

Question on July 9

1. P2Pは中央サーバーがないため低コストで頑健なシステムが作れる反面、悪意ある参加者の排除が難しいというデメリットが分かりました。現在、ビットコインなどのブロックチェーンでは多数決によって信頼性を担保していますが、一般的なWebアプリケーションでP2Pを安全に実装する場合、どのようなセキュリティ対策が主流なのでしょうか？

➡ Webが基本的にはCS型なので、サーバサイドで管理。しかし、ここには分散型のアーキテクチャが頑張って導入されるのが超大規模サーバサイトですね。

2. 従来のWebシステムが、現在の個別のタスクごとにAIを使い分けているシステムと類似しているということが面白かったです。確かに、Webで起こった歴史をたどるならAIポータル的なものにどんどんシフトしていてもおかしくないと思います。しかし一方で、個々のサーバに明確な役割の違いがあるWebと違い、AIは「知能」というある種相対的な基準で図られる機能なので、性能の良い一社の独占勝ちになってしまう未来も見えます。
3. Winnyにおいて、送信元やキャッシュの保持者を隠蔽し「匿名性」を確保するために、技術的にどのような暗号化や端末間のリレー転送の仕組みが使われていましたか。

➡ Winnyではあまり匿名性確保の話は意識・実装していなかったかな。

4. 学年の変わり目の安全講習会でP2Pソフトを使わないことをとても強調された記憶があるが、あれはP2Pソフトを「悪用しない」という意味だったのかもしれないと思った。
5. CPUの熱密度が核反応炉並みというのは衝撃でした。実際に出てくる熱量はそこまでではないかもしれませんが、データセンターと発電所が組み合わさったような施設ができそうだと思いました。
→ 密度と量は違うからね。でも、大きな量になってきています。One DCで原発1基とか。
6. 包丁や自動車などが悪用されて人が亡くなっても、それらの技術の開発者や生産者が罪に問われることはないのにも関わらず、どうしてWinnyが悪用された結果、Winnyの開発者が逮捕されたのでしょうか？
→ 不思議だよね。。。。
7. Webサービスのところの概念(ユーザがいちいち一つのサイトにアクセスするのではなくて、それらを1つに集約したWebサービスにアクセスする)は、最近発表されたSakana aiのFuguに似てると思いました。(複数のLLMモデルを束ねて出力を作る)先生が最初の方におっしゃっていた、イノベーションとは模倣であるというところにSakana aiのFuguは合致していて面白かったです。→ 😊
8. 仮想記憶などをネットワークに応用するなどの話があり、やはりハードウェアとソフトウェアどちらもの理解が重要だと感じた。

9. クラウドコンピューティングが主流になった現在でも、Edge Heavy Computingが注目されている一番の理由は、通信遅延以外にもあるのか

➡ **データの機密性 とかね。。。**

10. 処理負荷の分散手法として「水平分散」と「垂直分散」の2つが挙げられていましたが、実際のWebサービスを設計する際、これらはどのような基準で使い分けられるのでしょうか？あるいは、現在の大規模サービスでは常に両方を組み合わせて使うのが前提なのでしょうか？

➡ **はい。大規模サービスでは、両方を使います。**

11. P2Pの仕組みについて、サーバーからアドレスを入手して通信するピアに最初から直接通信を開始すると思っていたので、キャッシュを用いて負荷を分散させていた事実には驚きました

12. Webを普段使っていると、その向こうにある物理的なサーバの存在はあまり意識していませんでしたが、実際には、負荷分散や熱の問題など、物理的な制約がやはり多くあるというのが興味深かったです。

13. XMLは自由にタグを定義できるが、Reactコンポーネントも自分で定義したコンポーネントを再利用するという意味ではXMLに似ているように感じられる。

9. クラウドコンピューティングが主流になった現在でも、Edge Heavy Computingが注目されている一番の理由は、通信遅延以外にもあるのか

➡ **データの機密性 とかね。。。。**

10. 処理負荷の分散手法として「水平分散」と「垂直分散」の2つが挙げられていましたが、実際のWebサービスを設計する際、これらはどのような基準で使い分けられるのでしょうか？あるいは、現在の大規模サービスでは常に両方を組み合わせて使うのが前提なのでしょうか？

➡ **はい。大規模サービスでは、両方を使います。**

11. P2Pの仕組みについて、サーバーからアドレスを入手して通信するピアに最初から直接通信を開始すると思っていたので、キャッシュを用いて負荷を分散させていた事実には驚きました

12. Webを普段使っていると、その向こうにある物理的なサーバの存在はあまり意識していませんでしたが、実際には、負荷分散や熱の問題など、物理的な制約がやはり多くあるというのが興味深かったです。

13. XMLは自由にタグを定義できるが、Reactコンポーネントも自分で定義したコンポーネントを再利用するという意味ではXMLに似ているように感じられる。 ➡ **似たところを認識するための抽象化の習慣を！**

14. NDNにおいて、暗号的ハッシュを用いたコンテンツ検証は量子計算機の発展によってどのような影響を受ける可能性があるのでしょうか。

→ 普通に、暗号化の話になるでしょ。

15. Webサーバの応答遅延は、ユーザの再クリックによってDDoS的状况を生みますが、そうした人間の行動まで含めたシステム設計は可能なのでしょうか

→ 再クリックをさせないための工夫をたくさん考えて実装します。仕事をしているふりをしたり。誘導したり。

16. キャッシュの話がアーキテクチャでやった通りでネットワークと通じていることがよくわかりました。

17. 検索結果が人によって違うのって分散処理のためだったんですね。完全に個人に合わせてたもんだと思ってました。

→ 努力をしているけど、ごまかしもできる (笑)

18. いろんな工夫があり、同じようなことを利用しているのもあり、そのようなものを生み出すには根本の理解が大事なんだと感じました

19. コンピュータアーキテクチャにおける分散処理がそのままサーバー負荷の分散処理に活かされたり、Webサービスにおいても登場してくることが、本日に限らず、講義全体を通じて理解できました。→ 😊

24. この講義を通してフレームワークやアーキテクチャの理解が深められました。応用するには基礎が大切という当然の事実は、最先端ばかりを見ていると忘れがちだと感じます。

→ 「素直な眼」で、「一步引いて観てみるかなあ。」

25. ライブなど人気なチケットの発売日によく起きる、チケット販売サイトにおける人間の焦りによるサーバーへの集中アクセスを防ぐためには、根本的にどのような対策が一番有効なのでしょうか。

→ 焦らせないように誘導する。 性能を上げる。

26. 処理能力やネットワークの速度が上ったことによりwebサイトの大きさがかなり大きいものが登場しておりこれが逆にUXを下げているような気がしています。

→ だから、、、分散処理に向かうしかない。

→ 熱・エネルギー密度、さらに、光は遅いが 顕在化してきたので、分散化は 必然になる。