OpenLDAP

トラブルの早期発見の為にできること

2010/09/10

日本LDAPユーザ会

はじめに

- 本セッションでは、業務システムとも連携する、止まって欲しくないディレクトリサーバ、OpenLDAPに起こりえるトラブルの兆候を感じとる、monitorデータベースの活用方法に焦点をあてていきます
  - 本セッションの内容は、OpenLDAP 2.4.23、CentOS 5.5にて、動作確認を行っております
  - 本資料での表記は、次のような感じとなっております

目次

- はじめに
  - 定期健診
  - コンピュータの世界では
  - LAMPでは
- monitorデータベース
  - 概要
  - 設定方法
  - 使いどころ
- 終わりに

みなさま、健康診断 行ってますか？

定期的な健康診断の良いところ

- 早期発見
  - 時間の経過とともに、広がっていくような病気であれば、早期発見、早期治療が有効
- 定期評価
  - 継続的に、健康状態の評価を続けることで、長期的な健康の維持、疾病の予防に有効

※健康診断の考え方には、個人差があるかもしれません。あくまで、本セッションの導入とお考え下さいませ。

定期健診で、大切なこと

1. １年に1回とか、定期的に

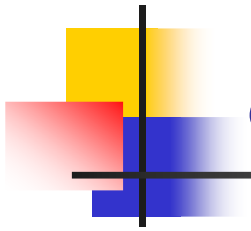
2. 同じ診療機関、同じ方法で健康診断を行い
   - 結果を記録し、過去との比較ができるように

3. 結果を、しっかり確認する

4. 問題があれば、アクションをとる

※みなさま、こんな感じで良いでしょうか。こちらも、本セッションに限っての導入とお考え下さいませ。

コンピュータの
世界では…？



どちらかというと …

- キャパシティ管理、リソース監視、性能監視…などと、呼ばれている分野
  - 定期的に取得したデータを蓄積しておき、リソースの利用状況の長期的な傾向を把握することが主な目的
  - すぐどうって事はないけど、グラフが描けると良い世界
- “障害監視” と呼ばれる分野ではない
  - 一定の閾値を超えた場合、トラップや、メール、表示上の色を変えるなどして通知し、早急に対処することが目的の監視

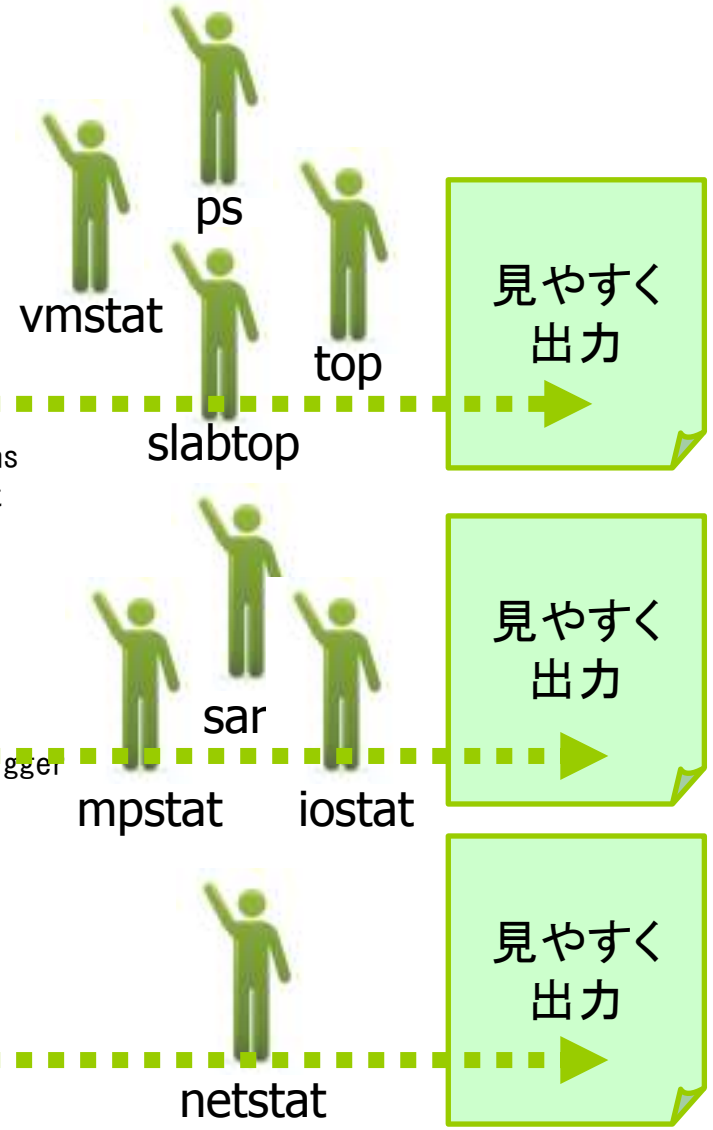
※本セッションの内容は、SNMPや、MIB、OIDなどを利用した監視方法を説明するものではありません。
OpenLDAPサーバに特化した、サーバプロセス内部の情報の取得方法を説明することで、安定した運用を支援することを目的としています。

身近な **LAMP**の世界、
内部情報の提供と取集
は、どうやって …？

LAMPの「L」、Linux

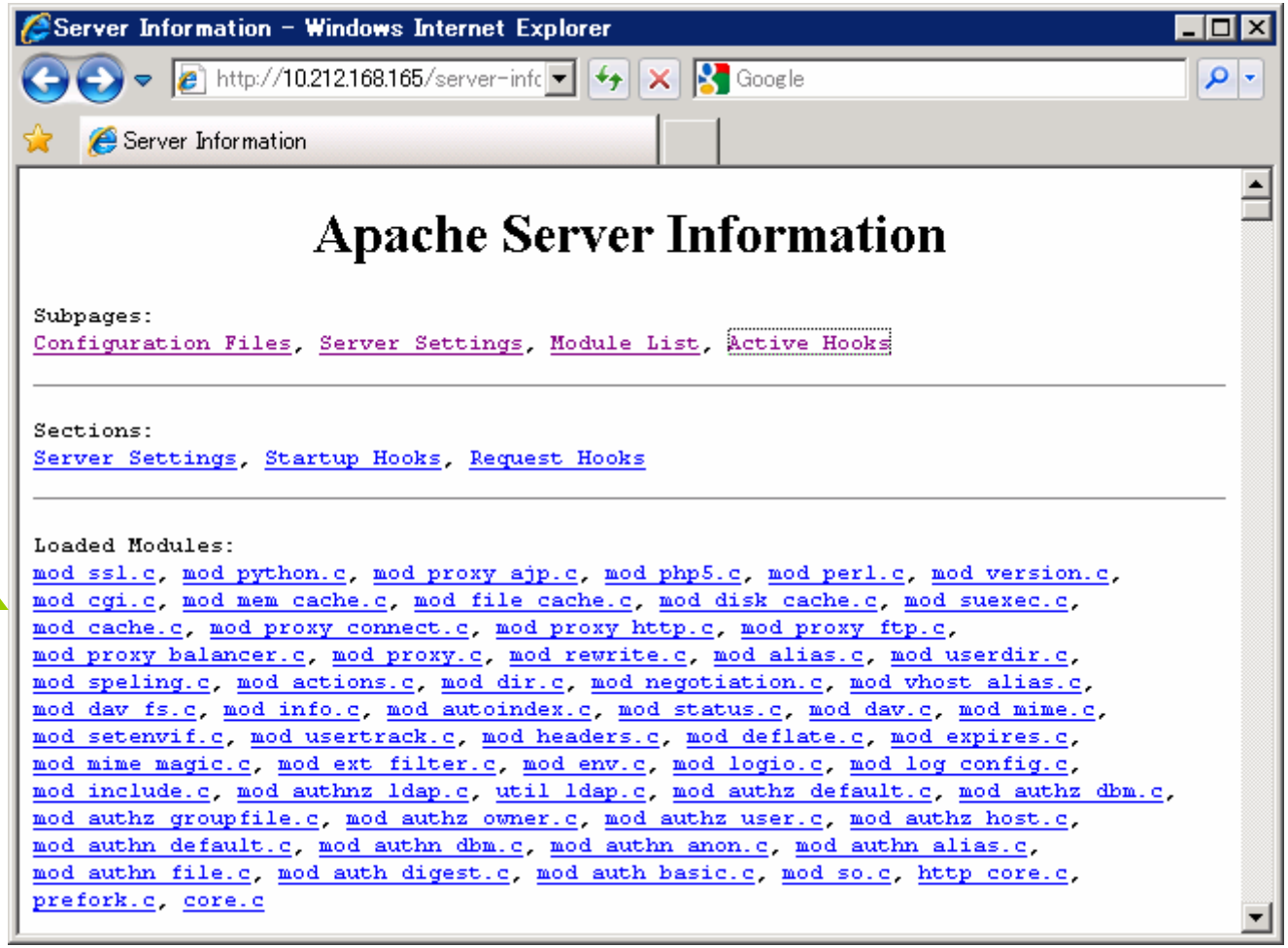
- Procファイルシステム

```
# ls /proc/
1       1885  2198   27727  379          devices      modules
10      1891  2228   27879  392          diskstats    mounts
1038    1912  2229   2831   4            dma          mpt
1037    1950  2289   2833   4093         driver       mtrr
1313    1960  2295   29900  418          execdomains  net
1314    1970  2296   29902  451          fb           partitions
1315    1971  2300   3      5            filesystems  schedstat
1380    1979  2310   30190  5518         fs           scsi
1695    1984  2316   30213  5520         ide          self
1697    1993  2319   30341  6            interrupts   slabinfo
170     2     2320   30464  7            iomem        stat
171     2012  2398   317    8            ioports      swaps
172     2033  2400   32     9            irq          sys
173     2050  2401   33     90           kallsyms     sysrq-trigger
174     2055  2415   3351   91           kcore        sysvipc
175     2068  2417   34     94           key-users    tty
1785    2077  2427   361    96           keys         uptime
1796    2091  2428   3611   acpi         kmsg         version
1823    2121  27     362    buddyinfo    loadavg      vmcore
1824    2129  27118  363    bus          locks        vmstat
183     2139  27120  367    cmdline      mdstat       zoneinfo
1855    2148  27166  368    cpuinfo      meminfo
1870    2181  27169  369    crypto       misc
```



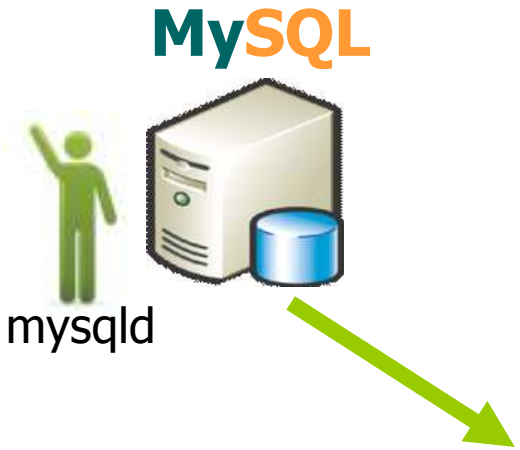
LAMPの「A」、Apache http server

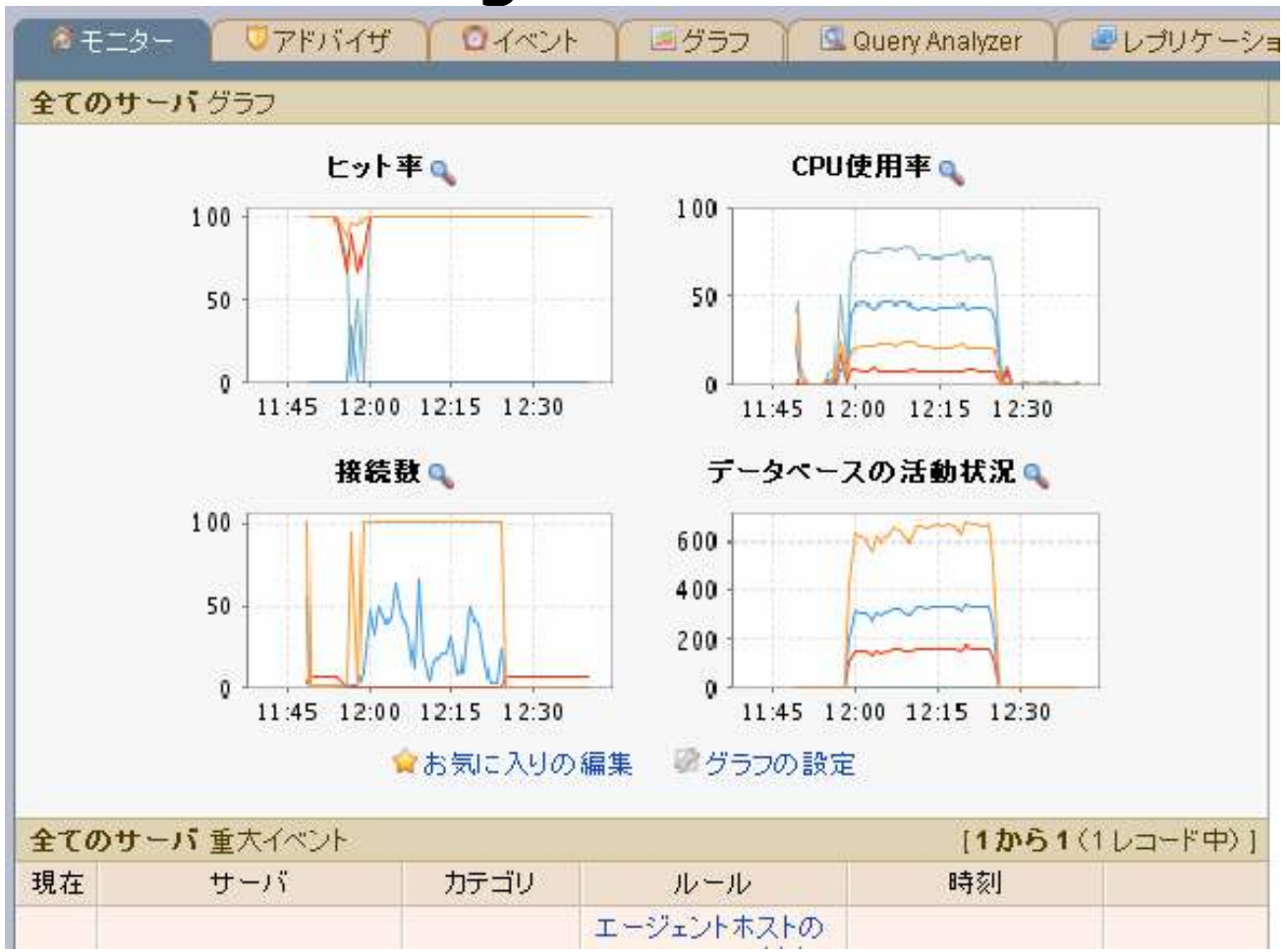
- mod_status、mod_info とか …



LAMPの「M」、MySQL

- show status、show engine status とか …



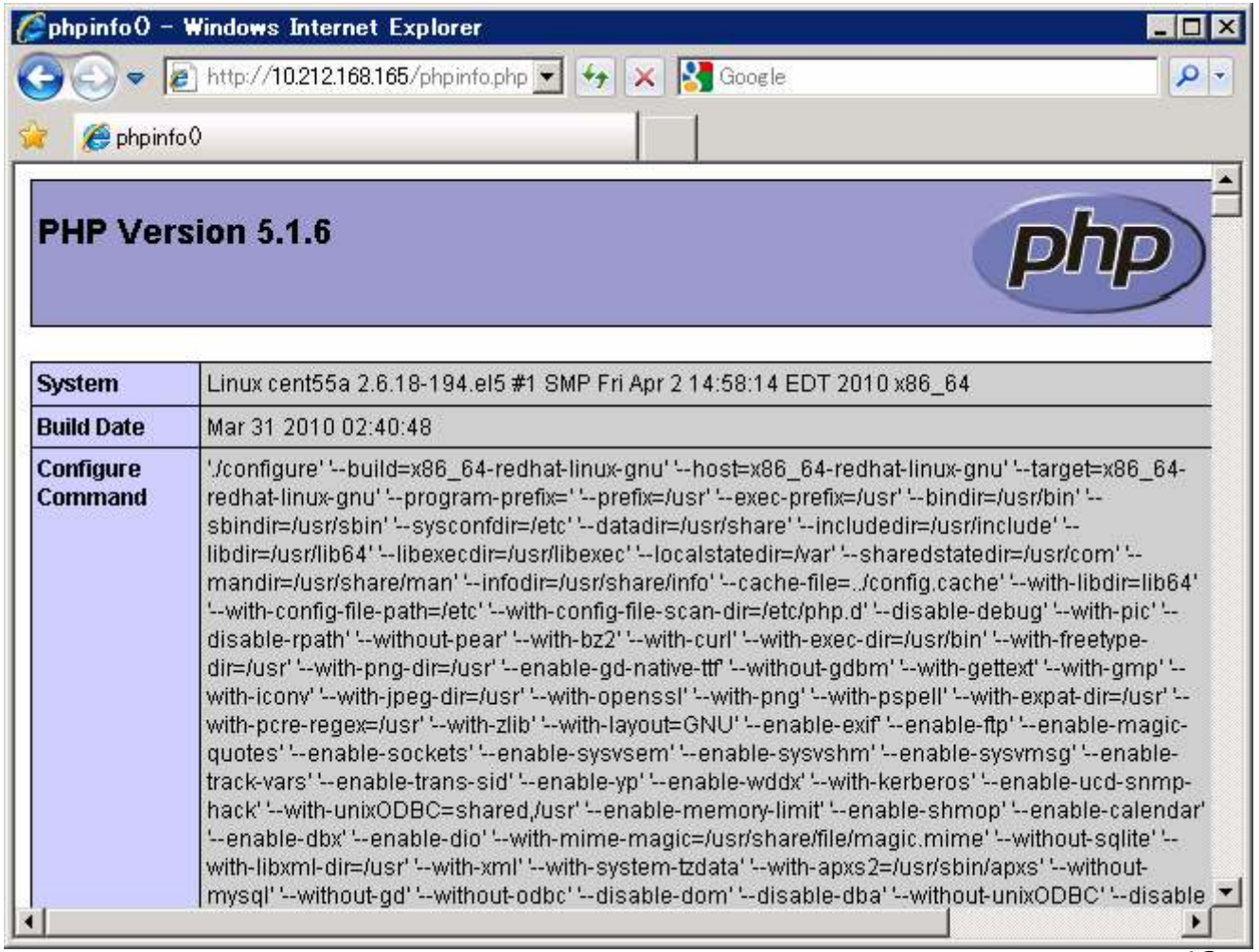


※グラフは、商用版MySQL Enterpriseに付属する、MySQL Enterprise Monitorを利用し、MySQLの内部情報をイメージしやすく表示させたものです。

LAMPの「P」、例えば…PHP

- phpinfo とか…

httpd (PHP)

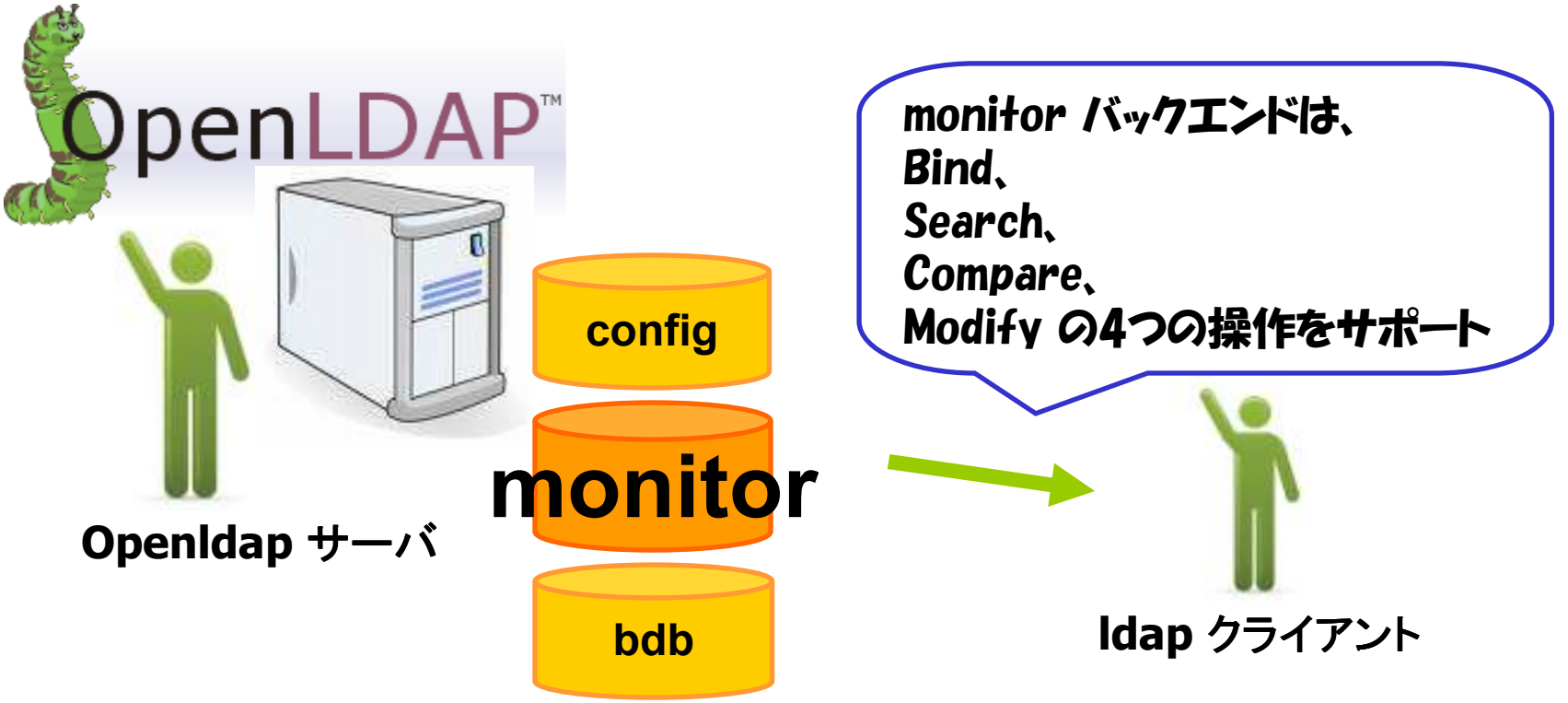


OpenLDAP ?

monitor データベース

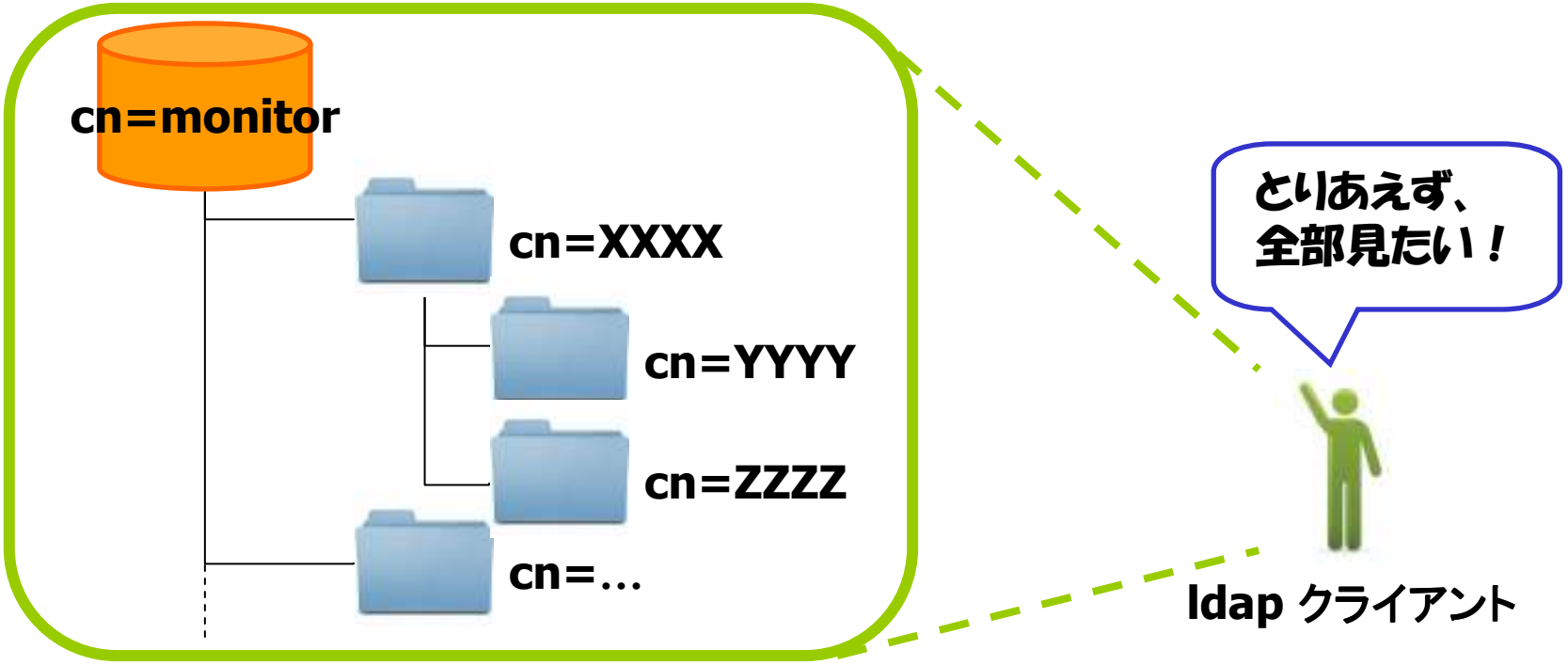
monitor データベースの概要

- 内部ステータスを管理するデータベース
  - OpenLDAPサーバの内部ステータスを、LDAPプロトコルで公開する、バックエンドDB



LDAPクライアントからの、情報収集

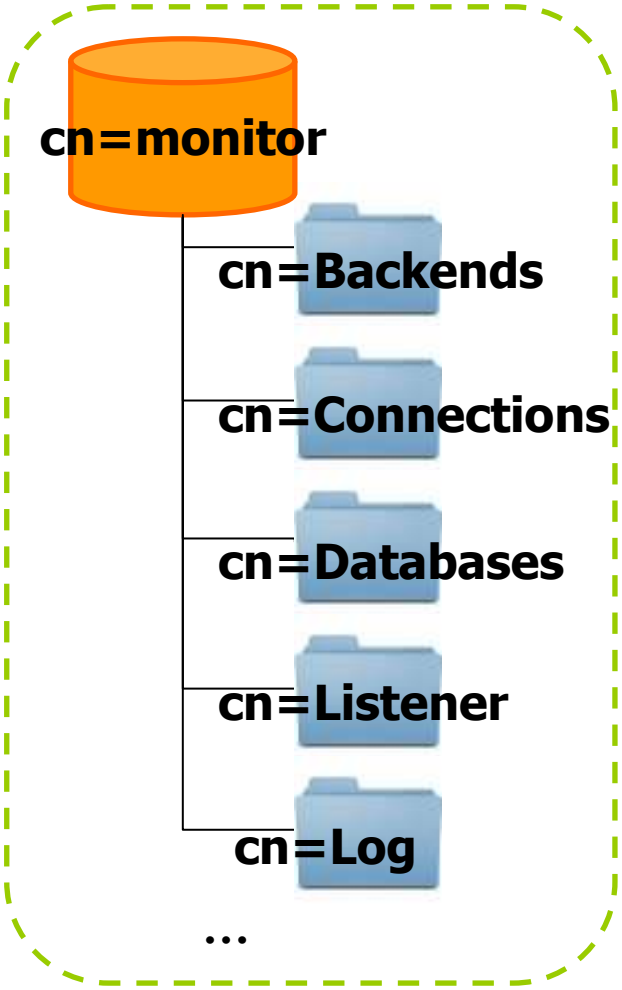
- cn=monitor 以下から、情報を取得
- ほとんどのステータスは、運用属性として取得可能
  - ldapsearch -x -b cn=Monitor -s sub +



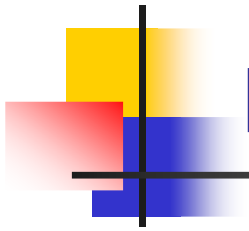
※上記ldapsearchコマンドでの"+"記号は、全ての運用属性を要求しています。

monitor データベースの内容

- cn=monitor のサブエントリ



RDN	内容
Backends	利用可能なバックエンドデータベース
Connections	コネクションに関連する統計情報
Databases	利用中の各バックエンドDBの情報
Listener	OpenLDAPサーバのリスニング情報
Log	ログレベルの動的な変更
Operations	サーバ全体の、各操作の実行回数の統計
Overlays	利用可能なオーバレイ機能
SASL	(未実装)
Statistics	サーバ全体の送信データ量に関する統計
Threads	ワーカースレッドの動作状況に関する情報
Time	OpenLDAPが起動した時刻、uptimeなど
TLS	(未実装)
Waiters	現時点での、送信、受信処理の実行数



monitor データベースの歴史

- OpenLDAP 2.0系
  - 実装なし
- OpenLDAP 2.1系
  - 2.1.2にて、--enable-monitor=[no] を、デフォルトのコンフィグレーションとして登場
- OpenLDAP 2.2系
  - 2.2.4にて、コンフィグのデフォルトは、--enable-monitor=[yes] に変更
- OpenLDAP 2.3系
  - ひたむきに成長
- OpenLDAP 2.4系
  - monitorに詳細な情報を提供するバックエンドDBが現れた為、起動時、ログへ「monitorを有効にして」と警告を出すことも

警告メッセージ

- 起動時、次の警告メッセージが出力される条件

```
…
monitoring disabled; configure monitor database to enable
…
```

  - --enable-monitor=yes でコンパイルされている
  - slapd.confに、database monitor の設定がない
  - バックエンドDB毎に、次の条件にあてはまる
    - bdb、hdb の場合、monitoring off の指定がない
    - ldap の場合、monitoring on を指定している

バックエンド	monitorへの詳細情報の提供
bdb	デフォルトで提供 (monitoring on)
hdb	デフォルトで提供 (monitoring on)
ldap	デフォルトでは提供しない (monitoring off)

monitor の設定

実践

3ステップで、準備完了

- コンパイル時に、--enable-monitor=yes を指定
  - OpenLDAP 2.2以降は、デフォルトでコンパイル対象
- 設定ファイルに、monitor DBと、ACLを追加

```
include         /usr/local/openldap/…/core.schema
…[略]…
database monitor
access to dn.subtree="cn=Monitor"
            by dn.exact="cn=Admin,cn=Monitor" write
rootdn          "cn=Admin,cn=Monitor"
rootpw          secret
```

- OpenLDAPを再起動

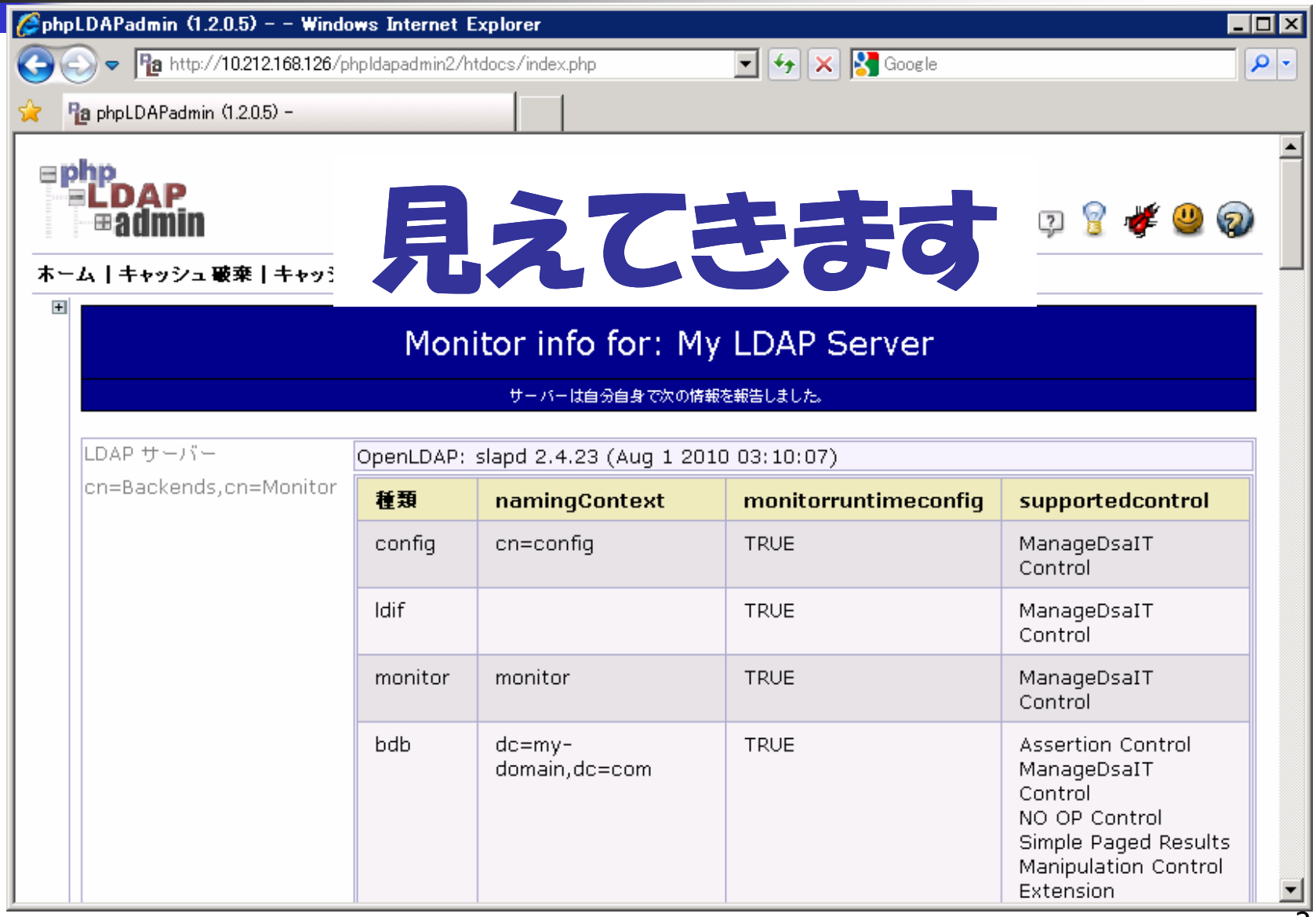
※本資料では、これ以降、monitorデータベースへのアクセスに必要となる認証に、ここで指定した”cn=Admin,cn=Monitor”と、”secret”を利用します。

実践

GUI ツールから、見えます

実践

ldapsearchして、こんな感じで



長期的なトレンドを把握しよう

- コネクション数の傾向
- OpenLDAPサーバ全体の処理数

悩み

コネクション数の傾向

- OpenLDAPサーバが起動してから、これまでの、平均コネクション数が知りたい
- ある時間帯の平均コネクション数が知りたい
- コネクション数の増加に、どのような傾向があるかを知りたい

悩み

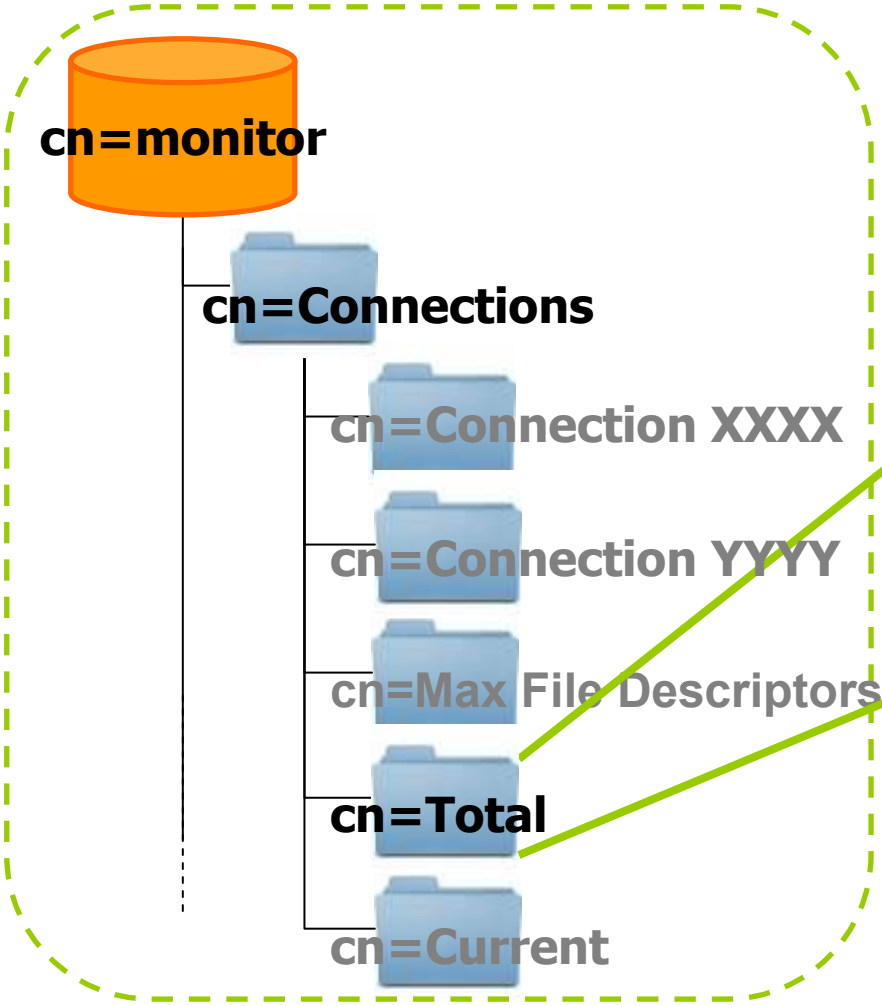
コネクション数の傾向

- OpenLDAPサーバへの、クライアントからのコネクション数の傾向がわかると
  - メンテナンスの為の、停止計画が立てやすい
  - 拡張計画が立てやすい
  - 性能に関するトラブルを、未然に防げる
- わからないと、勘や経験が重要に …

事実

クライアントからの接続数

- cn=Total,cn=Connections,cn=monitor の属性



属性	内容
monitorCounter	OpenLDAPサーバが起動してから、クライアントからの接続を確立した数の累積

実践

monitorに、聴いてみよう

- OpenLDAPサーバ起動後の累積コネクション

```
$ ldapsearch -x ¥
> -D cn=Admin,cn=Monitor -w secret ¥
> -b "cn=Total,cn=Connections,cn=Monitor" ¥
> monitorCounter -LLL
```

(答え)

dn: cn=Total,cn=Connections,cn=Monitor

monitorCounter: ***1017***

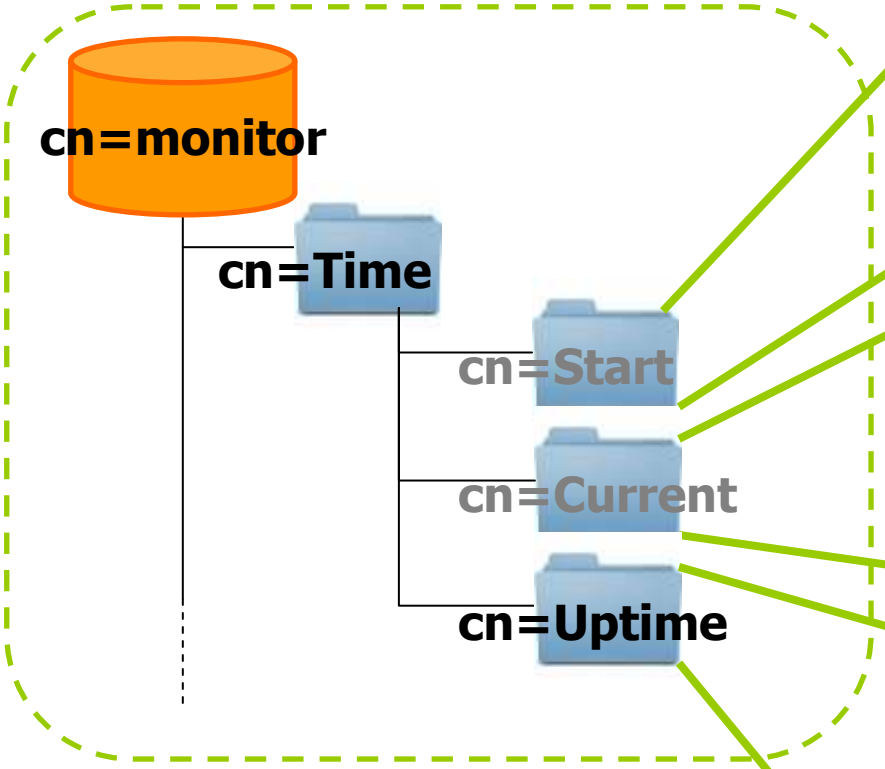
- 1000を引いた値、**17**が正解
  - OpenLDAP 2.4.19までは、そのままの値

※openLDAP 2.4.20での、ある修正で、OpenLDAPサーバ内部でのコネクション数が、1000からカウントされるようになってしまいました。

事実

次に、OpenLDAPサーバの時間

- cn=Time,cn=monitor のサブエントリの属性



属性	内容
monitorTimestamp	OpenLDAPサーバの起動時に、システムより取得した時刻

属性	内容
monitorTimestamp	cn=Currentへのアクセス時に、システムより取得する、現在の時刻

属性	内容
monitoredInfo	現在の時刻から、OpenLDAPサーバが起動した時刻を引いた差分を秒数で表示 (cn=Current) - (cn=Start)

※上記属性の時間は、time(2)関数でシステムより取得した時刻が利用されます。仮想環境など、時刻が不正確になりやすい環境での利用にはご注意下さい。

実践

起動秒数と、平均コネクション数

- OpenLDAPが起動している秒数、Uptime

```
$ ldapsearch -x ¥
> -D cn=Admin,cn=Monitor -w secret ¥
> -b "cn=Uptime,cn=Time,cn=Monitor" ¥
> monitoredInfo  -LLL
```

(答え)

dn: cn=Uptime,cn=Time,cn=Monitor

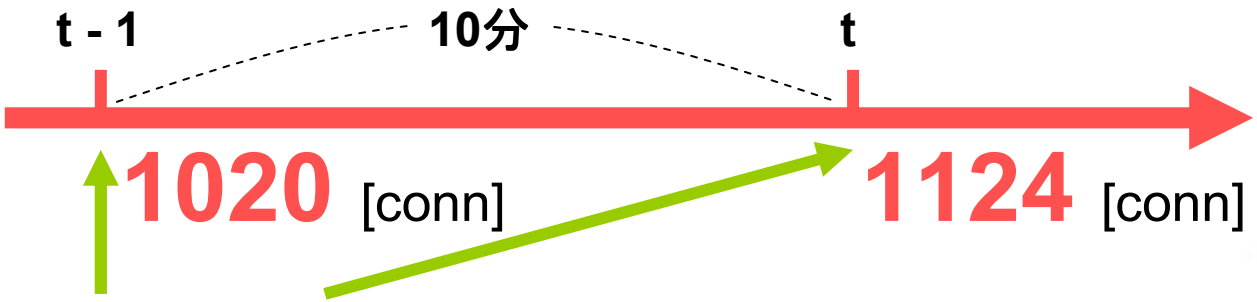
monitoredInfo: ***100***

- 起動後からの平均コネクション数
  - 累積コネクション数 / Uptime
  - 17 / 100 = **0.17** [conn/sec]

実践

ある期間の平均コネクション数

- 10分毎に、コネクションの累積数を確認




```
$ ldapsearch -x ¥
> -D cn=Admin,cn=Monitor -w secret ¥
> -b "cn=Total,cn=Connections,cn=Monitor" ¥
> monitorCounter -LLL
```

- ある10分間の平均コネクション数

$$\frac{1124 - 1020}{600} = 0.1733 \text{ [conn/sec]}$$

悩み

OpenLDAPサーバ全体の処理数

- サーバ起動後からの処理の傾向を知りたい
  - どんな処理が多いのか
  - 参照系、更新系処理の割合
- ある時間帯での処理の傾向を知りたい
- 処理の増加傾向があれば、知りたい

悩み

OpenLDAPサーバ全体の処理数

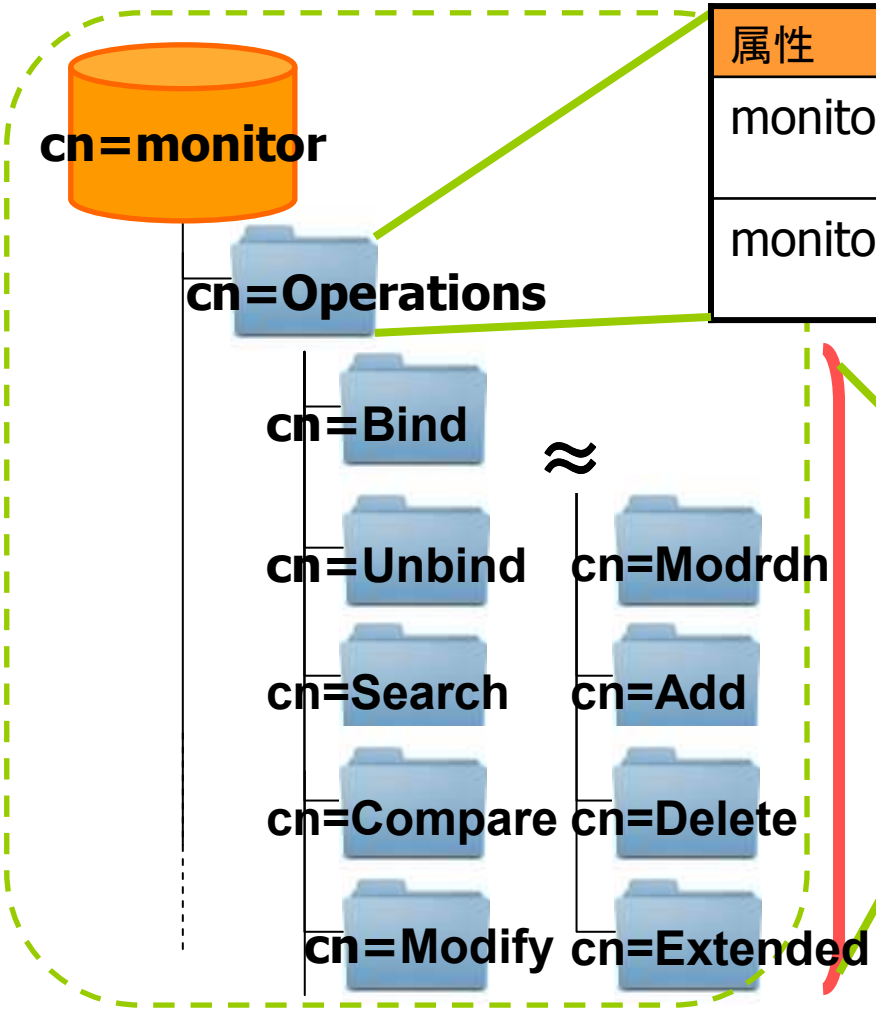
- サーバ単位での処理の傾向がわかると
  - 少ないセッションでも高負荷な時間が把握できる
  - どんな拡張や負荷分散が良いかを検討できる
  - バックアップに適した時間がわかる

- わからないままでの運用は、不安 …

事実

monitorによる処理数のカウント

- cn=Operations,cn=monitorと サブエントリの属性



属性	内容
monitorOpInitiated	OpenLDAPサーバ全体で、開始された、全ての処理数
monitorOpCompleted	OpenLDAPサーバ全体で、完了した、全ての処理数

属性	内容
monitorOpInitiated	当該の処理が、OpenLDAPサーバ全体で、開始された数
monitorOpCompleted	当該の処理が、OpenLDAPサーバ全体で、完了した数

実践

サーバ全体の個別操作の統計

- サーバ起動後からの、個々の操作の完了数

```
$ ldapsearch -x ¥
> -D cn=Admin,cn=Monitor -w secret ¥
> -b "cn=Operations,cn=Monitor" -s one ¥
> monitorOpInitiated monitorOpCompleted ·
dn: cn=Bind,cn=Operations,cn=Monitor
monitorOpInitiated: 39

monitorOpCompleted: 39
…[略]…
dn: cn=Search,cn=Operations,cn=Monitor
monitorOpInitiated: 43

monitorOpCompleted: 42
…[略]…
```

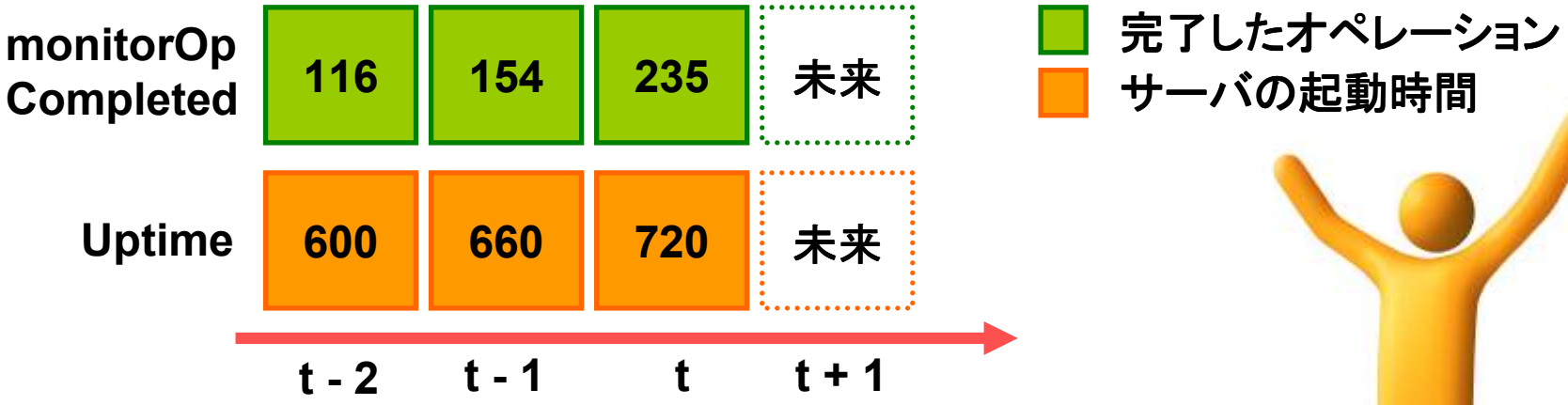
認証の回数

検索の回数

実践

ある期間の、サーバ全体の統計

- ある期間での、個別、または全てのオペレーションが完了した数



- ある操作の、ある期間の平均は…

$$\frac{\text{monitorOpCompleted}(t) - \text{monitorOpCompleted}(t-1)}{\text{Uptime}(t) - \text{Uptime}(t-1)}$$

トラブルの芽は、摘んでおこう

- クライアントからの最大接続数
- 各コネクションの動作概要

悩み

クライアントからの最大接続数

- 同時接続可能なクライアント数が分からない
  - 同時接続可能なクライアント数には制限がある
    - 実際は、利用可能なファイルディスクリプタ数の制限
    - Linuxのデフォルトは、1024 とか
  - OpenLDAP起動前に、シェルから変更可能
    - ulimit -n 8192; ./libexec/slapd …とか

悩み

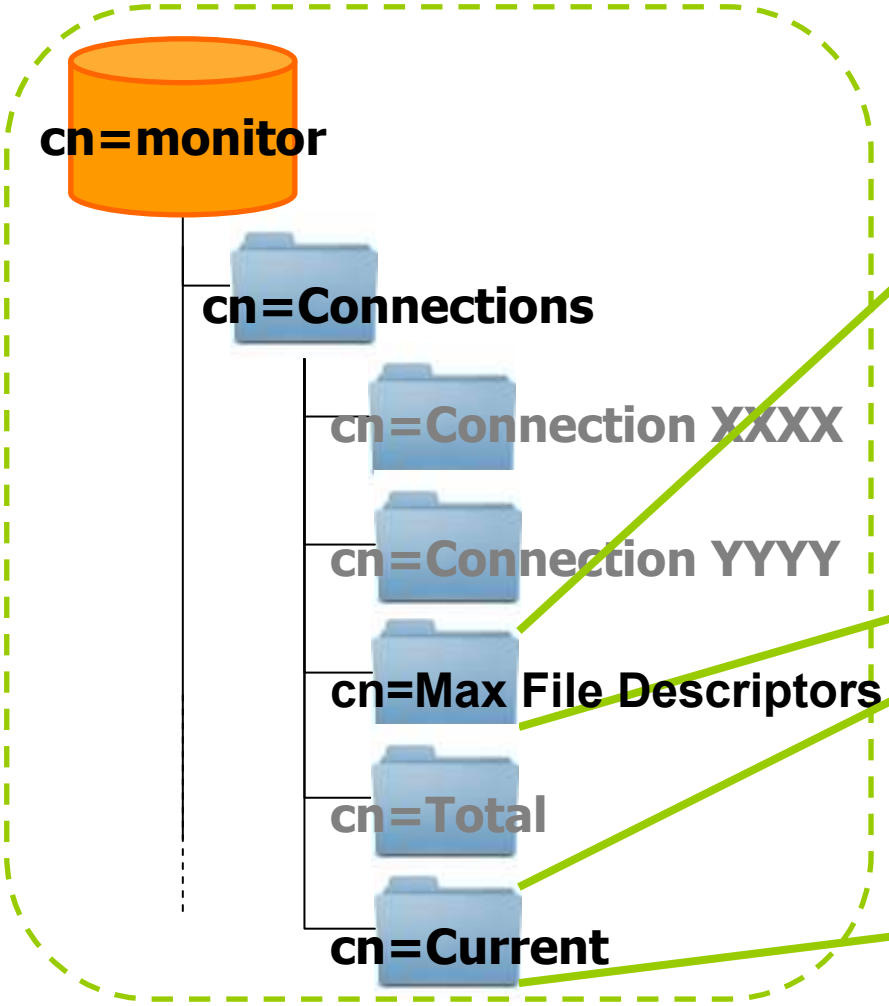
ファイルディスクリプタ数の制限

- ファイルディスクリプタ数の制限を超えると、新しく接続するクライアントは、接続待ちに
  - 接続中のクライアントが終了し、ファイルディスクリプタが利用可能になるまで、接続待ち
  - クライアント側では、レスポンスタイムの悪化に
- クライアントからの接続数の上限、知りたい

事実

最大接続数に関する情報

- cn=Connections,cn=monitor サブエントリの属性



属性	内容
monitorCounter	現在、起動しているOpenLDAPサーバが、利用可能なファイルディスクリプタの数

属性	内容
monitorCounter	起動しているOpenLDAPサーバに、現在、接続しているクライアントの数

実践

利用可能なファイルディスクリプタ数

- monitorに、聴いてみよう

```
$ ldapsearch -x ¥
> -D cn=Admin,cn=Monitor -w secret ¥
> -b "cn=Max File Descriptors,cn=Connections,cn=Monitor" ¥
> monitorCounter -LLL
```

(答え)

dn: cn=Max File Descriptors,cn=Connections,cn=Monitor

monitorCounter: **8192**

あっ、変更してた！

※既に、接続数が上限に達している場合は、monitorデータベースに接続できません。これは、接続数が上限に達することを予防する為の確認方法です。
接続数が上限に達している場合は、サーバ機にログインした後、cat /proc/`cat slapd.pid`/limits などを用いて確認下さい。

実践

現時点での、同時接続数

- monitorに、聴いてみよう

```
$ ldapsearch -x ¥
> -D cn=Manager,dc=my-domain,dc=com -w secret ¥
> -b "cn=Current,cn=Connections,cn=Monitor" ¥
> monitorCounter -LLL
```

(答え)

dn: cn=Current,cn=Connections,cn=Monitor

monitorCounter: **1**

- 今、この接続だけ
- 8192 - 1 ＝ 8191

ぜんぜん、大丈夫！

※接続数が、利用可能なファイルディスクリプタの上限に達している場合は、monitorデータベースに接続できません。これは、予防目的での同時接続数の確認です。
接続数が、利用可能なファイルディスクリプタの上限に達している場合は、サーバ機にログインした後、netstart、lsof、…などを用いて確認下さい。
ファイルディスクリプタは、クライアントと確立するTCP/IPコネクションの他、OpenLDAPサーバがオープンするファイル、例えば.bdbファイルにも消費されています。

悩み

各コネクション動作概要

- 思っていたより、同時接続数が多い気がする
  - どんなクライアントが接続しているのか知りたい
  - 無駄に長い接続を維持しているクライアントがいないか確認したい
    - しっかり、処理を繰り返していて接続が長いの？
    - それとも、ただ、つながってるだけなの？

悩み

各コネクション動作概要

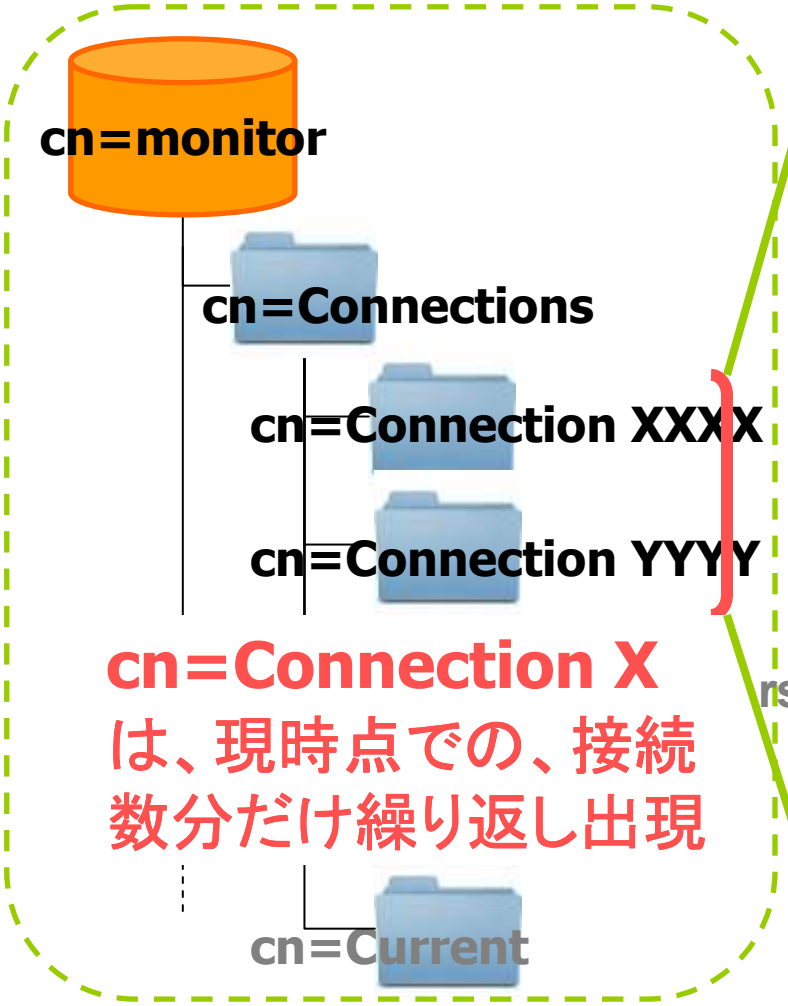
- どんなクライアントが、どんな内容の接続を維持しているかがわかると、対処しやすい
  - ulimit -n で対処するか …
  - idletimeout、writetimeout で対処するか …
  - クライアント側で、処理を見直すか …
- わからないと、ベストな対処を選択できない

※idletimeoutは、OpenLDAPサーバに接続したまま何も操作を行っていないクライアントを、一定の秒数が経過した後に切断する為のディレクティブです。
また、writetimeoutは、OpenLDAPサーバからのデータ送信が、一定秒数以上経過しても完了しない場合に中止する為のディレクティブです。

事実

各コネクションに関する情報

- cn=Connection XXXX の属性



属性	内容
monitorConnectionNumber	コネクション番号。ログファイルの「conn=xxxx」と同じ番号
monitorConnectionProtocol	当該のコネクションが利用している、プロトコルバージョン
monitorConnectionOpsReceived	当該コネクションから受信した操作(Bind、Searchなど)の数
monitorConnectionOpsExecuting	当該のコネクションから受信した操作を、実行している数
monitorConnectionOpsPending	当該のコネクションから受信した操作を、実行待ちしている数
monitorConnectionOpsCompleted	当該コネクションから受信した操作が、完了した数

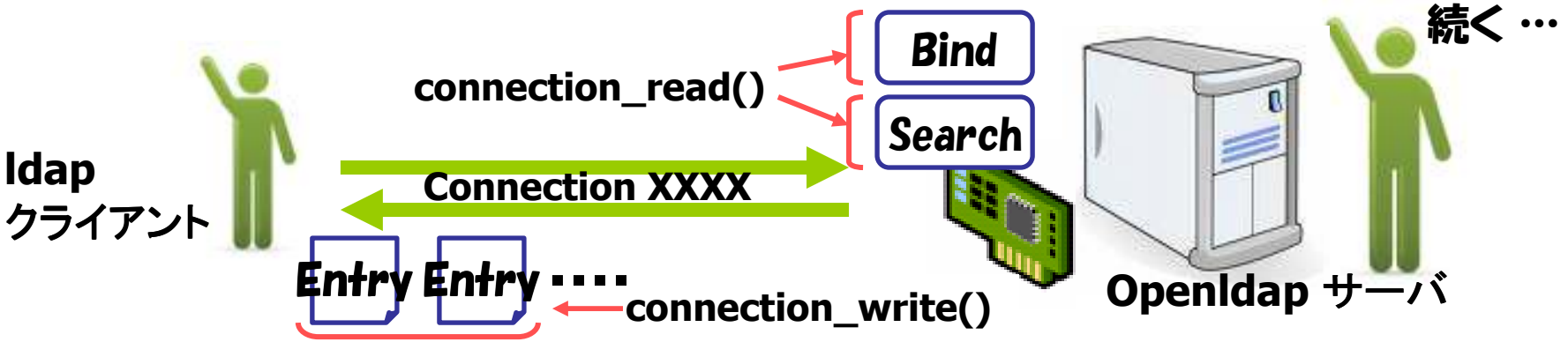
続く …

事実

各コネクションに関する情報 (2)

- cn=Connection XXXX の属性

属性	内容
monitorConnectionGet	当該のコネクションで、connection_get()関数が利用された回数(connection_read()と、connection_write()が利用された回数の合計)
monitorConnectionRead	当該のコネクションで、データ受信処理を行う、connection_read()関数が利用された回数
monitorConnectionWrite	当該のコネクションで、データ送信処理を行う、connection_write()関数が利用された回数
monitorConnectionMask	当該コネクションの状態をマスクして表示 (r:受信可能な状態、w:送信中、x:実行中の操作あり、p:実行待ちの操作あり、S:SASL認証中 …など)

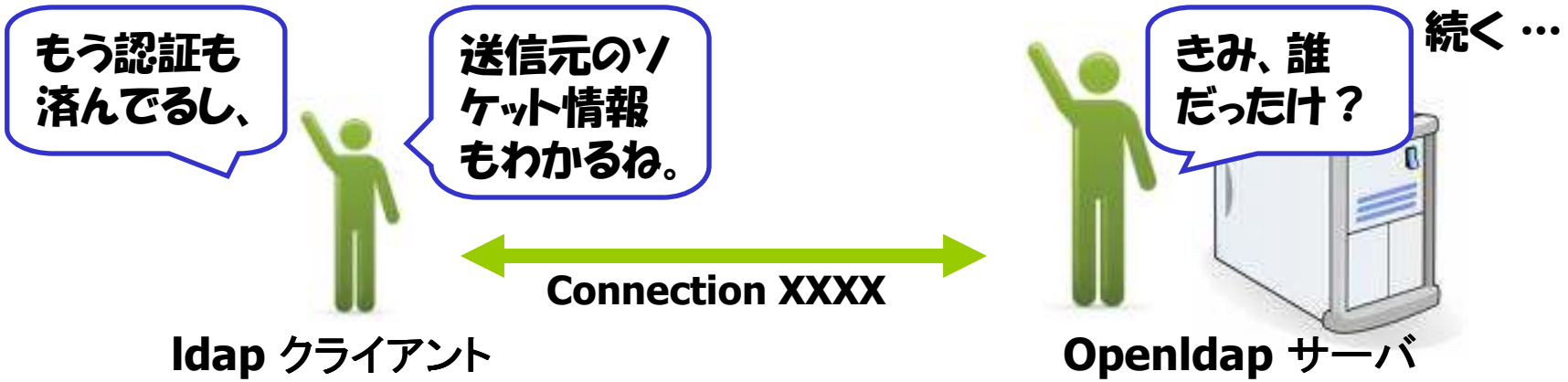


事実

各コネクションに関する情報 (3)

- cn=Connection XXXX の属性

属性	内容
monitorConnectionAuthzDN	当該のコネクションで、認証されたDN
monitorConnectionListener	当該のコネクションが接続した、リスナーのURL(リスニングするIPアドレスと、ポート番号)
monitorConnectionPeerDomain	当該のコネクションの、接続元ドメイン
monitorConnectionPeerAddress	当該のコネクションの、接続元IPアドレスと、ポート番号
monitorConnectionLocalAddress	当該のコネクションが接続した、ソケット情報(バインドしているIPアドレスと、ポート番号)

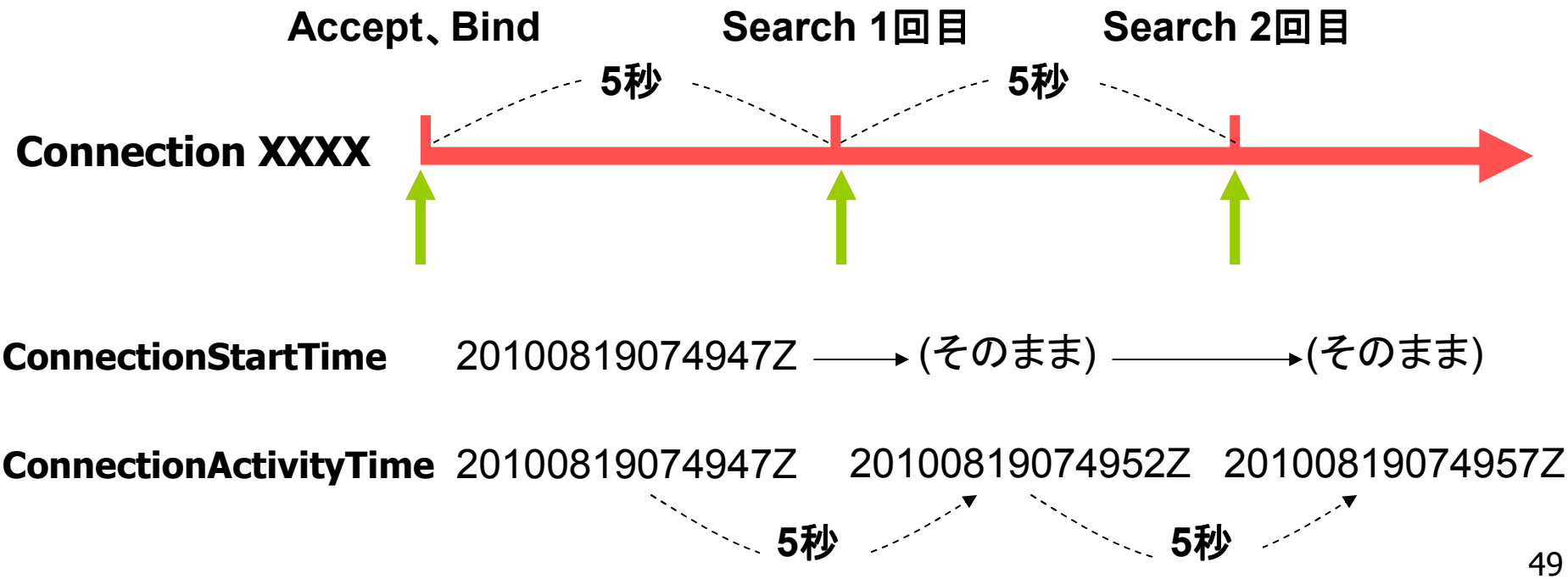


事実

各コネクションに関する情報 (4)

- cn=Connection XXXX の属性

属性	内容
monitorConnectionStartTime	当該のコネクションが、開始された時刻
monitorConnectionActivityTime	当該のコネクション内での、直近の操作(Bind、Search、Add、modify、Deleteなど)の開始時刻



実践

あるコネクションの状態チェック

- 特定のコネクションの動作状況が気になる

```
$ ldapsearch -x ¥
> -D cn=Admin,cn=Monitor -w secret ¥
> -b "cn=Connection XXXX,cn=Connections,cn=Monitor" + -LLL
…[略]…
monitorConnectionOpsReceived: 2
monitorConnectionOpsExecuting: 1
monitorConnectionOpsPending: 0
monitorConnectionOpsCompleted: 1
…[略]…
monitorConnectionAuthzDN: uid=xx,ou=yy,dc=zz,dc=com
monitorConnectionListener: ldap:///
monitorConnectionPeerDomain: unknown
monitorConnectionPeerAddress: IP=10.x.x.x:42947
monitorConnectionLocalAddress: IP=0.0.0.0:389
monitorConnectionStartTime: 20100819074947Z
monitorConnectionActivityTime: 20100819074947Z
```

オペレーションの実行状況

認証ユーザ、コネクション情報

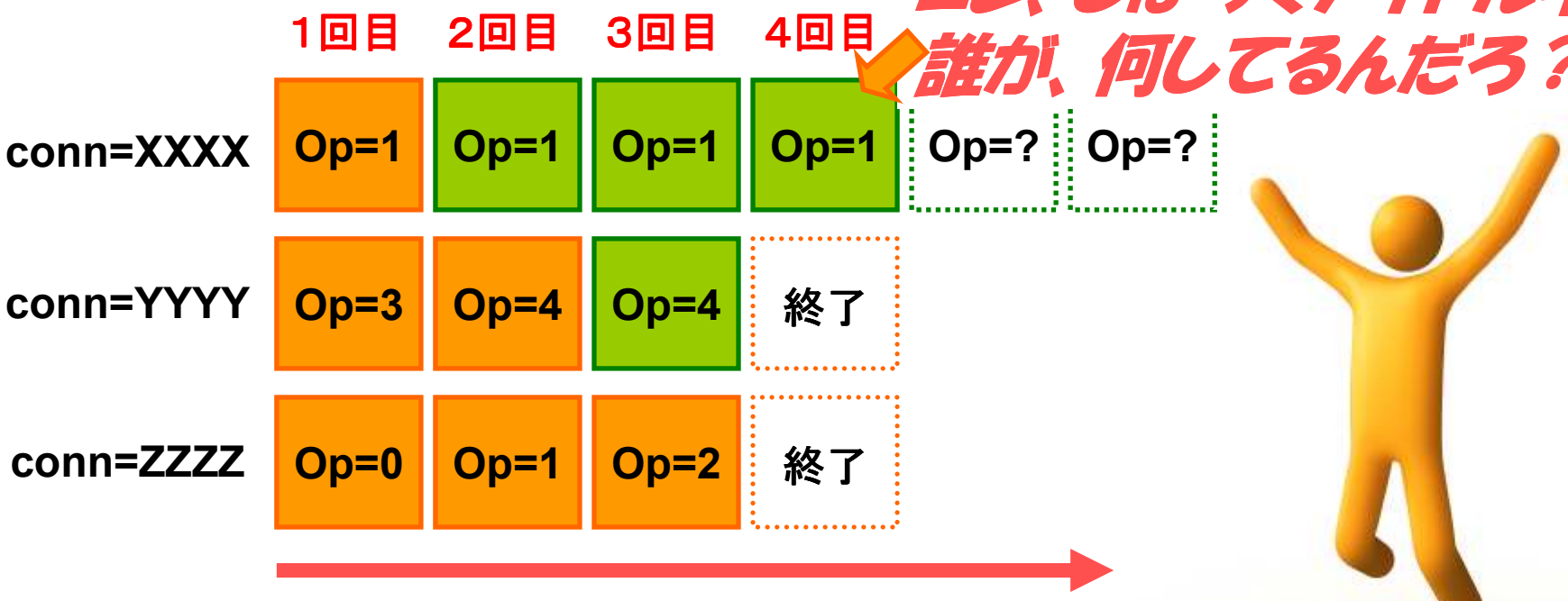
コネクション開始日時

オペレーション開始日時

実践

気になるものは、複数回チェック

- 複数回のチェックで、状態の変化が判明



ActivityTimeと操作番号が、前回と変わっていないコネクション

ActivityTimeと操作番号が変わっているか、接続直後のコネクション

※このように、状態の変化を確認することで、調査対象とするコネクションを、ある程度、絞り込むことができます。
ここでは、Javaのスレッドダンプや、MySQLの「show processlist;」を数回実行するような、動作状況の変化を確認するデバック方法をイメージ下さい。

突然デバックしたい、あなたに …

- ログの出力レベルと、デバック

悩み

ログの出力レベルと、デバック

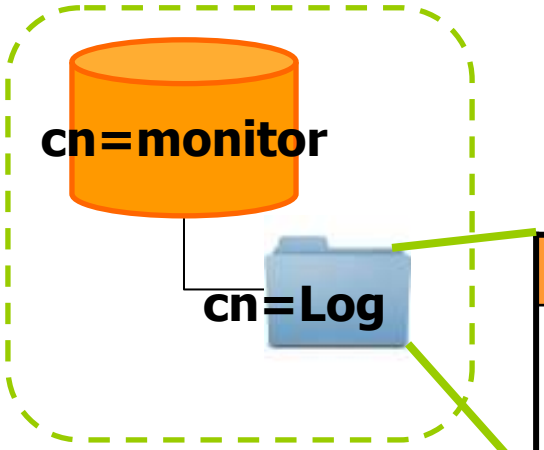
- ログの出力量があまりに多く、デフォルトのstats(256)から、stats2(512)等に変更した
  - ログファイルへの出力量は減ったけど …
  - 誰が、何処から接続しているかわからない

- 気になる接続は、どう追跡すればいいの？

事実

起動中、ログレベルは変更可能

- Trace
- Packets
- Args
- Conns
- BER
- Filter
- Config
- ACL
- Stats
- Stats2
- Shell
- Parse
- Sync



属性	内容
managedInfo (ユーザ属性)	実行中に、ログの出力レベルを、文字列で指定して変更可能

左の文字列なら、
大小文字区別なし。
マルチバリューで
いいから送って！

Openldap サーバ

実践

一時的な、ログレベルの変更例

- OpenLDAPサーバの起動中に、外部から、ログレベルを、一時的に変更できます
  - vi log.ldif

```
dn: cn=Log,cn=Monitor
changetype: modify
add: managedInfo
managedInfo: ACL
managedInfo: Stats
```

```
dn: cn=Log,cn=Monitor
changetype: modify
replace: managedInfo
managedInfo: Stats
```

  - ldapmodify -x -D cn=Admin,cn=Monitor -w secret -f log.ldif

※monitorデータベースからのログレベルの変更は、一時的な変更であり、OpenLDAPサーバの再起動後は、変更内容は維持されません。恒久的なログレベルの設定には、slapd.confのloglevelディレクティブをご利用下さい。

チューニング中の、あなたに …

- キャッシュ領域の利用状況

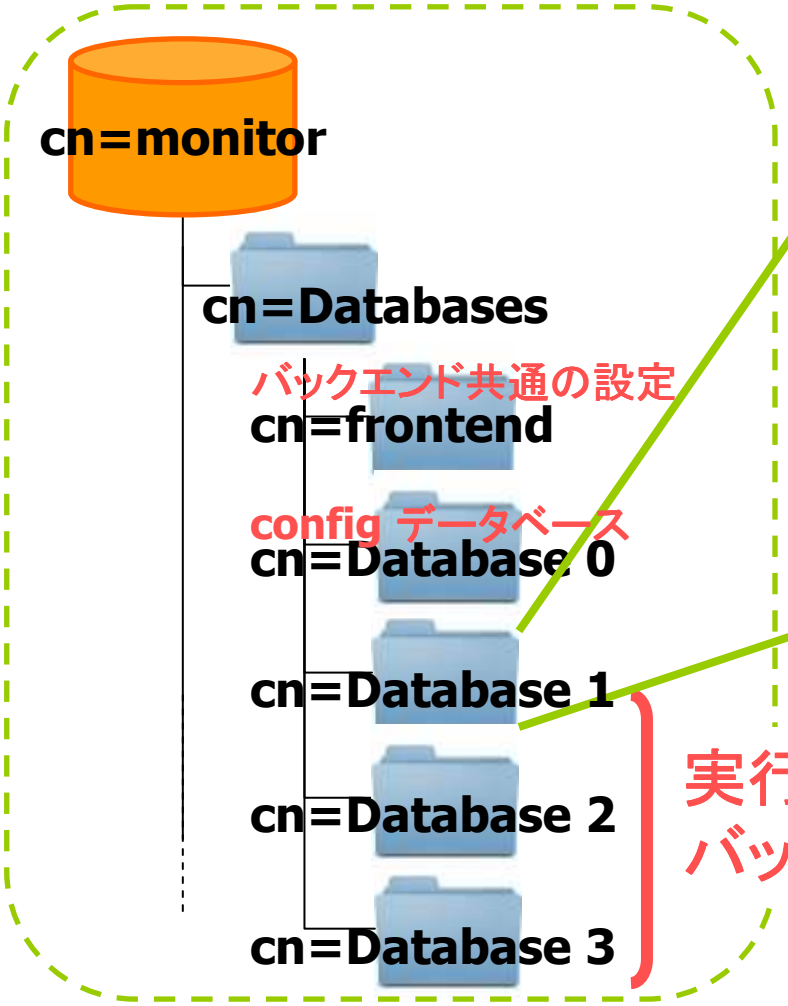
悩み

キャッシュ領域の利用状況

- バックエンドDB(BDB、HDB)に設定した、キャッシュ領域の利用状況を確認したい
  - エントリキャッシュ
  - IDリストキャッシュ
- 不足して、キャッシュ領域と、Berkeley DB間で入れ替えが発生していないだろうか…？

事実 cn=Databases,cn=monitor

- cn=Database X が、bdb、hdbの場合の属性



“Database 1”が、bdb、またはhdbである場合

属性	内容
olmBDBEntryCache	エントリキャッシュの利用数
olmBDBDNCache	DNキャッシュの利用数
olmBDBIDLCache	IDリストキャッシュの利用数
olmDbDirectory	データディレクトリ

実行時に、**slapd.conf**で指定した
バックエンド**DB**の数だけ繰り返し出現

実践

実際のエントリキャッシュの状態

- エントリ情報を、OpenLDAP側でデコードして、キャッシュする領域の空き状態は？

```
$ ldapsearch -x ¥
> -D cn=Admin,cn=Monitor -w secret ¥
> -b "cn=Database 1,cn=Databases,cn=Monitor" ¥
> olmBDBEntryCache -LLL
```

(答え)

dn: cn=Database 1,cn=Databases,cn=Monitor

olmBDB**Entry**Cache: ***1***

- まだ、**1** つだけ

まだまだ入る！

※Berkeley DBから取得したエントリは、OpenLDAPサーバ側にデフォルトで1000エントリまでがキャッシュされます。チューニングの観点からは、バックエンドDBに格納されたエントリと同数の値を設定することが推奨されています。バックエンドDB毎に、cachesizeディレクティブを用いて調整下さい。

実践

実際のIDリストキャッシュの状態

- インデックスファイルから取得したIDリストを、OpenLDAP側で保持する領域の空きは？

```
$ ldapsearch -x ¥
> -D cn=Admin,cn=Monitor -w secret ¥
> -b "cn=Database 1,cn=Databases,cn=Monitor" ¥
> olmBDBIDLCache -LLL
```

(答え)

dn: cn=Database 1,cn=Databases,cn=Monitor

olmBDB**IDL**Cache: ***1***

- まだ、**1** つだけ

まだまだ入る！

※インデックスに紐づく id のリストは、デフォルトではキャッシュされない設定となっています。バックエンドDB毎に、idlcachesizeディレクティブで調整下さい。
パフォーマンスチューニングの観点からは、bdbではエントリ数と同数。hdbではエントリ数の3倍の値を設定することが推奨されています。

思い出せない、あなたに…

- 利用可能なバックエンドDB
- 利用可能なオーバレイ機能
- IP、ポートのリスニング状況

悩み

利用可能なバックエンドDB

- コンフィグレーション時の設定を忘れた！
- config.log もない！
  - make clean とか、
  - もう一度 ./configure して上書きしているとか…
- 今動いているOpenLDAPサーバで、使えるバックエンドデータベースが、わからない …

悩み

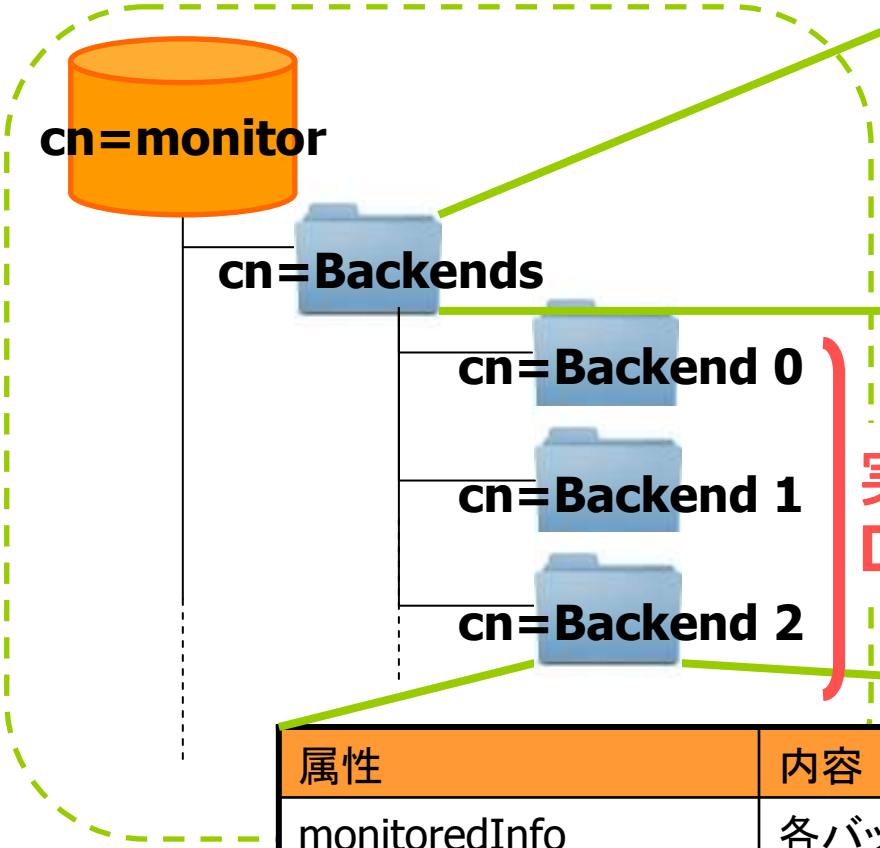
コンパイルオプション、ありすぎ

No.	オプション	デフォルト	備考
1	enable-backends	no	全部含めるかどうか
2	enable-bdb	**yes**	Berkeley DB
3	enable-dnssrv	no	DNS
4	enable-hdb	**yes**	hierarchical DB
5	enable-ldap	no	LDAP検索するプロキシ
6	enable-meta	no	LDAP検索するプロキシ
7	enable-monitor	**yes**	モニタ
…			
15	enable-sql	no	RDBMS
-	(config)	(**yes**)	(選択不可)

事実

cn=Backends,cn=monitor

- cn=Backends,cn=monitor とサブエントリの属性



属性	内容
monitoredInfo	OpenLDAPサーバに、コンパイルされ組み込まれているか、実行時にロードされているバックエンドDB

実行時に利用可能なバックエンドDBの種類の数だけ繰り返し出現

属性	内容
monitoredInfo	各バックエンドDBの名前。Backend 0は、config
monitorRuntimeConfig	実行時、構成可能か。TRUE / FALSE で表示
supportedControl	当該のバックエンドDBでサポートされるコントロール

実践

今、構成可能なバックエンドDB

- monitorデータベースに、聴いてみよう

$ ldapsearch -x ¥
> -D cn=Admin,cn=Monitor -w secret ¥
> -b "cn=Backends,cn=Monitor" -s base monitoredInfo -LLL
(答え)
dn: cn=Backends,cn=Monitor
monitoredInfo: ***config***
monitoredInfo: ***ldif***
monitoredInfo: ***monitor***
monitoredInfo: ***bdb***
monitoredInfo: ***hdb***
monitoredInfo: ***…***

思い出した！

悩み

利用可能なオーバレイ機能

- コンフィグレーション時の設定を忘れた！
- config.log もない！
  - make clean とか、
  - もう一度 ./configure して上書きしているとか…
- 今動いているOpenLDAPサーバで、使えるオーバレイ機能が、わからない …

悩み

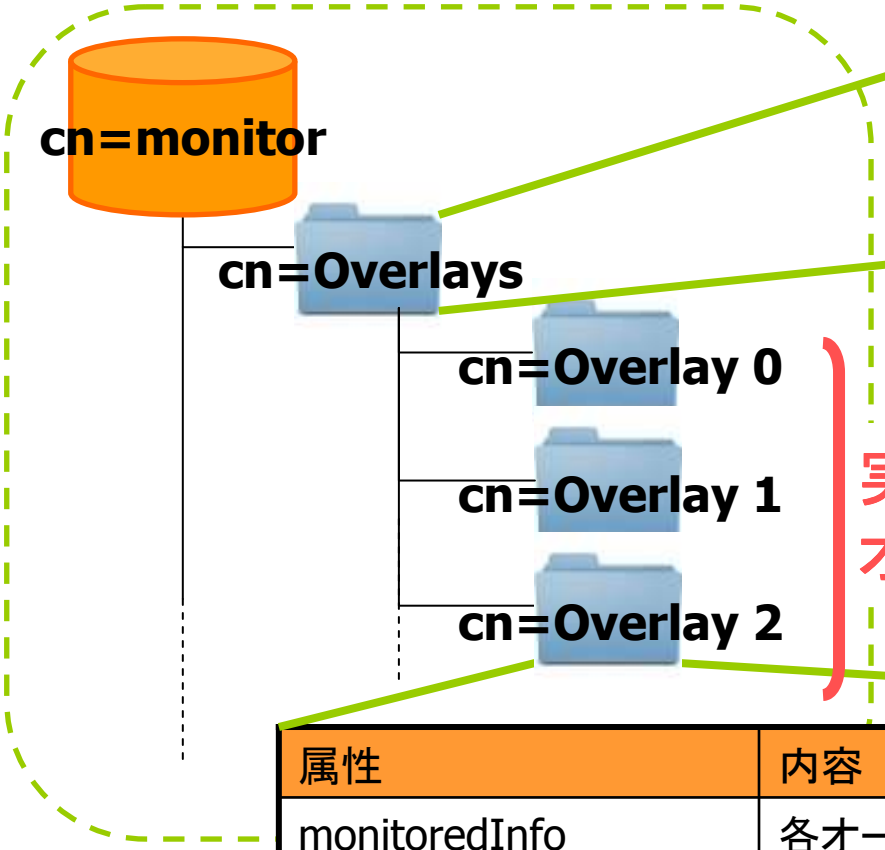
コンパイルオプション、ありすぎ

No.	オプション	デフォルト	備考
1	enable-overlays	no	全部含めるかどうか
2	enable-accesslog	no	アクセスログ
3	enable-auditlog	no	監査ログ
4		no	
5		no	
…		no	
18	enable-syncprov	**yes**	(唯一のデフォルトyes)
19	enable-translucent	no	上書き可能なプロキシ
20	enable-unique	no	属性値がユニークか
21	enable-valsort	no	複数の属性値のソート
-	(glue)	(**yes**)	(選択不可)

事実

cn=Overlays,cn=monitor

- cn=Overlays,cn=monitor とサブエントリの属性



属性	内容
monitoredInfo	OpenLDAPサーバに、コンパイルされ組み込まれているか、実行時にロードされているオーバーレイ機能

実行時に利用可能な、
オーバレイ機能の数だけ出現

属性	内容
monitoredInfo	各オーバレイ機能の名前
monitorRuntimeConfig	実行時、構成可能か。TRUE / FALSE で表示

実践

今、構成可能なオーバレイ機能

- monitorデータベースに、聴いてみよう

$ ldapsearch -x ¥
> -D cn=Admin,cn=Monitor -w secret ¥
> -b "cn=Overlays,cn=Monitor" -s base monitoredInfo -LLL
(答え)
dn: cn=Overlays,cn=Monitor
monitoredInfo: ***glue***
monitoredInfo: ***syncprov***

- あ、デフォルトだった …

思い出した！

悩み

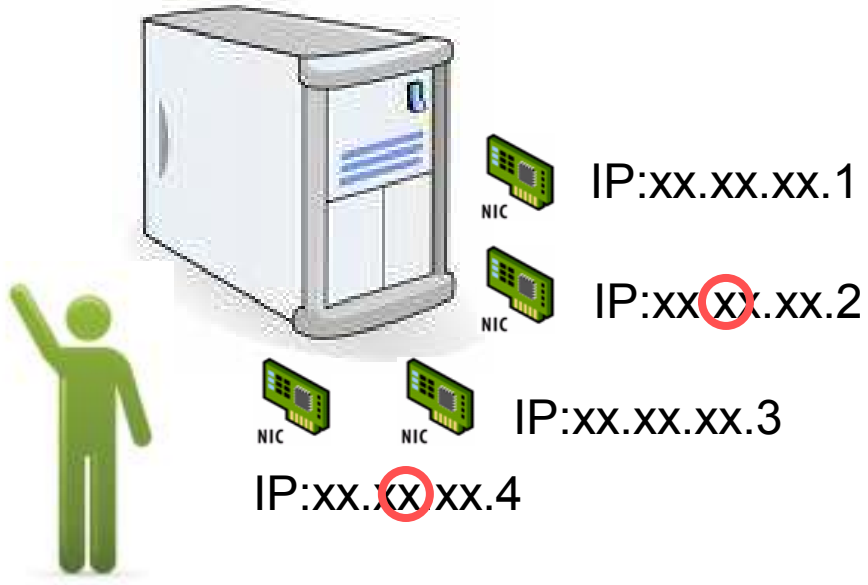
IP、ポートのリスニング状況

- 確認したいけど …
  - 起動パラメータを忘れた
    - slapd –h ldap://xx.xx.xx.xx:port/ … とかで起動したっけ？
  - OpenLDAPサーバのマシンにログインして、起動時に与えたパラメータとかを確認したくない
    - ps -eo args | grep slapd とか、netstat とかイヤ

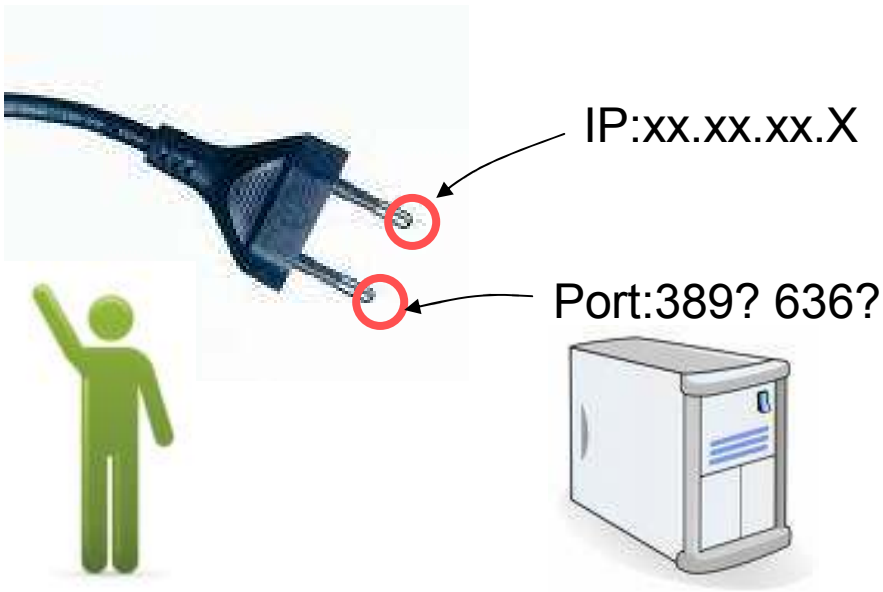
事実

OpenLDAPのリスニング設定

- 運用上の理由、セキュリティ上の理由により、OpenLDAPサーバの管理者は、リスニングするIPアドレス、ポート番号を調整可能



このIPにバインディング

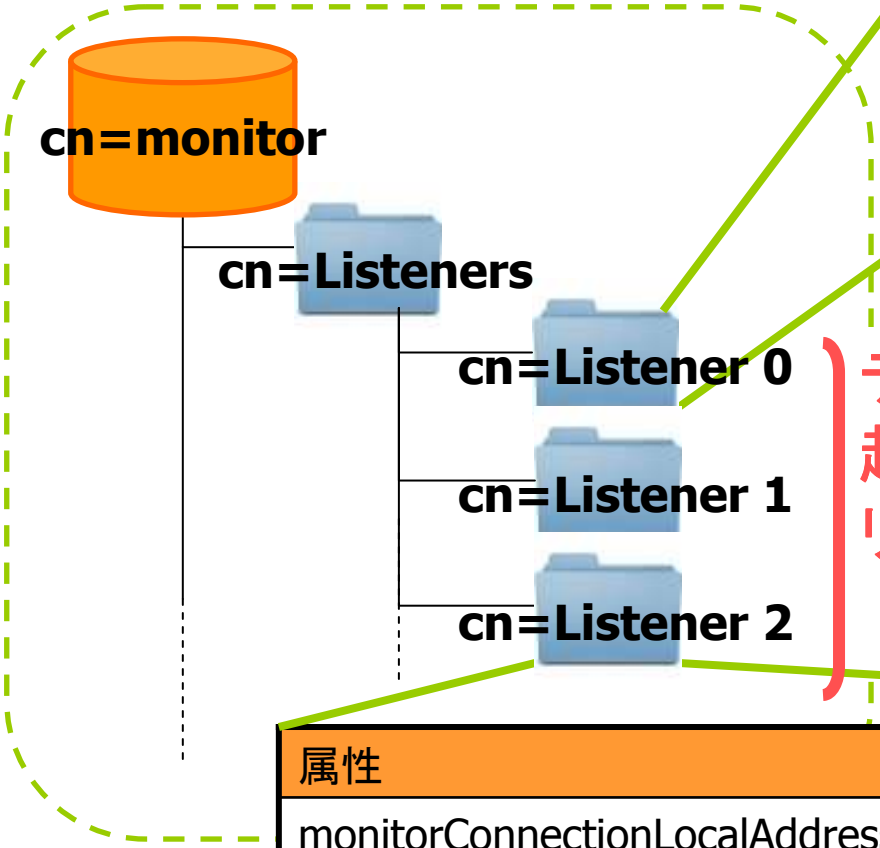


このポートでソケット通信

事実

cn=Listeners,cn=monitor

- cn=Listeners,cn=monitor のサブエントリの属性



属性	内容
labeledURI (ユーザ属性)	OpenLDAPサーバのリスニング情報を、URLで表記 デフォルトは、ldap:///

デフォルト、または、**OpenLDAP起動時に -h URLs で指定した、リスニングするURLの数だけ出現**

例) slapd -h "xx.xx.xx.xx:389 yy.yy.yy.yy:389"

属性	内容
monitorConnectionLocalAddress	IP=address:port と、リスニング情報を表記 デフォルトは、[::]:389と、0.0.0.0:389
monitoredInfo	ldapsでリスニングの場合、"TLS"と表示

実践

実際にリスニングするIPとポート

- monitorデータベースに、聴いてみよう

```
$ ldapsearch -x ¥
> -D cn=Admin,cn=Monitor -w secret ¥
> -b "cn=Listeners,cn=Monitor" -s one ¥
> monitorConnectionLocalAddress labeledURI -LLL
```

(答え)

dn: cn=Listener 0,cn=Listeners,cn=Monitor

labeledURI: ***ldap:///***

monitorConnectionLocalAddress: ***IP=0.0.0.0:389***

- IPバインドなし
- ポートも389だけ

思い出した！

※上記の内容は、全くリスニングするIPアドレス、ポート番号がわからず、monitorデータベースへ接続できない状況を想定したものではありません。
monitorデータベースに接続できない場合は、サーバ機にログインし、ps、netstatコマンドなどを用いてOpenLDAPサーバのリスニング情報をご確認下さい。

終わりに …

monitor データベース参考情報

- マニュアル
  - http://www.openldap.org/doc/admin24/monitoringslapd.html
- README
  - less servers/slapd/back-monitor/README [*]
- man
  - man doc/man/man5/slapd-monitor.5 [*]
- 説明
  - ldapsearch -x -b "cn=Monitor" description
- ソースコード
  - ls servers/slapd/back-monitor/ [*]

※ [*] 印の付いたコマンド実行時の各ファイルへのパスは、tar zxvf openldap-2.4.23.tgz; cd openldap-2.4.23 からとなります。

ご清聴、ありがとうございました！