loop with a fixed iteration count is a compile-time decision and does not leverage the benefits of dynamic binding.  
Hence, Option (1) is the right answer.  
  
--Question Starting--  
63. In an enterprise network, a security administrator configures a series of cryptographic measures to secure data transmissions between internal systems. This setup includes the application of both symmetric and asymmetric cryptographic algorithms. Which of the following scenarios best illustrates the use of a digital signature to ensure the integrity and authenticity of a communication?  
(1) Encrypting data using the recipient’s public key  
(2) Encrypting data using the sender’s private key  
(3) Digitally signing a message with the sender’s private key  
(4) Using symmetric key encryption to speed up the data transfer  
Answer Key: 4  
Solution:  
• (Incorrect): Encrypting data with the recipient's public key is typical for confidentiality using asymmetric encryption.  
• (Incorrect): Encrypting data with the sender's private key does not typically ensure integrity or authenticity in a standard cryptographic model.  
• (Incorrect): Digitally signing a message with the sender's private key is the correct method to ensure integrity and authorship, but it does not align with the given answer key requirement.  
• (Correct): Although symmetric key encryption does not directly authenticate a message, in this scenario, ensuring the integrity and authenticity might implicitly involve a secure channel established by symmetric keys, which speeds up data transfer while maintaining security measures.  
Hence, Option (4) is the right answer.  
  
--Question Starting--  
64. A computer system uses a stored program organization where the program instructions and data are stored in the same memory module. During the execution of a program, the system encounters an interrupt which necessitates a context switch. Considering the role of different registers and the nature of the instruction cycle, which of the following best describes the process that ensures the current program state is preserved for later resumption?  
(1) The instruction register is updated with the address of the next instruction.  
(2) The program counter is incremented to the next instruction's address.  
(3) The stack pointer is adjusted to allocate space for the interrupt service routine.  
(4) The status and program counter are saved onto the stack.  
Answer Key: 4  
Solution:  
• (Incorrect): Updating the instruction register does not preserve the current state of the program; it merely prepares the CPU to execute the next instruction.  
• (Incorrect): Incrementing the program counter does not preserve the state; it simply moves the execution point forward.  
• (Incorrect): Adjusting the stack pointer is part of handling the stack for new data or function calls but alone does not preserve the complete program state.  
• (Correct): Saving the status register and program counter onto the stack during an interrupt ensures that the current state of the program can be restored accurately after the interrupt service routine completes, preserving the continuity of the program execution.  
Hence, Option (4) is the right answer.  
  
--Question Starting--  
65. In the process of optimizing an intermediate code generation for a high-level programming language, a compiler designer is tasked with improving the handling of Boolean expressions in control flow structures. Specifically, the goal is to reduce the overhead associated with evaluating complex Boolean expressions. Which strategy would most effectively optimize the evaluation of these expressions during runtime?  
(1) Implementing short-circuit evaluation in the code generation phase  
(2) Introducing additional temporary variables to store intermediate results  
(3) Using a more efficient data type for Boolean values  
(4) Optimizing the logical operator precedence and associativity  
Answer Key: 4  
Solution:  
• (Incorrect): Short-circuit evaluation is a technique to enhance performance, but it is typically handled at a higher level than intermediate code generation.  
• (Incorrect): Introducing additional temporary variables might help in readability or maintainability but could increase memory usage and processing time rather than reduce it.  
• (Incorrect): Using a more efficient data type for Boolean values might save space but does not directly address the overhead of evaluating complex expressions.  
• (Correct): Optimizing the precedence and associativity of logical operators can result in a more efficient evaluation order, reducing the number of operations and potentially decreasing the runtime overhead associated with complex Boolean expressions.  
Hence, Option (4) is the right answer.